



**The Twelfth Conference
On Language Engineering (ESOLEC'2012)
December 12-13, 2012**

Organized by

Egyptian Society of Language Engineering (ESOLE)

Under the Auspices of

**PROF. DR. HUSSEIN EISSA
President of Ain Shams University**

**PROF. DR. SHERIF HAMMAD
Dean, Faculty of Engineering, Ain Shams University**

**CONFERENCE CHAIRPERSON
PROF. DR. M. A. R. GHONAIMY**

**CONFERENCE COCHAIRPERSON
PROF. DR. SALWA ELRAMLY**

**Faculty of Engineering –Ain Shams University
Cairo, Egypt**

<http://esole-eg.org>

Conference Chairman:

Prof. Dr. M. R. A. Ghonaimy

Conference Sponsors

Technical Program Committee:

Prof. Taghrid Anber , **Egypt**
Prof. I. Abdel Ghaffar , **Egypt**
Prof. M. Ghaly, **Egypt**
Prof. M. Z. Abdel Mageed, **Egypt**
Prof. Khalid Choukri, ELDA, **France**
Prof. Nadia Hegazy, **Egypt**
Prof. Christopher Ciri, LDC, **U.S.A**
Prof. Mona T. Diab, Stanford U., **U.S.A**
Prof. Ayman ElDossouki, **Egypt**
Prof. Afaf AbdelFattah, **Egypt**
Prof. Y. ElGamal, **Egypt**
Prof. M. Elhamalaway, **Egypt**
Prof. S. Elramly, **Egypt**
Prof. H. Elshishiny, **Egypt**
Prof. A. A. Fahmy, **Egypt**
Prof. I. Farag, **Egypt**
Prof. Magdi Fikry, **Egypt**
Prof. Wafaa Kamel, **Egypt**
Prof. S. Krauwer, **Netherlands**
Prof. Bente Maegaard, CST, **Denmark**
Prof. A. H. Moussa, **Egypt**
Prof. M. Nagy, **Egypt**
Prof. A. Rafae, **Egypt**
Prof. Mohsen Rashwan, **Egypt**
Prof. H.I. Shaheen, **Egypt**
Prof. S.I. Shaheen, **Egypt**
Prof. Hassanin M. AL-Barhamtoshy, **Egypt**
Prof. M. F. Tolba, **Egypt**
Dr. Tarik F. Himdi, **Saudi Arabia**



Organizing Committee

Prof. I. Farag	Prof. S. Elramly
Prof. Hany Kamal	Prof. H. Shaheen
Dr. Passant El-kafrawy	Dr. Fatma Newaigy
Eng. Manar Ahmed	Eng. Mona Zakaria

Conference Secretary General

Prof. Dr. Salwa Elramly



The Twelfth Conference on Language Engineering Final Program

Wednesday 12 December 2012

- 9.00 - 10.00 Registration
10.00 - 10.30 Opening Session (Celebration Hall)
10.30 - 11.30 **Session 1: Invited Paper 1: Semantic Web and Ontology**
Chairman: Prof. Dr. I. Farag

An Over View of Web Intelligence

Prof. M. Adeeb Ghonaimy

*Computers and Systems Engineering Department, Faculty of
Engineering, Ain Shams University, Cairo, Egypt*

- 11.30 - 12.00 Coffee break

- 12.00 - 12.45 **Session 2: Invited Paper 2: Computational Linguistics**
Chairman: Prof. Dr. M. Adeeb Ghonaimy

آفاق اللغويات للسانيات الحاسوبية : مصادرها النظرية وغاياتها العملية
د.نبيل على
خبير اللغويات الحاسوبية

- 12.45 - 14.30 **Session 3: Language Engineering and Artificial Intelligence**
Chairman: Prof. Dr. Aly Aly Fahmy

1. A Proposed Semantic-Oriented Error Correction Model for Enhancing Arabic Sign Language Recognition

A.Sami Elons*, Magdy Aboul-ela**, M.F.Tolba*

* *Scientific Computing Department- faculty of Computers and
Information Sciences- Ain Shams University, Egypt*

** *Sadat Academy, Egypt,*

2. An Approach for Mining Opinions in Arabic Religious Decrees

Ahmed M. Misbah, Ibrahim F. Imam

*Computer Science Department, Faculty of Computing and
Information Technology, Arab Academy for Science, Technology
and Maritime Transport, Cairo, Egypt*

3. A Novel Association Rule-based Document Clustering Approach

Noha Negm*, Passent Elkafrawy*, Abd-Elbadeeh M. Salem**

* *Faculty of Science, Menoufia University*

** *Faculty of Computers and Information, Ain Shams University,
Cairo, Egypt.*

4. Knowledge Acquisition under Overlapping Classes A Rough Sets Approach

N. El-Ramly *, A. Kozea **, Passent Elkafrawy *, Ahmed Bakri

* Faculty of Science, Menoufia University, Egypt

** Department of Mathematics, Faculty of Science, Tanta University, Egypt

*** Institute of National Planning (INP), Cairo, Egypt.

14.30 - 15.30 Lunch

15.30 - 17.00 **Session 4: Room A: Large Corpora**

Chairman: Prof. Dr. Mohamad Zaki Abdel Mageed

1. أثر الصرف في بناء المعجم الحاسوبي العربي
يوسف أبو عامر* ، أ.د. وفاء كامل فايد* ، أ.د. علي فرغلي**
* كلية الآداب- جامعة القاهرة
** كلية الحاسبات و المعلومات- جامعة القاهرة
2. المعجم العربي الحديث
أحمد محمد متولي, مصطفى رمضان, حمدي سليمان مبارك
قسم أبحاث اللغة العربية - شركة صخر لبرامج الحاسب- القاهرة - جمهورية مصر العربية

15.30 - 17.00 **Session 5: Room B: Speech Processing and Recognition**

Chairman: Prof. Dr. Hassanein Al- Barhamtooshy

1. **A Baseline Speech Recognition System for Levantine Colloquial Arabic**
Mohamed Elmahdy*, Mark Hasegawa-Johnson**, Iman Mustafawi*
* Qatar University, Qatar
** University of Illinois, USA
2. **Implementing Speech recognition System on Android platform**
*Mostafa Abdallah El-Hosiny, *Mostafa AbdEl-Raheem Saad,
*Mostafa Magdy Montaser,
*Moataz Ahmed Lasheen, *Motaz Mostafa AbdEl-Halim
**Sherif Abdou, *Mohsen Rashwan
* Electronics and Electrical Communication Engineering Department, Cairo University, Egypt
** Information Technology Department, Faculty of Computers and Information, Cairo University, Egypt
3. **Speech Recognition System Based on Wavelet Transform and Artificial Neural Network**
Prof Ashrf H. Yahia*, El- Sayed A. El-Dahshan*, Engy R. Rady**
* Physics Department, Faculty of Science, Ain shams university, Cairo, Egypt
** Basic Science Department, Faculty of Computers and Information, Fayoum University, El Fayoum, Egypt

Thursday 13 December 2012

10.00 - 10.45 **Session 6: Room A: Machine Translation**

Chairman: Prof. Dr. Taghride Anbar

Corpus-based Approach for the Automatic Development of UNL Grammars

Sameh Alansary

Department of Phonetics and Linguistics, Faculty of Arts, Alexandria University, Alexandria, Egypt

**Bibliotheca Alexandrina, Alexandria, Egypt*

10.45 - 11.15 Coffee Break

11.15 - 12.00 **Session 7: Room A: Invited Paper 3: Social Networks and Contents development challenges**

Chairman: Prof. Dr. Fahmy Tolba

Topic Extraction and Sentiment Classification in Social Media

Prof. Dr. Ahmed Rafea

Computer Science and Engineering Department

School of Science and Engineering

American University in Cairo, Egypt

12.00 - 12.30 **Session 8: Room A: Optical Character Recognition**

Chairman: Prof. Dr. Hani Mahdi

Educating Illiterate People on Mobile Sets

*Doha Yousef, *Manar Ahmed, *Manal Ezzat, *Marwa Mamdouh, *Marwa Mohsen,

**Sherif Abdou, *Mohsen Rashwan

**Electronics and Electrical Communication Engineering Department, Cairo University, Egypt*

***Information Technology Department, Faculty of Computers and Information, Cairo University, Egypt*

12.30 - 14.00 **Session 9: Room A: Language Analysis and Comprehension:**

Chairman : Prof. Dr. Mohsen Rashwan

1. IAN: An Automatic Tool for Natural Language Analysis

Sameh Alansary*, Magdy Nagy**, Noha Adly**

**Department of Phonetics and Linguistics, Faculty of Arts, Alexandria University, Alexandria, Egypt*

**Bibliotheca Alexandrina, Alexandria, Egypt*

***Computers and Systems Engineering Department, Faculty of Engineering, Alexandria, Egypt*

2. Smoothing Techniques for Arabic Diacritics Restoration

Yasser Hifny

University of Helwan, Egypt

3. Higher Order n-gram Language Diacritics Restoration

Yasser Hifny

University of Helwan, Egypt

14.00 - 15.00 Lunch

12.30 - 14.00 **Session 10: Room B: Speech Analysis and Recognition II**

Chairman: Mohamad Younis Elhamalawy

1. Automatic Speech Recognitions Using Wavelet Packet Increased Resolution Best Tree Encoding

Amr M. Gody, Magdy Amer, Maha M. Adham, Eslam E. Elmaghraby

Department of Electrical Engineering, Faculty of Engineering, Fayoum University, Egypt

2. Automatic Speech Recognition of Arabic Phones Using Optimal-Depth-Split-Energy Best Tree Encoding

Amr M. Gody, Rania Ahmed Abul Seoud, Eslam E. Elmaghraby

Electrical Engineering, Faculty of Engineering, Fayoum University, Egypt

3. Adaptive English Pronunciation Errors for Arab Learners of English

Hassanin Al-Barhamtoshy*, Kamal Jambi*, Wajdi Al-Jedaibi*, Mohsen rashwan**, Sherif Abdou**

**Faculty of Computing & Information Technology, King Abdulaziz University, Saudi Arabia*

*** Faculty of Engineering, Cairo University*

*** Faculty of Computers at Cairo University Egypt*

15.00 - 16.00 **Session 11: Room A: NLP for Information Retrieval**

Chairman : Prof. Dr. Mohsen Rashwan

1. Arabic Stemming: A Corpus-Based Approach

Yasser Sabtan

English Department, Faculty of Languages and Translation, Al-Azhar University Nasr City, Cairo, Egypt

2. Improved Tokenization and POS Tagging for Arabic Text

Michael N. Nawar, Mahmoud N. Mahmoud, Magda B. Fayek

Computer Engineering Department, Faculty of Engineering, Cairo University, Egypt

16.30 - 17.00 **Session 12: Room A: Closing Session**

Chairman: Prof. Dr. Salwa Elramly



أعضاء الجمعية من المؤسسات

- 1- مركز نظم المعلومات – كلية الهندسة - جامعة عين شمس
- 2- معهد الدراسات والبحوث الإحصائية - جامعة القاهرة
- 3- مركز الحساب العلمي - جامعة عين شمس
- 4- الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري
- 5- أكاديمية أخبار اليوم
- 6- معهد بحوث الإلكترونيات
- 7- معهد تكنولوجيا المعلومات
- 8- مكتبة الإسكندرية
- 9- المعهد القومي للاتصالات (NTI)
- 10- الشركة الهندسية لتطوير نظم الحاسبات (RDI)
- 11- الهيئة القومية للاستشعار من بعد و علوم الفضاء
- 12- كلية الحاسبات و المعلومات جامعة قناة السويس
- 13- دار التأصيل للبحث و الترجمة

أهداف الجمعية

- 1- الاهتمام بمجال هندسة اللغويات مع التركيز على اللغة العربية بصفقتها لغتنا القومية والتركيز على قواعد البيانات المعجمية و صرفها ونحوها ودلالاتها بهدف الوصول إلى أنظمة آلية لترجمة النصوص من اللغات الأجنبية إلى اللغة العربية والعكس, وكذلك معالجة اللغة المنطوقة والتعرف عليها وتوليدها, ومعالجة الأنماط مع التركيز على اللغة المكتوبة بهدف إدخالها إلى الأجهزة الرقمية.
- 2- متابعة التطور في العلوم والمجالات المختصة بهندسة اللغة
- 3- التعاون مع الجمعيات العلمية المماثلة على المستوى المحلى والقومى والعالمى.
- 4- إنشاء قواعد بيانات عن البحوث التى سبق نشرها والنتائج التى تم التوصل إليها فى مجال هندسة اللغة بالإضافة إلى المراجع التى يمكن الرجوع إليها سواء فى اللغة العربية أو اللغات الأخرى.
- 5- إنشاء مجلة علمية دورية للجمعية ذات مستوى عال لنشر البحوث الخاصة بهندسة اللغة وكذلك بعض النشرات الدورية الإعلامية الأخرى بعد موافقة الجهات المختصة.
- 6- عقد ندوات لرفع الوعى فى مجال هندسة اللغة
- 7- تنظيم دورات تدريبية يستعان فيها بالمختصين وتتاح لكل من يهيمه الموضوع. وذلك من أجل تحسين أداء المشتغلين فى البحث لخلق لغة مشتركة للتفاهم بين الأعضاء
- 8- إنشاء مكتبة تتاح للمهتمين بالموضوع تشمل المراجع وأدوات البحث من برامج وخلافه.
- 9- خلق مجال للتعاون وتبادل المعلومات وذلك عن طريق تهيئة الفرصة لعمل بحوث مشتركة بين المشتغلين فى نفس الموضوعات.
- 10- تقييم المنتجات التجارية أو البحثية التى تتعرض لعملية ميكنة اللغة.
- 11- رصد الجوائز التشجيعية للجهود المتميزة فى مجالات هندسة اللغة.
- 12- إنشاء فروع للجمعية فى المحافظات.



المؤتمر الثاني عشر لهندسة اللغة
12-13 ديسمبر 2012

القاهرة- جمهورية مصر العربية

ينظم المؤتمر
الجمعية المصرية لهندسة اللغة

تحت رعاية

الأستاذ الدكتور/ حسين عيسى
رئيس جامعة عين شمس

الأستاذ الدكتور/ شريف حماد
عميد كلية الهندسة - جامعة عين شمس

رئيس المؤتمر
الأستاذ الدكتور/ محمد أديب رياض غنيمي
كلية الهندسة - جامعة عين شمس

مقرر المؤتمر
الأستاذ الدكتور / سلوى حسين الرملى
كلية الهندسة - جامعة عين شمس

مكان عقد المؤتمر : كلية الهندسة - جامعة عين شمس

<http://esole-eg.org>

Arabic Stemming: A Corpus-Based Approach

Yasser Sabtan

English Department, Faculty of Languages and Translation, Al-Azhar University

Nasr City, Cairo, Egypt

yasser_naguib@yahoo.com

Abstract— This paper presents a light stemmer for Arabic, using a corpus-based (or data-driven) approach. The current stemmer groups morphological variants of words in an Arabic corpus based on shared characters, before stripping off their affixes to produce their common stem. The aim of developing such a stemmer is to investigate the effectiveness of using word stems for extracting bilingual equivalents from an Arabic-English parallel corpus. Experimental results show that using Arabic word stems has significantly improved the accuracy score for bilingual lexicon extraction.

1 INTRODUCTION

Stemming has been widely used in a number of natural language processing (NLP) applications, such as information retrieval, machine translation, document classification and text analysis. Stemming is the process of reducing a word to its stem, base or root form after removing all of its affixes. This means that different morphological variants of a word can be conflated to a single representative form. For instance, *play*, *plays*, *played* and *playing* are grammatically conditioned variants of the base form "play". Stemming, thus, is an NLP task to conflate all word variants to a single form called the stem.

Morphological variants of words that are semantically similar are considered to belong to the same stem and to be equivalent for NLP purposes such as information retrieval, text analysis and bilingual lexicon extraction. Therefore, a number of stemming algorithms have been developed to group all words that have some semantic relation and reduce them to their stem. As far as Arabic is concerned, several stemming approaches, described below, have been proposed for achieving this goal. However, the effectiveness of most of these Arabic stemmers has been assessed in the framework of information retrieval [1], [2], [3], and text analysis [4]. In this paper we assess the effectiveness of using a word stem on the overall performance of a bilingual lexicon extraction method.

This paper presents a light stemmer, which removes word prefixes and suffixes, using a data-driven approach. The main aim of this stemmer is to group variant word-forms that are semantically related under one reduced form in our attempt to extract translation equivalents of open-class (or content) words from a bilingual parallel corpus. We use an undiacritized version of the Qur'anic text, written in Classical Arabic (CA), and its English translation rendered by Ghali [5] as our parallel corpus. The aim is to test our approach on such a corpus, with a view to be tested in future on any other type of corpus.

In fact, we have developed our own stemmer to be used among a number of other preprocessing steps before starting our main task of bilingual lexicon extraction. We have not used any of the other available stemmers such as Khoja [6], for instance, or other similar ones, because we aimed to do the whole task without using a lexicon. This lexicon-free approach has been adopted in all preprocessing tools; a stemmer, a part-of-speech (POS) tagger (described in detail in [7]) and a shallow dependency parser (as shown in [8]). This has the double advantage of investigating the effectiveness of different techniques without being distracted by the properties of the lexicon and at the same time saving much time and effort, since constructing a lexicon is time-consuming and labor-intensive. Thus, we use as little, if any, hand-coded information as possible. The accuracy score could be improved by adding hand-coded information. However, the point of the work reported here is to see how well one can do without any such manual intervention.

The basic assumption behind using stemming as a preprocessing step is that using word stems is expected to improve the accuracy of the lexicon extraction process. This is due to the fact that Arabic is morphologically rich where words contain numerous clitic items (conjunctions, prepositions and pronouns) attached to the stem. Thus, different Arabic words share the same stem. This stem in all similar word-forms is translated into the same English word, while the clitics have different corresponding words in English. For example, the word-forms *كتابه ktAbh* "his book"¹, *كتابها ktAbhA* "her book", *كتابك ktAbk* "your book" and *كتابهم ktAbhm* "their book", share the same stem (i.e. *كتاب ktAb*) with the same English equivalent "book". When these word variants are reduced to one representative form (i.e. the stem), the frequency of occurrence for this stem will be as high as that of the English target word and there will be a higher probability for choosing the right equivalent, since our automatic lexicon extraction method makes use of word co-occurrence frequencies in the parallel corpus. The automatic extraction method is described in detail in [8], but we will discuss it briefly in section 4.

¹ Throughout this paper, Arabic words are presented in the Arabic script followed by Buckwalter transliteration in italic and an English gloss in double quotes.

The remainder of this paper is organized as follows: in the following section we give a brief review of Arabic morphology and orthography, describe the used corpus, and discuss different approaches to Arabic stemming. Section 3 introduces the proposed method for stemming the Arabic corpus. In section 4 we present the evaluation criteria and the experimental results that were obtained for the stemming process and its effect on the main task of learning bilingual equivalents. Finally, in section 5 we conclude the paper with possible directions for future work.

2 BACKGROUND AND RELATED WORK

A. Arabic Morphology and Orthography

Arabic is a highly inflected language with a rich and complex morphological system, where words are explicitly marked for case, gender, number, definiteness, mood, person, voice, tense and other features [9]. The Arabic morphological system is generally considered to be of the non-concatenative type where morphemes are not combined sequentially, but root letters are interdigitated with patterns to form stems. A root is a sequence of mostly three or four consonants which are called radicals. The pattern, on the other hand, is represented by inserting a template of vowels in the slot within the root's consonants [10]. Thus, as McCarthy [11] points out, stems are formed by a derivational combination of a root morpheme and a vowel melody. The two are arranged according to canonical patterns. For example, the Arabic stem *katab* "(he) wrote" is composed of the root morpheme *ktb* "the notion of writing" and the vowel melody morpheme 'a-a'. The two are integrated according to the pattern CVCVC (C=consonant, V=vowel). This combination of root, pattern and vocalism is normally referred to as templatic morphemes. Thus, an Arabic word is constructed by first creating a word stem from templatic morphemes to which affixes are then added. Arabic word-forms are thus complex units which comprise the following:-

- **Proclitics**, which occur at the beginning of a word. These include mono-consonantal conjunctions (such as *w* "and", *f* "then"), prepositions (e.g. *b* "with" or "by", *l* "to")...etc.
- **Prefixes**. This category includes, for instance, the prefixes of the imperfective, e.g. *y*, prefixed morpheme of the 3rd person. It also includes the definite article *al* "the".
- **A stem**, which can be represented in terms of a ROOT and a PATTERN, as described above.
- **Suffixes**, such as verb endings, nominal cases, nominal feminine ending, plural markers ...etc.
- **Enclitics**, which occur at the end of a word. In Arabic enclitics are complement pronouns.

Table I below shows an Arabic word with a number of attached affixes.

TABLE I
AN EXAMPLE FOR A MORPHOLOGICALLY COMPLEX WORD IN ARABIC

Proclitic	Prefix	Root+Pattern (Stem)	Suffix	Enclitic
ل	ي	فاوض	ون	هم

As shown in this table, the Arabic word *lyfAwDwnhm* "to negotiate with them" contains a number of attached affixes that have corresponding words in English.

This rich morphology in Arabic makes morphological analysis a tough process. In Arabic very often a single word will consist of a stem with multiple fused affixes and clitics. Sometimes an Arabic word could stand as a complete sentence, as in *f>sqynAkmwh* "then we gave it to you to drink". This morphological richness is a source of an added increase in ambiguity that is a big challenge to Arabic NLP. For instance, the word *wjdnA* can be analyzed (among other analyses) as *wjad+nA* "we found" or as *wa+jad~+u+nA* "and our grandfather" [12]. In other words, this complex nature of Arabic morphology leads, in many cases, to internal word structure ambiguity. This means that a complex word could be segmented in different ways [13]. This is due to the fact that a number of clitics (prepositions, pronouns and conjunctions) may be attached to stems. For example, the word *kmAl* can be segmented in different ways, leading to different meanings. Thus, it can be *k+mAl* "as money", or *kmAl* "perfection". This word segmentation ambiguity is sometimes termed 'coincidental identity'. This occurs when clitics accidentally produce a word-form that is homographic with another full form word [14], [15].

A key feature of Arabic orthography is that it is normally written without diacritics or short vowels, which results in a great number of ambiguities and consequently represents a challenge for any NLP task [9]. This makes morphological analysis of the language very difficult. It is normally the case that a single written form may correspond to a number of different lexemes. For instance, the word-form *Elm* is composed of only three letters but has seven different readings, as shown in the following table.

TABLE II
 AMBIGUITY CAUSED BY THE LACK OF DIACRITICS

Arabic diacritized word	Meaning
عِلْمٌ <i>Eilomu</i>	knowledge
عَلَمٌ <i>Ealamu</i>	flag
عَلِمَ <i>Ealima</i>	knew
عُلِمَ <i>Eulima</i>	is known
عَلَّمَ <i>Eal~ama</i>	taught
عُلِّمَ <i>Eul~ima</i>	is taught
عَلِّمَ <i>Eal~im</i>	teach!

B. Description of the Corpus

As pointed out above, our main aim is to automatically learn translation lexicons from parallel corpora. We, thus, have to get a parallel corpus to be our resource for achieving this task. We use the Qur'anic text with an English translation [5] as our parallel corpus. We start with carrying out a number of preprocessing steps on this corpus: labeling words in the corpus with their POS tags, reducing word variants to one representative form (the stem), and labeling words with dependency relations for some basic constructions. Firstly, we will discuss the rationale behind choosing the Qur'anic text as our corpus and then shed light on some linguistic features of the corpus.

1) Reasons for Using the Current Corpus

As noted earlier, we adopt a lexicon-free approach in building all our modules: the preprocessing tools (the POS tagger, the stemmer, and the shallow parser) as well as the main tool of bilingual lexicon extraction. In this way, we minimize the resources required to achieve our task. Nonetheless, building a lexicon-free POS tagger for undiacritized Arabic, which is massively ambiguous, is not easy. Therefore, we had to start with a diacritized text to get the POS tagger off the ground. Then, in later stages we removed diacritics and ended up with a POS tagger for undiacritized text, as shown in [7]. This tagger has achieved 95% accuracy over a set of 15 tags. We then used the Arabic undiacritized text for all subsequent stages of processing, including the stemmer. We also needed an Arabic text with an available English translation. Hence, the reasons for using the Qur'anic text as our corpus can be succinctly summarized in the two following points:

- The need for an available Arabic-English parallel corpus.
- The need to start with a diacritized text in the early stage of the entire project.

The reason for removing diacritics from the corpus is to mimic the way Modern Standard Arabic (MSA) is written so that our approach could be extended to an MSA corpus. It should be noted that MSA is a simplified form of CA, and follows its grammar. According to Mubarak et al. [16], MSA tends to be simpler than CA in grammar usage, syntax structure, and morphological and semantic ambiguity.

2) Some Linguistic Features of the Corpus

The Qur'anic text is a small-sized corpus, containing 77,800 word tokens. The diacritized version of the corpus contains around 19,000 vowelized word-forms (or types), which are reduced to nearly 15,000 non-vowelized word types when diacritics are removed. Here are some of its main linguistic features:

- The Qur'anic text is composed of unpunctuated verses with mostly long sentences. A Qur'anic verse is one of the numbered subdivisions of a chapter in the Qur'an. A verse, which may reach up to 129 words, contains one or more sentences. There is no sentence boundary but only a verse marker that denotes the end of a verse.
- The Qur'anic text is characterized by many rhetorical devices, such as foregrounding and backgrounding, grammatical shift, idiomatic expressions, culture-bound items, and lexically compressed items where lengthy details of semantic features are compressed and encapsulated in a single word [17].

All these features make the current corpus a challenging type of text for any NLP task. This, consequently, refers to the robustness of the adopted approach, since our logical assumption is that experimenting with a less challenging corpus is expected to lead to improvement in accuracy scores.

C. Approaches to Arabic Stemming

Different approaches have been taken towards Arabic stemming. They can be summarized as follows:-

- Manually constructed dictionaries of words. This approach involves developing a set of lexicons of Arabic stems, prefixes and suffixes, with truth tables indicating legal combinations. In other words, each word uses a unique entry in a lookup table. In this technique, words could be stemmed via a table lookup.
- Light stemmers, which remove prefixes and suffixes. This approach, as the case in ours, refers to a process of stripping off a number of prefixes and suffixes, without any attempt to handle infixes, or recognize patterns and find roots. Light stemming can correctly conflate many morphological variants of words into large stem classes. However, it can fail to

conflate other forms that should be grouped together. For example, broken (or irregular) plurals for nouns do not get conflated with their singular forms. Examples of light stemmers include [1], [2], [4], [18].

- Morphological analyses which attempt to find roots based on the idea of pattern matching. The root is extracted after stripping off the affixes attached to a given word. Several morphological analyzers have been developed for Arabic, such as [6], [10], [19]. These analyzers find the root, or any number of possible roots for each word.
- Statistical stemmers, which group word variants using clustering techniques. In this technique, association measures between words are calculated based on shared unique N consecutive letters (i.e. the same shared root). Words that have a similarity above a predefined threshold are clustered and represented with only one word. This statistical method can provide a more language-independent approach to conflation [1]. De Roeck and Al-Fares [20], for instance, present a clustering algorithm for Arabic words to find classes sharing the same root. Their clustering was based on morphological similarity, using a string similarity metric after applying light stemming. Another class of statistical stemmers makes use of parallel corpora. Chen and Gey [3], for example, used a parallel English-Arabic corpus and an English stemmer to cluster Arabic words into stem classes based on their mappings to English stem classes.
- Hybrid stemmers, which make use of a combination of techniques. Goweder et al. [21], for example, propose a hybrid method for stemming Arabic, which uses light stemming, dictionaries and morphological analysis.

3 A PROPOSED METHOD FOR ARABIC STEMMING

This paper proposes a method for light stemming of Arabic, using a corpus-based approach. The current method groups morphological variants of words in the Arabic corpus and reduces them to their common stem. This grouping (or clustering) is based on shared characters between words. Having conditioned this character-string (or letter-sequence) similarity, a set of affixes (prefixes and suffixes) is removed from clustered words. This resource-frugal method makes use of only a number of inflectional and clitical affixes. It should be noted that clitics are included in affixes. So, proclitics and prefixes are classified under one category and enclitics are classified along with suffixes in the same category. This method is applied to the entire corpus in an iterative way. In other words, every word is compared with the other words in the corpus, and if there is similarity of at least three characters, the words in question are grouped and their attached affixes are removed to get the stem. Our approach to Arabic stemming is illustrated in the following figure.

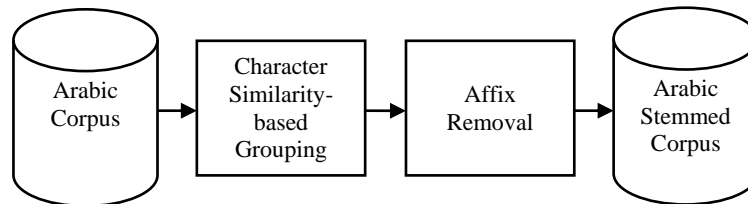


Figure 1: Corpus-based approach to Arabic stemming

For our main purpose of extracting translational equivalents from the parallel corpus we need to conflate similar words in the corpus into one reduced form so as to have a better chance of getting the right target language (TL) word. This is because Arabic is morphologically rich, where many morphological variants express the same semantic meaning of a lexical item. In addition, as noted before, since we rely on statistical information about the co-occurrence of words in the corpus to obtain the lexical equivalents, grouping similar words under one stem will increase the frequency of occurrence for such a stem and thus increase the chance of getting the TL word right.

The method we adopt to get an Arabic word stem comprises two steps. The first and second steps pertain to prefix and suffix removal respectively. We set a given threshold before removing all affixes: the obtained stem should be at least three characters. This covers all roots that contain at least three letters. In fact, biliteral roots are not covered, but they are not so common in comparison to other types of root. Also, we experimented with lowering the threshold to cover biliteral roots but this resulted in overstemming problems, where some semantically unrelated words that begin with the same letter are erroneously grouped under the same class. This occurs when the first letter is a part of a word but a prefix in another word. For example, فهم *fhm* "understood" could be mistakenly clustered with فهم *fhm* "then they". So, we increased the threshold to allow only roots with three letters or more, since they are the most common in the language.

The stemmer is applied to the entire corpus, including both content and function words. Nonetheless, function words are later excluded from the main task of lexicon building, since we focus on content words only. We use the POS tags associated with words in the corpus as a clue to exclude such function words. The 77,800 word tokens in the corpus are first collected in a list and then a dictionary is automatically constructed to contain the 15,000 undiacritized word-forms. Then we apply the two steps of prefix and suffix removal to this dictionary in order.

A. Step 1: Prefix Removal

In this stage words in the dictionary are compared with each other with regard to the final character. If the words in question end with the same character, the remaining characters are then checked to find a shared string. Then, if any of such words has an attached prefix, it is removed and thus the stem is obtained. This prefix removal occurs in case there are at least three characters in a given word. In this way all the letters in the word are retained except the attached prefixes. Figure 2 illustrates the way strings are matched based on their character similarity, starting with the final character, before stripping off attached prefixes.

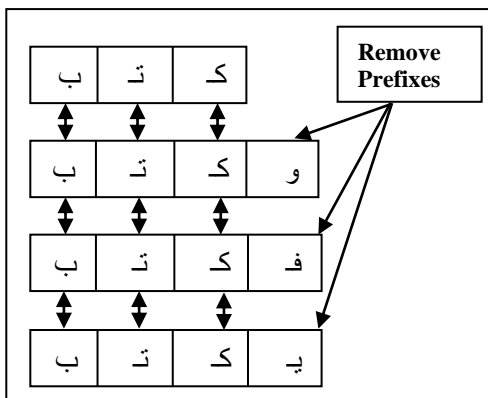


Figure 2: String matching and prefix removal

The Arabic prefixes that are removed from words are shown in the following table.

TABLE III
ARABIC PREFIXES AND THEIR MEANINGS

Prefix	Meaning
و <i>w</i>	conjunction (and)
ف <i>f</i>	conjunction (then)
أ > <i>A</i>	question particle (is it true that)
ب <i>b</i>	preposition (with, by, in)
ل <i>l</i>	preposition (to)
ك <i>k</i>	preposition (as)
ال <i>Al</i>	the definite article (the)
س <i>s</i>	future marker (will)
ي <i>y</i>	pres. tense (sing. masc.)
ت <i>t</i>	pres. tense (sing. fem.)
ن <i>n</i>	pres. tense (pl.)
أ <i>A</i>	imperative marker

It should be noted that some of the prefixes listed in the previous table may be attached to both nouns and verbs, such as the conjunctions and the question particle. Other prefixes are used with nouns only, such as prepositions and the definite article, while others are used with verbs only, such as the different tense markers. Moreover, those prefixes are classified into two sub-categories: the first category contains the proclitics, i.e. conjunctions, prepositions and the question particle, whereas the second category comprises the definite article and the tense markers.

All the prefixes in the previous table consist of one letter, except the definite article which contains two letters. Sometimes a combination of two or more prefixes is attached to a word. This may result in a prefix with three or more letters, as in *wAl* وال "and the" or *wbAl* وبال "and with the". We included such combinations in the list of prefixes that should be removed. Table IV below shows an example from the corpus, where some words are grouped based on letter-sequence similarity and then prefixes are removed to produce the stem.

TABLE IV
AN EXAMPLE FOR STEMMED WORDS WITH PREFIXES REMOVED

Clustered Words	Meaning	Removed Prefixes	Possible Stem
ختم <i>xtm</i>	sealed	----	ختم <i>xtm</i>
يختم <i>yxtm</i>	(he) seals	ي <i>y</i>	ختم <i>xtm</i>
وختم <i>wxtm</i>	and (he) sealed	و <i>w</i>	ختم <i>xtm</i>
نختم <i>nxtm</i>	(we) seal	ن <i>n</i>	ختم <i>xtm</i>

In this table the verbal word-forms ختم *xtm*, يختم *yxtm*, وختم *wxtm*, and نختم *nxtm* were grouped together, then prefixes were removed, resulting in the correct stem ختم *xtm*.

B. Step 2: Suffix Removal

In this stage words in the dictionary are compared with each other with regard to the initial character. If the words in question begin with the same character, the remaining characters are then checked to find a shared string. Then, if any of such words has an attached suffix, it is removed and thus the stem is obtained. This suffix removal occurs in case there are at least three characters in a given word. In this way all the letters in the word are retained except the attached suffixes. Figure 3 illustrates the way strings are matched based on their character similarity, starting with the initial character, before removing attached suffixes.

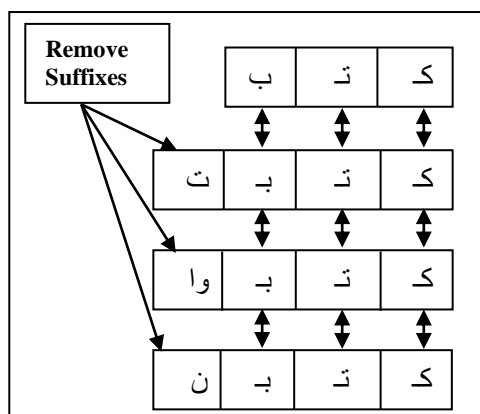


Figure 3: String matching and suffix removal

Table V below illustrates the Arabic suffixes that are removed from words.

TABLE V
ARABIC SUFFIXES AND THEIR MEANINGS

Suffix	Meaning	Suffix	Meaning
ة <i>p</i>	fem. marker	ي <i>y</i>	gen. pronoun (my)
ت <i>t</i>	sing. (masc.- fem.)	ني <i>ny</i>	obj. pronoun (me)
ان <i>An</i> , ا <i>A</i>	dual (masc.)	نا <i>nA</i>	(pl.) gen. or obj. pronoun (our, us)
تان <i>tAn</i> , تا <i>tA</i>	dual (fem.)	ك <i>k</i>	(sing.) gen. or obj. pronoun (your, you)
ون <i>wn</i> , ين <i>yn</i> , وا <i>wA</i>	plural (masc.)	كما <i>kmA</i>	(dual) gen. or obj. pronoun (your, you)
ات <i>At</i> , ن <i>n</i>	plural (fem.)	كم <i>km</i>	(masc. pl.) gen. or obj. pronoun (your, you)
تما <i>tmA</i>	dual (masc.- fem.)	كن <i>kn</i>	(fem. pl.) gen. or obj. pronoun (your, you)
تم <i>tm</i>	plural (masc.)	ه <i>h</i>	(masc. sing.) gen. or obj. pronoun (his, him)
تن <i>tn</i>	plural (fem.)	ها <i>hA</i>	(fem. sing.) gen. or obj. pronoun (her)
		هما <i>hmA</i>	(dual) gen. or obj. pronoun (their, them)
		هم <i>hm</i>	(masc. pl.) gen. or obj. pronoun (their, them)
		هن <i>hn</i>	(fem. pl.) gen. or obj. pronoun (their, them)

The previous table includes two kinds of suffixes: the first kind contains the number and gender markers for nouns and agreement markers for verbs, whereas the second kind comprises the enclitics (i.e. genitive and object pronouns). Genitive (or possessive) pronouns are attached to nouns, while object pronouns are attached to verbs.

As the case with prefixes, sometimes a combination of two suffixes is attached to a word. This may result in a suffix with three or more letters, such as *wnhm* as in *tktbwnhm* "(you) write them". We included such combinations in the list of suffixes that should be removed. Table VI shows an example from the corpus for the suffix removal of some clustered words based on character-string similarity matching.

TABLE VI
AN EXAMPLE FOR STEMMED WORDS WITH SUFFIXES REMOVED

Clustered Words	Meaning	Removed Suffixes	Possible Stem
أصاب > <i>SAb</i>	afflicted	----	أصاب > <i>SAb</i>
أصابها > <i>SAbhA</i>	afflicted (masc.) her/it	ها <i>hA</i>	أصاب > <i>SAb</i>
أصابت > <i>SAbt</i>	afflicted (fem.)	ت <i>t</i>	أصاب > <i>SAb</i>
أصابك > <i>SAbk</i>	afflicted you (sing.)	ك <i>k</i>	أصاب > <i>SAb</i>
أصابهم > <i>SAbhm</i>	afflicted them	هم <i>hm</i>	أصاب > <i>SAb</i>
أصابكم > <i>SAbkm</i>	afflicted you (pl.)	كم <i>km</i>	أصاب > <i>SAb</i>
أصابه > <i>SAbh</i>	afflicted him	ه <i>h</i>	أصاب > <i>SAb</i>

As can be noticed, a number of word variants for the base form *أصاب > SAb* "afflicted" were conflated to its shortest form, i.e. the stem, after suffixes were removed.

When there are variants for a given word, the stemmer conflates them to a reduced form. However, when there is a word-form in the corpus that has no related variants the word-form is not stemmed and remains as it is. For example, the word-form *مذعنين m*Enyn* "compliant" is the only form of its class that has occurred in the Qur'anic text and so the stemmer did not change it.

4 RESULTS AND DISCUSSION

As mentioned earlier, the purpose of developing such an Arabic stemmer is to investigate the effectiveness of using word stems on learning bilingual equivalents from a parallel Arabic-English corpus. Therefore, we are mainly interested in grouping word variants that are semantically related under one reduced form (i.e. the possible stem), whether the outputted form is the legitimate stem or not. So, firstly, we will evaluate the stemmer with regard to this point. Secondly, we will evaluate the stemmer's accuracy with regard to the percentage of grouped words in the test set that have been reduced to their legitimate stem. In this regard, we will discuss some of the problems that face the current stemmer. Finally, we will evaluate the effectiveness of using word stems on the bilingual lexicon extraction process.

As for the first evaluation, we use the following standard, shown in table VII, to measure the stemmer's accuracy.

TABLE VII: ARABIC STEMMER'S ACCURACY STANDARD

No.	Word-Forms	Meaning	Possible Stem	Hypotheses & Scoring
1	شاهد <i>\$Ahd</i>	witness	شاهد <i>\$Ahd</i>	1-2 (✓) 2-3 (✓) 3-5 (✓)
2	شاهدا <i>\$AhdA</i>	a witness	شاهد <i>\$Ahd</i>	1-3 (✓) 2-4 (✓) 4-5 (✓)
3	شاهدون <i>\$Ahdwn</i>	witnesses	شاهد <i>\$Ahd</i>	1-4 (✓) 2-5 (✓)
4	شاهدين <i>\$Ahdyn</i>	witnesses	شاهد <i>\$Ahd</i>	1-5 (✓) 3-4 (✓)
5	وشاهد <i>w\$Ahd</i>	and a witness	شاهد <i>\$Ahd</i>	
1	ماء <i>mA'</i>	water	ماء <i>mA'</i>	1-2 (✓) 2-3 (✓) 3-5 (✓)
2	بماء <i>bmA'</i>	with water	ماء <i>mA'</i>	1-3 (✓) 2-4 (✓) 3-6 (✗)
3	وماء <i>wmA'</i>	and water	ماء <i>mA'</i>	1-4 (✓) 2-5 (✓) 4-5 (✓)
4	كماء <i>kmA'</i>	as water	ماء <i>mA'</i>	1-5 (✓) 2-6 (✗) 4-6 (✗)
5	الماء <i>AlmA'</i>	the water	ماء <i>mA'</i>	1-6 (✗) 3-4 (✓) 5-6 (✗)
6	سماء <i>smA'</i>	heaven	ماء <i>mA'</i>	

As the previous table shows, we set a number of hypotheses for scoring the relatedness of clustered words. So, the hypothesis 1-2, for example, checks whether the first and second words in a given group are semantically related. If so, they are correctly grouped and are thus scored. If they are unrelated, they are considered wrong and are not scored. Accordingly, in the first example all combinations are correctly grouped because they are all related. But in the second example the final word is unrelated to all the other five words and is not scored with them. The Arabic stemmer has achieved 86% accuracy when tested on a random set of 200 words, comprising about 800 hypotheses. The remaining 14% of words in the test set have been wrongly grouped, where words are not semantically related, though they may be conflated under the correct stem. For example, الذهب *Al*hb* "gold" has been conflated with different word-forms for the verb ذهب **hb* "to go" under the reduced form ذهب **hb*.

Although this reduced form is the correct stem for both the noun and the verb, they are not scored because they are semantically unrelated. As for the second evaluation, 72.2% of the words in the test set were reduced to their legitimate stem. So, we have two related evaluations here: the first one, which is of more interest to us for our main task, is concerned with grouping semantically related words under a reduced form (which may be the actual stem or not). The score obtained for this evaluation, based on the criteria outlined in table VII above, is 86%. The second evaluation is concerned with the percentage of grouped words in the test set that were reduced to their actual stem. In this respect we got 72.2% accuracy. The stemmer's errors are due to a number of reasons which are discussed below.

Broadly speaking, stemmers make two types of errors. Strong stemmers tend to form larger stem classes in which unrelated forms are erroneously conflated, while weak stemmers fail to conflate related forms that should be grouped together. Most stemmers fall between these two extremes and make both types of errors [1]. There are a number of errors made by our stemming algorithm, which can be classified into different types as shown in the following table.

TABLE VIII: TYPES OF ERRORS PRODUCED BY THE STEMMER

Word-Form	Actual Stem	Produced Stem	Error Type
سماء <i>smA'</i>	سماء <i>smA'</i>	ماء <i>mA'</i>	overstemming
نقول <i>nqwl</i>	قال <i>qAl</i>	قول <i>qwl</i>	spelling
يظنون <i>yZnwn</i>	ظن <i>Zn</i>	يظن <i>yZn</i>	understemming
ربه <i>rbh</i>	رب <i>rb</i>	ربه <i>rbh</i>	unchanged
شركاءكم <i>\$rka'km</i>	شريك <i>\$ryk</i>	شركاء <i>\$rka'</i>	broken plural case

The errors listed in the previous table were produced by the stemmer due to a number of reasons. The first word سماء *smA'* "heaven" was stemmed wrongly because the first letter is similar to the future marker prefix. It is, thus, wrongly grouped with the word ماء *mA'* "water", causing an overstemming problem. However, when the word is used in the definite case, i.e. السماء *AlsmA'*, it is stemmed correctly. As for the second word in the table, the spelling of the produced stem is not correct. The current phase of the stemmer does not have rules for handling orthographic alternations, which causes such spelling errors. The produced stem for the third word still has an attached prefix. This understemming problem occurs because the stemmer truncates affixes when they are attached to words with three or more letters, as stated earlier. This condition causes such a type of error but evades other errors. This condition is also the reason for the error in the fourth word, where the produced stem has the same shape as the cliticized word-form. Finally, the last word in the table is a broken plural case. The broken plural is made from the singular through infixes and patterns. Due to such internal differences light stemmers normally fail to conflate broken plurals with their singular forms. In future we will investigate ways to reduce such errors.

As for the final evaluation, we will show the effectiveness of using word stems on the performance of an automatic method for bilingual lexicon extraction of content words from a parallel corpus. The automatic extraction method, described in detail in [8], is based on the following principle:

- For each sentence-pair, each word of the target language (TL) sentence is a candidate translation for each word of the aligned source language (SL) sentence.

This principle means that (S, T) is a candidate if T appears in the translation of a sentence containing S. Following the above principle we compute the absolute frequency (the number of occurrences) of each word in the SL and TL sentences, giving preference to the target words that have the highest score in the TL sentences that correspond to the SL sentences. In addition, we take into account the relative distance between SL and TL words in their specific contexts, producing our algorithm for bilingual lexicon extraction. This method is applied to both raw and POS-tagged texts. However, in case of POS-tagged texts we add the following constraint:

- A chosen TL candidate for a given SL word must have the same POS tag as that of the SL word.

This notion of matching equivalents based on similarity of their POS tags is also emphasized by Melamed [22] in the following statement:

".....word pairs that are good translations of each other are likely to be the same parts of speech in their respective languages."

The Arabic corpus is tagged using our lexicon-free POS tagger [7]. As for the English corpus, we use an English POS tagger that was developed using the same lexicon-free approach. The English tagger, which is based on the BNC basic (C5) tagset with some modifications, is described in [8].

The extraction method was tested on both raw and POS-tagged texts. F-measure, which computes both precision and recall, has been used to evaluate the lexicon extraction method. The F-measure can be defined as follows:

$$F\text{-Measure} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

Precision and recall scores for candidate item Y with respect to reference item X are calculated according to equations (2) and (3) respectively.

$$\text{Precision (Y|X)} = \frac{|X \cap Y|}{|Y|} \quad (2)$$

$$\text{Recall (Y|X)} = \frac{|X \cap Y|}{|X|} \quad (3)$$

In our evaluation framework precision can be simply defined as the number of correct translations out of the total number of the output. Recall, on the other hand, is defined as the number of correct translations out of the total number of the words that should have been translated. We evaluate an extracted lexicon with regard to the top translation candidate. Other candidates that occupy any other position in an extracted lexicon are not scored, even though they may include the correct equivalent. We use a random set of 100 content words to evaluate extracted lexicons. The English translation in the parallel corpus is used as our reference translation (i.e. gold standard) for scoring the output. Different scores have been obtained for stemmed and unstemmed Arabic with both raw and POS-tagged texts as shown in the following table.

TABLE IX: BILINGUAL LEXICON EXTRACTION SCORES

Type of Corpus	Type of Text	Precision	Recall	F-score
Raw Corpus	Unstemmed Arabic	0.353	0.29	0.318
	Stemmed Arabic	0.516	0.465	0.489
POS-tagged Corpus	Unstemmed Arabic	0.459	0.455	0.457
	Stemmed Arabic	0.623	0.605	0.614

It is obvious that using Arabic word stems has improved the lexicon extraction score for both raw and POS-tagged texts. But using a POS-tagged corpus has resulted in a better score for both stemmed and unstemmed Arabic, with the best score being achieved for stemmed Arabic. The accuracy score for lexicon extraction is expected to increase if the error rate of both the stemmer and the tagger is reduced. This is one of the tasks we aim to carry out in our future work.

5 CONCLUSIONS

Stemming is important for highly inflected languages such as Arabic for many NLP applications that require the stem of a word. This paper presented a corpus-based method for Arabic stemming, which attempts to get a word stem after grouping word variants in the corpus based on shared characters and removing their common affixes. Most existing Arabic stemmers have been assessed in the framework of information retrieval. In our work we assess the effectiveness of using a word stem on the overall performance of a bilingual lexicon extraction method. With this goal in mind, we aimed at grouping word variants that share the same meaning in an attempt to improve the lexicon extraction process. The results show that 86% of words in the test set were correctly grouped under a similar reduced form (i.e. the possible stem). In some cases the reduced form is not the legitimate stem. The evaluation shows that 72.2% of the words in the test set were reduced to their legitimate stem. The proposed method generates some errors, which are classified into different types (e.g. overstemming, understemming, and spelling). As regards the bilingual lexicon extraction, using Arabic word stems has significantly improved the extraction process for both raw and POS-tagged texts. The best score has been achieved for POS-tagged texts, with an F-score of 0.614, where stemming has increased the score nearly 0.16. In future we will work on enhancing the stemmer by trying to reduce its error rate. In addition, we plan to improve the lexicon extraction by using a number of bootstrapping techniques through making use of dependency relations between words in the parallel corpus.

ACKNOWLEDGMENT

The author would like to thank Prof. Allan Ramsay in the School of Computer Science, University of Manchester, UK for important suggestions and helpful discussions.

REFERENCES

- [1] L. Larkey, L. Ballesteros and M. Connell, "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis," in *Proc. of the 25th International Conference on Research and Development (SIGIR)*, pp. 275-282, Tampere, Finland: ACM, 2002.

- [2] M. Aljlayl and O. Frieder, "On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach," in *Proc. of the Eleventh Conference on Information and Knowledge Management (CIKM'02)*, pp. 340 - 347, VA, USA, 2002.
- [3] A. Chen and F. Gey, "Building an Arabic Stemmer for Information Retrieval," in *Proc. of the Eleventh Text REtrieval Conference (TREC 2002)*, National Institute of Standards and Technology, November, 2002.
- [4] N. Thabet, "Stemming the Qur'an," in *Proc. of the Workshop on Computational Approaches to Arabic Script-Based Languages, COLING-04*, pp. 85-88, Geneva, Switzerland, 2004.
- [5] M. M. Ghali, *Towards Understanding the Ever-Glorious Qur'an*, 4th ed., Daar al-nashr lil-Jaami'at, Cairo, Egypt, 2005.
- [6] S. Khoja, "Stemming Arabic Text," Computing Department, Lancaster University, Lancaster, UK, 1999.
- [7] A. Ramsay and Y. Sabtan, "Bootstrapping a Lexicon-Free Tagger for Arabic," in *Proc. of the 9th Conference on Language Engineering (ESOLEC'2009)*, pp. 202-215, Cairo, Egypt, 23-24 December 2009.
- [8] Y. Sabtan, "Lexical Selection for Machine Translation," Doctoral dissertation, University of Manchester, UK, 2011.
- [9] M. Maamouri, A. Bies and S. Kulick, "Diacritization: A Challenge to Arabic Treebank Annotation and Parsing," in *Proc. of The Challenge of Arabic for NLP/MT Conference*, pp. 35-47, The British Computer Society, London, UK, 2006.
- [10] K. R. Beesley, "Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001," in *Proc. of The ACL Workshop on Arabic Language Processing: Status and Prospects*, pp. 1-8, Toulouse, France, 2001.
- [11] J. McCarthy, "A Prosodic Theory of Nonconcatenative Morphology," *Linguistic Inquiry*, vol. 12, no. 3, pp. 373-418, 1981.
- [12] I. M. Saleh and N. Habash, "Automatic Extraction of Lemma-based Bilingual Dictionaries for Morphologically Rich Languages," in *Proc. of The 3rd Workshop on Computational Approaches to Arabic Script-based languages at the MT Summit XII*, Ottawa, Ontario, Canada, 2009.
- [13] A. Farghaly and K. Shaalan, "Arabic Natural Language Processing: Challenges and Solutions," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 8, no. 4, pp. 1-22, 2009.
- [14] D. Kamir, N. Soreq and Y. Neeman, "A Comprehensive NLP System for Modern Standard Arabic and Modern Hebrew," in *Proc. of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, pp. 1-9, Philadelphia, PA, USA, 2002.
- [15] M. A. Attia, "An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks," in *Proc. of The Challenge of Arabic for NLP/MT Conference*, pp. 48-67, The British Computer Society, London, UK, October 2006.
- [16] H. Mubarak, A. Metwally and M. Ramadan, "Analyzing Arabic Diacritization Errors of MADA and Sakhr Diacritizer," in *Proc. of the 11th Conference on Language Engineering (ESOLEC'2011)*, 14-15 December 2011, Cairo, Egypt.
- [17] H. Abdul-Raof, *Qur'an Translation: Discourse, Texture and Exegesis*, London and New York: Routledge, 2001.
- [18] K. Darwish, "Building a Shallow Arabic Morphological Analyzer in One Day," in *Proc. of ACL 2002 Workshop on Computational Approaches to Semitic languages*, July 11, 2002.
- [19] T. Buckwalter, "Buckwalter Arabic Morphological Analyzer Version 1.0," Linguistic Data Consortium. Catalog number LDC2002L49, and ISBN 1-58563-257-0, 2002.
- [20] A. N. De Roeck and W. Al-Fares, "A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots," in *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 199-206, Hong Kong, 2000.
- [21] A. Goweder, H. A. Alhammi, T. Rashid and A. Musrati, "A Hybrid Method for Stemming Arabic Text," in *Proc. of The 2008 International Arab Conference on Information Technology (ACIT'2008)*, Tunisia, 2008.
- [22] I. D. Melamed, "Automatic Evaluation and Uniform Filter Cascades for Inducing N-Best Translation Lexicons," in *Proc. of the 3rd Workshop on Very Large Corpora (WVLC3)*, pp. 184-198, Boston, MA, U.S.A, 1995.

Topic Extraction and Sentiment Classification in Social Media

Ahmed Rafea

*Computer Science and Engineering Department
School of Science and Engineering
American University in Cairo
rafea@aucegypt.edu*

Abstract— Social networks have become the most important source of news and people’s feedback and opinion about almost every daily topic. With this massive amount of information over the web from different social networks like Twitter, Facebook, Blogs, etc, there has to be an automatic tool that can determine the topics that people are talking about and what are there sentiments about these topics. This need has led to the introduction of an area of research in Text Mining called Sentiment Classification. The techniques used in this area, can be mainly divided into two approaches: machine learning (ML) and semantic orientation (SO). Machine learning algorithms have succeeded in classic text categorization, and so they were implemented for sentiment classification, but with having the target classes as “positive” and “negative”. The goal of the research described in this paper was to develop a prototype that can "feel" the pulse of the Arabic users with regards to a certain hot topic. The paper presents our experience in extracting Arabic hot topics from Twitter and classifying the sentiment of these tweets. Using unigram words that occurred more than 20 times in the whole corpus as features for using bisecting k-mean clustering algorithm, resulted in purity of 0.704 and entropy of 0.275. The score generated for the quality of the generated topic was 72.5%. A lexicon of approximately 1600 sentiment words for Egyptian dialect and their weights for their polarities as positive or negative was built to develop an unsupervised classifier using Semantic Oriented approach. Comparing the results obtained in the SO experiment with the ML experiment, related to the accuracy measure, has revealed that the accuracy of the SVM learning algorithm was 73.25% while the accuracy of the SO approach was 69.5%.

1. INTRODUCTION

Social networks have become the most important source of news and people’s feedback and opinion about almost every daily topic. Analyzing data from social networks is the key to know what people think or up to about certain topics. In general, the two main tasks that interest many researchers are to extract hot topics from social media and to classify the sentiment of a text describing the position of its author toward one of these hot topics.

With this massive amount of information over the web from different social networks like Twitter, Facebook, Blogs, etc, there has to be an automatic tool that can determine what people are talking about in certain locations and over certain period of time. Several researches have been done with different approaches. A lot of them achieved good performance regarding topic detection, which is basically grouping (clustering) similar data together indicating they are about the same topic. Topic extraction can be considered a next step whose goal is to label these groups through extracting a topic title for every group of data.

Sentiment classification techniques can be mainly divided into two approaches: machine learning (ML) and semantic orientation (SO). Machine learning algorithms have succeeded in classic text categorization, and so they were implemented for sentiment classification, but with having the target classes as “positive” and “negative”. On the other hand, in the SO approach, a sentiment lexicon is built either manually, semi-automatically or automatically with each word having its semantic intensity as a number indicating whether it is positive or negative as well as its intensity. Then, this lexicon is used to extract all sentiment words from the document and sum up their polarities to determine if the document is holding an overall positive or negative sentiment besides its intensity.

The goal of the research described in this paper was to develop a prototype that can "feel" the pulse of the Arabic users with regards to a certain hot topic. The paper presents our experience in extracting Arabic hot topics from Twitter and classifying the sentiment of these tweets. The second section describes the work related to different approaches of topic extraction and sentiment classification. The third section presents the approach used for topic extraction and the preliminary results obtained. The fourth section explains the approach used for sentiment classification and the preliminary results obtained. The fifth section concludes our findings and proposes future work needed.

2. RELATED WORK

In this section we will present works related to topic extraction and sentiment classification approaches.

A. Topic Extraction Approaches

The idea of this research domain has originated back in the 1990's with a project called TDT (Topic Detection and Tracking). The basic idea was originated in 1996 when the Defense Advanced Research Projects Agency (DARPA) realized that there is a technological need to determine the topical structure of news streams without human intervention [4]. Topic detection is the problem of identifying stories in several continuous news streams that pertain to new or previously unidentified events. It involves detecting the occurrence of a new event such as a plane crash, a murder, a jury trial result, or a political scandal in a stream of news stories from multiple sources. Clustering a group of news items, blogs or tweets and then discovering the labels of these clusters mainly achieve this task. These clusters labels are actually the topics extracted from this group of news items, blogs, or tweets.

1) Clustering

Seo & Sycara [43] used hierarchical clustering for text clustering for topic detection. Dai & Sun [13] used agglomerative clustering with time decay to identify events in news. Dai et al. [12] improved the agglomerative hierarchical clustering algorithm based on the average link method for online topic detection and tracking of financial news. Young-dong et al. [61] used hierarchical agglomerative clustering technique for text hierarchical topic identification algorithm based on the dynamic diverse thresholds clustering. Huang & Cardenas [23] used hierarchical agglomerative method to group articles into clusters of same events. Their work aimed at extracting hot events from news feeds. Okamoto & Kikuchi [34] used agglomerative clustering for topic extraction from blog entries within a neighborhood. Qiu et al. [39] used K-mean algorithm for topic detection while Wartena & Brussee, [55] used the induced bisecting k-mean algorithm for their experiment in topic detection by clustering key words of documents. Zhang et al. [63] discussed using incremental clustering for automatic topic detection. They proposed a new topic detection method called TPIC that adds the aging nature of topics to pre-cluster stories.

2) Labeling Approaches

In order to extract the topic described by a cluster, key phrase extraction techniques are mainly used to do that. Generally key-phrase extraction techniques can be categorized into simple statistics, linguistic, and machine learning.

Tomokiyo & Hurst [49] used the statistical language model in their work. El-Beltagy & Rafea [16] developed a system called KP-Miner. It extracts key phrases from English and Arabic texts using heuristics rules and statistics. Huang & Cardenas [23] in their work they relied on extracting hot events from news feeds. The cluster with more hot terms or with high weighted hot terms is examined for hot terms. The study presented by Huafeng et al [22] discussed the optimization design of subject indexing. Their work is based on the word frequency statistics. They took into consideration the word length, position and frequency in the weighting coefficient of the word. They considered long words as more specialized and short words are more generic. TFPDF algorithm is used to recognize the terms that try to explain the main topics [26]. This algorithm is designed to assign heavy term weight to these kinds of terms and thus reveal the main topics. Cataldi et al [10] tackled extracting emerging topics on Twitter based on temporal and social terms evaluation.

Jain & Pareek, [24] used part of speech tagging in their work, formatting features and position of words in their work. Their results achieved high matching against the annotated data. Wang et al, [54] used semantic information for automatic key phrase extraction in their work. Wang et al. [52] developed an automatic online news topic key phrase extraction system; they combined TDT algorithms with aging theory for topic detection and tracking. Lopez et al. [28] worked on automatic titling of electronic document with noun phrase extraction. It is based on the morpho-syntactic study of human written titles in a corpus of various texts.

Witten et al. [58] developed KEA, which is a tool for key-phrase extraction. It identifies candidate key phrases using feature values vector for each candidate, and uses a machine-learning algorithm to predict which candidates are good key phrases. Sarmiento & Nunes [41] developed a tool called "verbatim" which is available online that automatically extracts quotes and topics from news feeds. They used classification using support vector machines SVM, and Rocchio classifier. The challenges they face that the topic titles are so generic and news streams aren't consistent in using the same name for describing the same event.

B. Sentiment Classification

In supervised ML, a piece of text is converted into a feature vector so that the classifier would learn from a set of data labeled with its class (called training data) that a combination of specific features yields a specific class, and then it can test its accuracy of learning on another set of data (called testing data). On the other hand, in the SO approach, a sentiment lexicon is built either manually, semi-automatically or automatically with each word having its semantic intensity as a number indicating whether it is positive or negative as well as its intensity. Then, this lexicon is used to extract all sentiment words from the document and sum up their polarities to determine if the document is holding an overall positive or negative sentiment besides its intensity.

1) *The Machine Learning Approach*

Machine learning algorithms are discussed in detail in [42]. Supervised ML algorithms, such as Support Vector Machines (SVM), Naive Bayes (NB) and Maximum Entropy (ME) have been used extensively in the sentiment analysis research [27], [37]. In the ML approach, each document is represented as a feature vector with representative features for the target class. Various feature sets have been tried specifically for sentiment classification, as discussed in [37]. Some of the significant features that are related to our work are:

N-grams: N-grams are common features to use in text classification. Many researchers have used n-grams, especially unigrams, since they result in a high accuracy and they added other features as improvements to their systems [35]; [56]; [32]; [1]; [2]; [14]. Some researchers use unigrams and bigrams [14], while others use unigram, bigram and trigrams [2]; [14]. Several ML algorithms have been compared, and SVM outperformed other algorithms [35] and hence, it has been used as a common algorithm for sentiment classification.

Part-Of-Speech tag n-grams: Part-Of-Speech (POS) tag n-grams have been proven to be good indicators of sentiment [19]; [18], [2]. For instance, the authors in [18] experimented the effect of a combination of positive and negative nouns, verbs, adverbs and nouns and have shown that the appearance of a positive adjective followed by a noun is more frequent in positive documents than in negative ones, and that the appearance of a negative adjective followed by a noun is more frequent in negative documents.

Stylistic features: These include lexical and structural attributes as well as punctuation marks and function words, as explained in [1]; [2]. The lexical features include character- or word-based statistical measures of word variation. Examples of the character-based lexical features are: 1) Total number of characters; 2) Characters per sentence; 3) Characters per word; and 4) The usage frequency of individual letters. Some examples of the word-based lexical features are: 1) Total number of words; 2) Words per sentence; 3) Word length distribution; and 4) Vocabulary richness measures; such as: the number of words that occur only once in the document (also called hapax legemona) and the number of words that occur twice in the document (also called hapax dislegemona). The structural features include text organization and layout, for example, signatures, number of paragraphs, average paragraph length, total number of sentences per paragraph and others. These features were used along with other features in sentiment classification research [1]; [2]; [14]; [3] and they improved the system's accuracy when added [2].

Negation and modification features: These are two of the important categories of the contextual valence shifters. Early work in sentiment classification did not investigate the effect of negation or modifiers [35], as the authors only used Bag-Of-Words (BOW) and higher-order n-grams. As a result, two sentences like these: "I like this movie" and "I don't like this movie" would be similar to each other since both contain the verb "like", although the first one holds a positive sentiment while the second holds a negative sentiment [37]. Modifiers also play a crucial role when classifying sentiment since they can increase or decrease the semantic intensity of polar terms or even shift them towards the positive or negative sentiment. Therefore, later researches focused on how to detect and extract negation and modifiers and add them as features [59]; [46]; [25]; [47]; [9]. For instance, the authors focusing on sentence-level sentiment classification in [59] added a binary feature for each word token to determine whether it was negated or not in addition to other features to indicate whether it was modified by an adjective or adverb intensifier or one or more contextual valence shifters that they extracted: general, negative and positive polarity shifters. The general polarity shifter reverses the polarity of the word modified by it, e.g. little truth. The negative polarity shifter yields an overall negative sentiment for the overall expression, e.g. lack of wisdom. The positive polarity shifter changes the overall sentiment of the expression to positive; for example, abate the damage [59]. In addition, the authors in [25] used a dependency parser to extract typed dependencies and developed special rules to determine the scope of each negation term. Furthermore, the authors in [9] used typed dependencies and a negation and quantifiers lists to extract negated, amplified and down-toned sentiment words and build a sentence-level polarity and intensity sentiment classifier.

Dependency relations: Some researchers have explored the effect of dependency relations in sentiment classification, since they can hold more information than neighboring words (such as: higher-order n-grams). Dependency relations are typically extracted using a dependency parser. For example, the authors in [59] used a dependency parser to extract modifiers and other dependency relations (e.g. words connected by a conjunction, like and). It was shown in [32] that adding dependency relations with focus on adjectives preceded by nouns and nouns that have a dependency relation with polar terms to n-grams did not improve the performance.

Researchers have tried using different ML algorithms to compare their performance in the sentiment classification task. It was shown in [35] that SVM outperforms both NB and ME when using unigrams, bigrams as well as other features; and hence, most research has used SVM and it became the default algorithm in sentiment classification.

Another problem in machine learning besides choosing the right features is feature selection. Since there can be redundant or irrelevant features in the feature vector that makes it unnecessarily a huge vector, feature selection is performed to reduce the dimensionality of the feature vector so that only representative features for the target class are remaining. By selecting the most relevant features for the target concept, the classifier learns better which class label to give to each vector. In addition, it reduces the running time required for the classifier on the training data, which is crucial for real-world applications [15]. Several researchers have implemented different feature selection methods that were used for classic text classification and applied them to sentiment classification [2], [14]. Other researchers have developed new algorithms for feature selection specified for sentiment classification [53], [33], [3]. But, most research uses the Information Gain (IG) heuristic as the feature selection method due to its reported effectiveness [2]; [14]; [20].

2) *The Semantic Orientation Approach*

In the SO approach, a document's polarity is calculated as the sum of its polar terms and/or expressions. Early work in this direction calculated the phrase's polarity as the difference between the Point Mutual Information (PMI) between the phrase and the word "excellent" as representative for the positive class and the PMI between the phrase and the word "poor" as representative for the negative class, with hit counts from AltaVista search engine using the NEAR operator [50]. Some researchers built a sentiment lexicon manually, such as: the General Inquirer1 (GI) and SO-CALculator (SO-CAL) [8], [48]. Others used semi-supervised and supervised techniques to build a lexicon either from a small seed list of positive and negative words or an already-existing online dictionary, like the Subjectivity dictionary and SentiWordNet [59]; [17]; [6]. The Subjectivity dictionary contains a list of about 8,000 words labeled with their corresponding prior polarity (positive, negative, neutral or both), type (strong (or weak) subjective (or objective)) and part of speech. Senti-WordNet contains a list of all words from WordNet [30] with each Part-Of-Speech having 3 types (objective, positive and negative) and each type having a score describing its intensity (from 0 to 1). The GI and SentiWordNet deal with individual words only as holding positive or negative sentiment. On the other hand, the Subjectivity dictionary is accompanied with two lists of intensifiers and contextual valence shifters to extract the contextual polarity instead of the prior polarity of the individual words, but this dictionary was built for sentence-level sentiment classification using a supervised approach [59]. Also, SO-CAL tries to capture the contextual polarity of the SO-carrying words in the document by creating separate lists of intensifiers (divided into amplifiers and down-toners), negation expressions, irrealis markers (words whose appearance results in the ignorance of the SO of polar words in the sentence, such as: modals and conditional markers) beside their lists of SO-carrying nouns, verbs, adverbs and adjectives [48].

3) *Hybrid Approaches*

There are some significant differences between the machine learning and semantic orientation approaches. First of all, the ML approach performs better than the SO approach on a single domain; the highest accuracy achieved on the Polarity dataset version 2.0 [36] using a classifier was 91.7% [2] versus an accuracy of 76.37% using SO-CAL [48]. Secondly, the classifier can learn domain-specific polarity; e.g. "long battery life" (+ve) vs "long time to focus" (-ve), unlike the lexicon where a prior polarity is known for each word. However, to build a high accuracy classifier, we need to have a huge corpus labeled with its class (positive or negative) whereas a dictionary does not need one. This can be a labor-intensive task; to collect data from different genres (reviews, news articles, blogs . . . etc) and domains (movies, products, politics . . . etc) and label them manually. In addition, the authors in [8]; [48] argued that a ML approach does not take linguistic context into account, such as negation and intensification, since there can be three features "good", "very good" and "not good" but the classifier does not know they are related to each other.

Since both the machine learning and semantic orientation approaches have some advantages and disadvantages, some researchers have attempted to combine them together to benefit from the advantages of each approach [5]; [39]; [40]; [20]; [14]. For example, the authors in [20] built a classifier with unigrams and bigrams with a threshold of 5 and stylistic features and they developed a feature-calculation strategy to extract sentiment features (verbs, adjectives and adverbs) from SentiWordNet 3.0. Similarly, the authors in [5] tried to develop a high-accuracy domain-independent system at the sentence level by building an ensemble model of two classifiers; one with unigrams, bigrams and trigrams as features and trained on in-domain data, and the other one with sentiment words from both WordNet and the General Inquirer as features. The authors in [40] then improved this system to be self-supervised so that it does not need labeled data. They developed a two-phase model; the first one is a sentiment lexicon and a negation list to label the data and they took the high-confidence labeled documents as training data to a classifier with sentiment words being the feature set. Another technique to solve the problem of manually labeling the data was shown in [20] where the authors trained an initial classifier with sentiment features from the Subjectivity lexicon. Then, from

the high-confidence labeled data, they extracted the most indicative features for each class (using the Information Gain heuristic) as self-learned features to train another classifier instead of training it with self-labeled data. However, all these models did not take into account contextual valence shifters.

3. A PROPOSED APPROACH FOR EXTRACTING TOPICS OF ARABIC TWEETS

The task of topic detection and extraction consists of several phases and the most important phase that can be considered the core of this task is clustering. We have conducted preliminary experiments to decide on the number of clusters to get the best purity, the features to be used and the labeling approach. The following subsection will describe the results obtained.

A. Determine the Number of Clusters

There are many clustering algorithms used for clustering textual data. One of the most used algorithms for clustering is bisecting-k-mean. Determining the number of clusters is important in many clustering algorithms. We did that by investigating to what extent the number of clusters will have impact on the purity of the clusters.

We collected 110 tweets over a span of 4 days. The tweets are manually annotated so we can get the topic of sentiment beforehand. We have 12 topics; they are all around the impact of Jan 25th revolution in Egypt. The topics are the hottest events happened during that period of time. We took topics containing more than two tweets so they are relevant. The feature used in this experiment is TF-IDF (term frequency-inverse document frequency). During the preprocessing of data, each word is given an id and counted how many times it appears in each tweet. The tweet is presented as vector model with each word represented as two integers. One integer represents the word id, and the other represents the number of occurrence of this word in the object (tweet). Each word afterwards is multiplied by its IDF, which is obtained by dividing the total numbers of objects by the total number of objects containing the word. A word gets a high tf-idf when it occurs a lot in a certain object and less in the whole objects.

We used CLUTO tool for data preprocessing and clustering. The preprocessing consists of these tasks: tokenize words, stem words, remove stop words, calculate term weight, and represent the items to be clustered. In this experiment we did not apply stemming as we found difficulty to integrate an Arabic stemmer with the Perl script suggested by CLUTO tool to perform preprocessing. There was a problem in using Arabic letters, so we had to modify it so it can accept Arabic letters. We developed another small script to remove the stop words only; it was easier that way rather than merging both in one script. The stop words list was obtained by simply translating English list suggested by the tool as there was no Arabic stop word list available.

We performed 3 experiments that differ from each other in the numbers of clusters. We have 12 predetermined labels for our clusters, so we perform experiment on 6, 12 and 20 way clustering to compare results. The best result obtained was for the 20 clusters and the entropy for these 20 clusters was 0.385 and the purity was 0.573. The results for 12 clusters were 0.51 for entropy and 0.51 for purity. Increasing the number of clusters increased the average clustering quality. But for sure it could give more than one cluster containing tweets related to the same topic. However in real application we do not know the actual number of topics of the retrieved tweets. Therefore taking 20 as an empirical value of the number of clusters in all the following experiments.

B. Select Features

There are many features that can be used to represent the tweets such as TF-IDF (used in the above experiment), n-grams, part of speech tags n-gram, stylistic features and/or others. Determining the features that will produce the best possible purity using the common simple representations used in clustering namely TF-IDF and n-grams is required.

We used the same 110 tweets describe in the above experiments to compare the results obtained using TF-IDF as features representing the tweets with the results obtained using unigram, unigram and bigram, and unigram, bigram and trigram as features

The following results were obtained. The results obtained using unigram words that occurred more than 20 times in the whole corpus as a feature, resulted in purity of 0.704 and entropy of 0.275. When we combined unigrams and bigrams that occurred more than 20 times we got results of 0.694 for purity and 0.289 for entropy. When we combined unigrams, bigrams and trigrams that occurred more than 20 times we got results of 0.694 for purity and 0.289 for entropy. The results showed that using n-gram of words gave better results than the results obtained using TF-IDF for representing the tweets (the purity was 0.573 and the entropy was 0.385). The unigram of words gave the best results of all n-grams combination, but still using the combination of unigram, bigram and trigram didn't decrease the performance significantly. The improvements we got for purity and entropy of unigram of words over the TF-IDF representations are 22.86% and 40.00%. Therefore the unigram of words will be used for representing the features of tweets.

C. Determine the Clusters Labels

Key phrase extraction is the widely used technique for labeling the cluster, which is, actually can be considered the topic discussed in the cluster of tweets. One of the key phrase extraction techniques proved to be accurate is the KP-Miner [22]. Finding out whether KP-miner could be used for determining the clusters labels was investigated.

In effect, we used the 110 tweets clustered as describe here above. After clustering the tweets, we annotated each cluster and then extracted the first 3 key phrases using the KP-miner tool. The following procedure was applied to get a quantitative measure of how good the KP-miner tool did:

1. If the first key phrase generated matched the topic phrase we gave the tool score 1.0
2. If the first key phrase generated was part of the topic phrase we gave the tool score 0.75
3. If the second key phrase generated was part of the topic phrase we gave the tool score 0.5
4. If the key phrases generated was not part of the topic phrase we gave the tool score 0.0
5. If key phrases were generated while no topic was given by the human annotator we gave the tool score 0.0

Clustering algorithm could generate clusters with different topics of tweets inside this cluster. Comparing the key phrases generated with these topics is another measure for the KP-miner capability to label clusters generated. The evaluation procedure was done using the following procedure:

1. If the first key phrase generated matched the best topic phrase of the tweets we gave the tool score 1.0
2. If the first key phrase generated matched the second best topic phrase of the tweets we gave the tool score 0.75
3. If the first key phrase generated matched the third best topic phrase of the tweets we gave the tool score 0.5
4. If the first key phrase generated matched the fourth best topic phrase of the tweets we gave the tool score 0.25

The best topic phrase of the tweets in a cluster is the topic annotated by the human annotator to the majority of the tweets in a cluster. The second best topic phrase is the topic that was assigned to number of tweets less than the tweets assigned the best topic. The third best and fourth best can be defined in the same way.

The total score of the KP-miner tool using the first procedure was 14.5 out of 20 as we have 20 clusters, which represents 72.5%. The total score of the KP-miner tool using the second procedure was 15.75 out of 20 as we have 20 clusters, which represents 78.75%. Although the experiment was run on small number of tweets and the number of tweets per cluster was between 3 and 13 the KP-miner shows good performance as extracting the topic of a cluster is very challenging task. We will continue in experimenting this tool with larger number of clusters. Comparing these key phrases with the annotated topic of the clusters, we found that the topic could be one of the topmost 3 key phrases extracted. We also found that the purity of the cluster affects the quality of the results produced by the key phrase-extracting program.

4. A PROPOSED APPROACH FOR SENTIMENT CLASSIFICATION OF TWEETS

The proposed approach is a hybrid approach and is based on conducting a set of experiments to determine: sentiment words, and the thresholds of each n-gram to be included as a feature for the ML classification module, compare the ML and SO approaches and choose the ML classification algorithm.

A. Extract the sentiment words

Given the limited work done for Arabic text in the field of sentiment analysis, especially for the Egyptian dialect, we had first to start by manually building two lists: one for the most occurring positive sentiment words, and one for the most occurring negative sentiment words, using 600 sentiment annotated tweets (300 positive and 300 negative). Then for each word in these lists a weight is given to it based on its frequency in 300 positive tweets and its frequency in 300 negative tweets. A list of about 1200 positive and negative words was collected.

B. Determine the optimum threshold for each n-gram model

The feature vectors applied to the classifier consisted of the term frequency, as we are using statistical machine learning. Also, the different n-gram models were studied to analyze their influence on the classification problem. That is why we have chosen to work with unigrams, bigrams and trigrams as our work is on word/Phrase level sentiment analysis. Unigrams are considered the simplest features to extract and they provide good coverage for the data, while bigrams and trigrams provide the ability to capture any negation or sentiment expression patterns. Therefore, the process starts by extracting all the unigrams, bigrams, and trigrams in a 1000 annotated tweets corpus. Since the work done on the Egyptian dialect is very limited, we had to figure out the optimum threshold for each n-gram model. We have tried different frequency thresholds for each n-gram model until we reach

the thresholds, which are more suitable to our case. Figure 1 shows the performance of using each n-gram model separately as features, in which 0 indicates that all the n-grams were used, 1 indicates that n-grams greater than 1 were used, and so on.

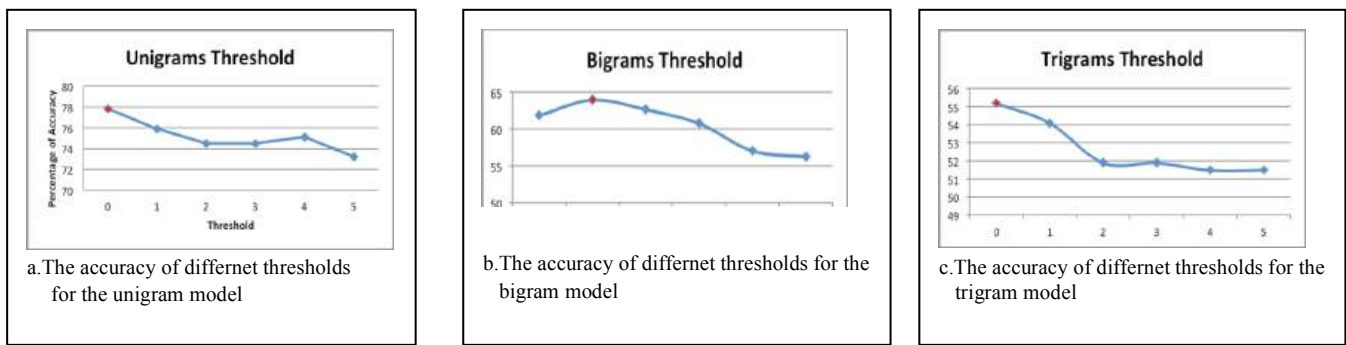


Figure 1: The accuracy of different thresholds for each n-gram model

It is clear from the graph that unigrams outperform bigrams and trigrams when determining the sentiment of the tweets. The results we have obtained using the NB and SVM classifiers are very similar to those obtained by Pang et al [35] as they have obtained 81.0% and 82.9% accuracy in case of unigrams for the same set of classifiers. This behavior is due to the fact that unigrams are able to provide good coverage for the data, whereas bigrams and trigrams tend to be very sparse. Therefore, it is better to combine unigrams, bigrams and trigrams as features to improve the performance of the sentiment classification problem.

Following the observations we got from the previous step, we then went into the process of trying various combinations of the different n-gram models to further improve the performance of the classification process. Since unigrams produced the best performance at threshold 0, we have fixed this threshold for the unigrams, and tried different thresholds for the bigrams and trigrams for the aim of reaching the highest possible performance. Figure 2 shows the performance of using a combined unigram and bigram model, and a combined unigram, bigram, and trigram model in which the threshold for the unigrams is fixed while the thresholds for the bigrams and trigrams are changing.

C. Compare the ML and SO approaches

To compare the proposed ML and the SO approaches, it was important to use the same inputs for training and testing in both approaches. Thus, the same 600 tweets used in SO approach to extract and give weights to the sentiment words will be used for

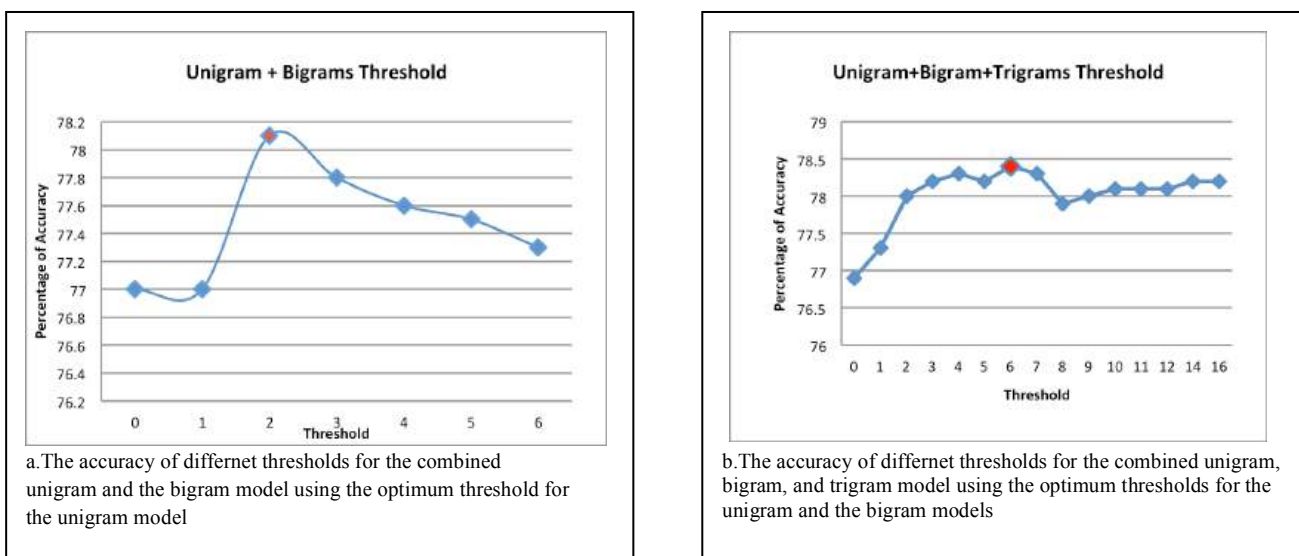


Figure 2: the accuracy of different combinations of the n-gram models

training both the SVM and the NB classifiers. Also, the same 400 tweets used to test the performance of the SO approach and construct confusion tables will be used to test the performance of both the SVM and NB classifiers.

To determine the class of each tweet In case of the SO approach, a cumulative score is calculated using the sentiment words in the tweet to determine its class. For each sentiment words present, its score is added to the total in the following way:

$$score = \sum_{i=1}^n (w_{pi} - w_{ni})$$

where w_{pi} is the positive weight of the word, w_{ni} is the negative weight of the word, and they are calculated based on the number of times this word appeared in the positive tweets, and the number of times this word appeared in the negative tweets. The weights assigned to the sentiment words are used to determine how close it is to positive “1” or to negative “-1”. The final value of the score (score > 0 or score < 0) determines polarity of the whole tweet. Since, in this stage we are only dealing with two classes building a binary classifier, positive and negative, the neutral class, where either no sentiment words were found or both numbers of positive and negative sentiment words are equal, is not acceptable. Thus for each class a classifier is built determining whether the tweet belongs to its corresponding class, or it belongs to the class named “other”. Then, the accuracy, the precision, the recall, and the F-measure of each classifier will be calculated, which will be averaged at the end to reach a final unified classifier.

Comparing the results obtained in the SO experiment with the ML experiment, related to the accuracy measure, it was clear that the best result (73.25%) obtained using the SVM learning algorithm in the ML approach outperforms the average result (69.5%) obtained using the SO approach. This improvement is almost 3.75% given that only one feature was considered in the ML experiment, which is the frequency of n-grams in the corpus. Thus for ML, adding more features is expected to further improve the performance, however for the SO it depends only on the reliability of the sentiment words list to improve its performance. It is important to note that the performance of the positive classifier is better than the performance of the negative classifier in the SO approach, which means that if we only used the positive classifier we could reach a performance closer or even better than the ML approach. However, for both experiments, the presence of different forms for the Arabic words plays a major role in decreasing the performance of both experiments like “المؤيد والمؤيد”. This creates the need for extensive preprocessing to be performed before the classification is performed like stemming and normalization, yet great care should be taken so as not to change the sentiment of the tweet. Like for example, the word “وفي - loyal” which implies a positive sentiment, however if this word is stemmed then it would become the particle word “في - in” which is originally a stop word, so it will be removed from the tweet, thus the sentiment of the tweet will then become unknown.

D. Choosing the Machine Learning classification algorithm

To compare the performance of the NB and SVM classification methods using unigram, bigram, and trigram as features, 500 positive tweets and 500 negative tweets were tested using 10-fold cross validation method.

Comparing the results of SVM and NB in both experiments, it is clear that SVM has better results than NB in almost all the cases. The improvement between the best accuracy results of both models is almost 7.8% for SVM. This behavior was observed in more than one study as usually SVM produces more accurate results than the NB. Moreover, the results obtained by the SVM have been shown to be highly effective in sentiment analysis outperforming the results obtained by the NB. By comparing the results obtained by SVM in sentiment analysis in general, it is noticeable that SVM overcomes other machine learning techniques

5. CONCLUSION AND FUTURE WORK

The following paragraphs summarize the results that helped us in determining: our approaches for topic extraction and sentiment classification, and the research that is still needed to enhance the results.

The results reported in this paper regarding clustering, used the translated set of English stop words which needs a lot of addition. We have a new stop word list developed but did not try it yet on the clustering algorithm. Using unigram words that occurred more than 20 times in the whole corpus as features for using bisecting k-mean clustering algorithm resulted in purity of 0.704 and entropy of 0.275. These results were better than using tf-idf of words as features where the results were 0.6 for purity and 0.36 for entropy. Therefore unigrams were decided to be the features we used for clustering. However more research is needed to investigate other clustering algorithms, study the impact of removing the whole set of stop words and perform stemming.

A methodology for evaluating Key-phrase extraction algorithm to recognize the sentiment topic contained in a cluster was developed and applied on a small set of 110 annotated tweets. The score generated for the quality of the generated topic was 72.5%. There is still work to be done here, as other methods using different approaches still need to be investigated such as considering: name entity recognition as most of the topics are actually taking about name entities, hash tags, supervised learning approach, and hybrid approaches.

A lexicon of approximately 1600 sentiment words for Egyptian dialect and their weights for their polarities as positive or negative was built to develop an unsupervised classifier using Semantic Oriented approach. Comparing the results obtained in the SO experiment with the ML experiment, related to the accuracy measure, has revealed that the accuracy of the SVM learning algorithm was 73.25% while the accuracy of the SO approach was 69.5%. However, we used the SO approach in the prototype implementation as we faced difficulty in exporting the classifier built using WEKA tool to the application we built on the Web. Early work conducted to investigate the valence shifter on sentiment classification in English using benchmark data set appeared in [31]. Preliminary results of sentiment classification of Arabic Tweets, has been published in [44] and the impact of pre-processing on the accuracy of the sentiment classification was published in [45].

ACKNOWLEDGEMENT

This work was done in a project entitled Sentiment and Opinion Mining for Arabic Web funded by ITIDA, Ministry of Communication and Information Technology (MCIT). I would like to acknowledge: Sara Morsy and Amira Shoukry who worked in this project and some parts of the material used in this paper were done with their collaborations and published in the proceedings of International conferences as indicated in the paper.

6. BIBLIOGRAPHY

1. Abbasi, Ahmed, & Chen, Hsinchun. 2005. Applying Authorship Analysis to Extremist-Group Web Forum Messages. *IEEE Intelligent Systems*, 20(September), 67–75.
2. Abbasi, Ahmed, Chen, Hsinchun, & Salem, Arab. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Trans. Inf. Syst.*, 26(June), 12:1–12:34.
3. Abbasi, Ahmed, France, Stephen, Zhang, Zhu, & Chen, Hsinchun. 2011. Selecting Attributes for Sentiment Classification Using Feature Relation Networks. *IEEE Trans. On Knowl. and Data Eng.*, 23(March), 447–462.
4. Allan, James; Carbonell, Jaime; Doddington, George ; Yamron, Jonathan ; Yang, Yiming ,1998. "Topic Detection and Tracking Pilot Study Final Report" In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop* pp. 194-218
5. Andreevskaia, A., & Bergler, S. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. Pages 290–298 of: *Proceedings of ACL-08 HLT*. Columbus, Ohio, USA: Association for Computational Linguistics.
6. Baccianella, S., Esuli, A., & Sebastiani, F. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. Page 2010 of: *Proceedings of the 7th conference on International Language Resources and Evaluation*, vol. 25.
7. Blitzer, J., Dredze, M., & Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. Pages 440–447 of: *Annual Meeting-Association For Computational Linguistics*, vol. 45.
8. Brooke, J. 2009. A Semantic Approach to Automated Text Sentiment Analysis. Masters Thesis, Simon Fraser University.
9. Carrillo de Albornoz, Jorge, Plaza, Laura, & Gervás, Pablo. 2010. A hybrid approach to emotional sentence polarity and intensity classification. Pages 153–161 of: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. CoNLL '10. Stroudsburg, PA, USA: Association for Computational Linguistics.
10. Cataldi, Mario; Di Caro, Luigi; Schifanella, Claudio. 2010." Emerging topic detection on Twitter based on temporal and social terms evaluation." In *Proceedings of the Tenth International Workshop on Multimedia Data Mining (MDMKDD '10)*. ACM, New York, NY, USA, , Article 4 , 10 pages.
11. Choi, Yoonjung; Jung, Yuchul; Myaeng, Sung-Hyon" Identifying Controversial Issues and Their Sub-topics in News Articles" Book Title: "Intelligence and Security Informatics" Book Series Title:" Lecture Notes in Computer Science" , 2010, Chen,H. et al. (Eds.) Springer Berlin / Heidelberg pp. 140-153
12. Dai, Xiang-Ying; Chen, Qing-Cai; Wang, Xiao-Long; Xu Jun; , "Online topic detection and tracking of financial news based on hierarchical clustering," *Machine Learning and Cybernetics (ICMLC)*, 2010 International Conference on , vol.6, no., pp.3341-3346, 11-14 July 2010

13. Dai, Xiangying; Sun, Yunlian; 2010. "Event identification within news topics," *Intelligent Computing and Integrated Systems (ICISS)*, 2010 International Conference on , vol., no., pp.498-502, 22-24
14. Dang, Yan, Zhang, Yulei, & Chen, HsinChun. 2010. A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews. *IEEE Intelligent Systems*, 25(July), 46–53.
15. Dash, M., & Liu, H. 1997. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4), 131–156.
16. El-Beltagy, S.R.; Rafea, A.,” KP-Miner: A key phrase extraction system for English and Arabic Documents”, *Information Systems* (2008)
17. Esuli, A., & Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. Pages 417–422 of: *Proceedings of the 5th Conference on International Language Resources and Evaluation*, vol. 6. Citeseer.
18. Fei, Zhongchao, Liu, Jian, & Wu, Gengfeng. 2004. Sentiment Classification Using Phrase Patterns. Pages 1147–1152 of: *Proceedings of the The Fourth International Conference on Computer and Information Technology. CIT '04*. Washington, DC, USA: IEEE Computer Society.
19. Gamon, Michael. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: *Proceedings of the 20th international conference on Computational Linguistics. COLING '04*. Stroudsburg, PA, USA: Association for Computational Linguistics.
20. He, Yulan. 2010. Learning sentiment classification model from labeled features. Pages 1685–1688 of: *Proceedings of the 19th ACM international conference on Information and knowledge management. CIKM '10*. New York, NY, USA: ACM.
21. Hoogma, Niek “The Modules and Methods of Topic Detection and Tracking” 2nd Twente Student Conference on IT, 2005.
22. Huafeng, Xie; Fang, Wu; Xuying, Lu “Study on the Optimization Design of the Subject Indexing Based on the Word-frequency Statistics” *computer and information science*, vol. 4 no.2, March 2011
23. Huang, Zhen; Cardenas, Alfonso F., “Extracting Hot Events from News Feeds, Visualization, and Insights”, 2009
24. Jain, S.; Pareek, J. 2009. "KeyPhrase Extraction Tool (KET) for Semantic Metadata Annotation of Learning Materials," 2009 International Conference on Signal Processing Systems , vol., no., pp.625-628, 15-17.
25. Jia, Lifeng, Yu, Clement, & Meng, Weiyi. 2009. The effect of negation on sentiment analysis and retrieval effectiveness. Pages 1827–1830 of: *Proceeding of the 18th ACM conference on Information and knowledge management. CIKM '09*. New York, NY, USA: ACM.
26. Khoo, Khyou Bun; Ishizuka, M. 2002. "Topic extraction from news archive using TF*PDF algorithm," *Web Information Systems Engineering, 2002. WISE 2002. Proceedings of the Third International Conference on , vol., no., pp. 73- 82, 12-14*
27. Liu, B. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 978–1420085921.
28. Lopez, C.; Prince, V.; Roche, M.;"Automatic titling of electronic documents with noun phrase extraction," *Soft Computing and Pattern Recognition (SoCPar)*, 2010 International Conference of , vol., no., pp.168-171, 7-10 Dec. 2010
29. Makkonen, Juha “Semantic Classes in Topic Detection and Tracking” University of Helsinki Finland, Department of Computer Science, PhD Thesis, Series of Publications A, Report A-2009-8 Helsinki, November 2009, 165 pages
30. Miller, George A. 1995. WordNet: a lexical database for English. *Commun. ACM*, 38(November), 39–41.
31. Morsy, Sara and Rafea, Ahmed. 2012. Improving Document-Level Sentiment Classification Using Contextual Valence Shifters *Natural Language Processing and Information Systems Lecture Notes in Computer Science Volume 7337*, 2012, pp 253-258
32. Ng, V., Dasgupta, S., & Arifin, SM. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. Pages 611–618 of: *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics.
33. Nicholls, C., & Song, F. 2010. Comparison of Feature Selection Methods for Sentiment Analysis. *Advances in Artificial Intelligence*, 286–289.
34. Okamoto, Masayuki; Kikuchi, Masaaki. 2009. “Discovering Volatile Events in Your Neighborhood: Local-Area Topic Extraction from Blog Entries.” In *Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology (AIRS '09)*, Gary Geunbae Lee, Dawei Song, Chin-Yew Lin, Akiko Aizawa, Kazuko Kuriyama, Masaharu Yoshioka, and Tetsuya Sakai (Eds.). Springer-Verlag, Berlin, Heidelberg, 181-192

35. Pang, B., Lee, L., & Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. Pages 79–86 of: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics.
36. Pang, Bo, & Lee, Lillian. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics.
37. Pang, Bo, & Lee, Lillian. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retrieval.*, 2(January), 1–135.
38. Polanyi, L., & Zaenen, A. 2006. Contextual valence shifters. *Computing attitude and affect in text: Theory and applications*, 1–10.
39. Qiu, Likun, Zhang, Weishi, Hu, Changjian, & Zhao, Kai. 2009. SELC: a self-supervised model for sentiment classification. Pages 929–936 of: Proceeding of the 18th ACM conference on Information and knowledge management. CIKM '09. New York, NY, USA: ACM.
40. Read, Jonathon, & Carroll, John. 2009. Weakly supervised techniques for domain-independent sentiment classification. Pages 45–52 of: Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion. TSA '09. New York, NY, USA: ACM.
41. Sarmiento, Luis; Nunes, Sergio “Automatic Extraction of Quotes and Topics from News Feeds”
<http://hdl.handle.net/10216/7080>
42. Sebastiani, Fabrizio. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(March), 1–47.
43. Seo, Young-Woo; Sycara, Katia. “Text Clustering for Topic Detection”, Carnegie Mellon University, 2004
44. Shoukry, Amira, and Rfaea, Ahmed. 2012. Sentence-level Arabic sentiment analysis, *Collaboration Technologies and Systems (CTS)*, 2012 International Conference on , vol., no., pp.546-550, 21-25 May 2012 doi: 10.1109/CTS.2012.6261103
45. Shoukry, Amira, and Rafea, Ahmed. 2012. Preprocessing Egyptian Dialect Tweets for Sentiment Mining, *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages. AMTA 2012 - San Diego, CA USA*, November, 2012
46. Sokolova, M., & Lapalme, G. 2008. Verbs speak loud: Verb categories in learning polarity and strength of opinions. *Advances in Artificial Intelligence*, 320–331.
47. Sokolova, M., & Lapalme, G. 2009. Classification of opinions with non-affective adverbs and adjectives. Pages 416–420 of: Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing.
48. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. 2010. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 1–41.
49. Tomokiyo, Takashi; Hurst, Matthew. 2003. “A language model approach to keyphrase extraction”. In Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18 (MWE '03), Vol. 18. Association for Computational Linguistics, Stroudsburg, PA, USA, 33-40.
50. Turney, Peter D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. Pages 417–424 of: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics.
51. Tweedie, F.J., & Baayen, R.H. 1998. How variable may a constant be? Measures of lexical richness in perspective. *Computers and the Humanities*, 32(5), 323–352.
52. Wang, Canhui; Zhang, Min; Ru, Liyun; Ma, Shaoping; 2008. "An Automatic Online News Topic Keyphrase Extraction System," *Web Intelligence and Intelligent Agent Technology*, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on , vol.1, no., pp.214-219, 9-12.
53. Wang, S., Li, D., Wei, Y., & Li, H. 2009. A Feature Selection Method Based on Fishers Discriminant Ratio for Text Sentiment Classification. *Web Information Systems and Mining*, 88–97.
54. Wang, XiaoLing; Mu, DeJun; Fang, Jun; 2008, "Improved Automatic Keyphrase Extraction by Using Semantic Information," *Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2008 International, vol.1, no., pp.1061-1065, 20-22.

55. Wartena, C.; Brussee, R.; "Topic Detection by Clustering Keywords," Database and Expert Systems Application, 2008. DEXA '08. 19th International Workshop on, vol., no., pp.54-58, 1-5 Sept. 2008
56. Whitelaw, C., Garg, N., & Argamon, S. 2005. Using appraisal groups for sentiment analysis. Pages 625–631 of: Proceedings of the 14th ACM international conference on Information and knowledge management. ACM.
57. Witten, I.H., & Frank, E. 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann Pub.
58. Witten, Ian H.; Paynter, Gordon W.; Frank, Eibe; Gutwin, Carl; Nevill-Manning, Craig G. 1999. "KEA: Practical automatic keyphrase extraction". In *Proceedings of the fourth ACM conference on Digital libraries (DL '99)*. ACM, New York, NY, USA, 254-255.
59. Wilson, Theresa, Wiebe, Janyce, & Hoffmann, Paul. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. Pages 347–354 of: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics.
60. Xu ,Rui; Wunsch, Donald C. " Clustering; 2009. " IEEE Press Series on Computational Intelligence A JOHN WILEY & SONS, INC., PUBLICATION.
61. Xu, Yong-Dong; Quan, Guang-Ri; Xu, Zhi-Ming; Wang Ya-Dong; 2009. "Research on Text Hierarchical Topic Identification Algorithm Based on the Dynamic Diverse Thresholds Clustering," Conference on Asian Language Processing, 2009. IALP '09. International, vol., no., pp.206-210.
62. Zhang, Chengzhi; Zhang, Qingguo; 2008. "Topic Navigation Generation Using Topic Extraction and Clustering," Knowledge Acquisition and Modeling, 2008. KAM '08. International Symposium on , vol., no., pp.333-339.
63. Zhang, Xiaoming ; Li, Zhoujun , 2010. "Automatic Topic Detection with an Incremental Clustering Algorithm" Wang, F.L. et al. (Eds.): WISM 2010, LNCS 6318, pp.344-351, Springer-Verlag
64. Zhang, Yan; Shi, Lei; Sun, Bai; Kong, Liang, 2009. "Web Forum Sentiment Analysis Based on Topics," Computer and Information Technology, 2009. CIT '09. Ninth IEEE International Conference on , vol.2, no., pp.148-153, 11-14.

Smoothing Techniques for Arabic Diacritics Restoration

Yasser Hifny

Univeresity of Helwan, Egypt

yhifny@fci.helwan.edu.eg

Abstract

An algorithm to restore Arabic diacritics using dynamic programming approach was presented in [1]. The possible word sequences with diacritics are assigned scores using statistical n -gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm. The maximum likelihood (ML) estimation of the stochastic language model parameters leads to poor diacritization accuracy due to data sparse problem. Smoothing aims to handle this problem by taking some probability mass from the observed n -gram and distribute it to the unseen n -grams. In this paper, we show that applying smoothing techniques dramatically improve the diacritization accuracy. The interpolated version of the absolute discounting method leads to the best results for different model orders.

Index Terms: Arabic diacritics restoration, dynamic programming, statistical language modeling, smoothing

1 INTRODUCTION

Restoration of Arabic diacritics is an active area of research [2],[3],[4],[5],[6],[7],[8],[1]. When the diacritics are present, the Arabic script provides enough information about the correct pronunciation and the meaning of the words. In some applications like Arabic text to speech, diacritics are necessary to get the correct pronunciation [9]. Moreover, diacritics help to get the reference transcription for speech recognition systems [10].

In [1], an algorithm to restore Arabic diacritics using a dynamic programming approach was presented. The possible word sequences with diacritics are assigned scores using statistical n -gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm. The presented technique is purely statistical approach and depends only on an Arabic corpus annotated with diacritics. The described algorithm is closely related to the Viterbi algorithm which is used to find the most likely sequence in the hidden Markov models. However, in our formulation we do not have hidden states as in Hidden Markov Models (HMMs).

The possible word sequences with diacritics are assigned scores using statistical n -gram language models. The maximum likelihood estimate for these models leads to poor results due to the data sparse problem. Smoothing techniques aims to provide reliable probability estimate and may lead to better results. In this paper, we provide empirical study using different smoothing techniques commonly used in speech recognition field. The empirical results show that applying smoothing techniques dramatically improve the diacritization accuracy. Absolute discounting and the interpolated version of the Modified Kneser-Ney methods lead to the best results.

In Section 2, a mathematical formulation of the restoration of Arabic diacritics is described. Some smoothing techniques are reviewed in Section 3. Sections 4 and 5 give experimental results on a public domain task and conclusions.

2 PROBLEM FORMULATION

A fundamental problem in natural language processing is to give a score to a sentence hypothesis or sequence of words. The assigned score for a sentence can be used to disambiguate between different solutions. Large vocabulary speech recognition systems (LVCSR) use a n -gram language model, which gives an approximate probability score of an allowable word sequence $\mathbf{W} = w_0 w_1 \dots w_m$ in the recognition task. This probability score is calculated by accumulating local scores using $n - 1$ Markov chains over the word sequence and is given by

$$P(\mathbf{W}) = \prod_{i=1}^m P(w_i | w_{i-n+1}^{i-1}) \quad (1)$$

Table 1: Some possible word diacritics for an Arabic sentence.

يعلم	لم	ما	الانسان	علم
يَعْلَمُ	لَمْ	مَا	الْأَنْسَانُ	عَلِمَ
يُعَلِّمُ			الْأَنْسَانَ	عَلِمَ
يَعْلَمُ			الْأَنْسَانِ	عَلِمَ
يَعْلَمُ				عُلِّمَ

For bigram language model, This probability score is given by

$$P(\mathbf{W}) = \prod_{i=1}^m P(w_i|w_{i-1}) \quad (2)$$

and the $P(w_i|w_{i-1})$ is estimated by

$$P(w_i|w_{i-1}) = \frac{c(w_i, w_{i-1})}{c(w_{i-1})} \quad (3)$$

where $c(w_i, w_{i-1}), c(w_{i-1})$ can be computed from a large training corpus.

A. Diacritics restoration problem

In the context of Arabic restoration problem, the diacritized Arabic word sequence can be assigned a probability score based on a language model. This language model must trained on a large corpus of diacritized Arabic text. For example, Table 1 shows some possible diacritics alternative for the undiacritized sentence:

علم الانسان ما لم يعلم

Some of the possible hypotheses for this undiacritized sentence are:

عَلِمَ-الْأَنْسَانَ مَا لَمْ يَعْلَمُ
 عَلِمَ-الْأَنْسَانَ مَا لَمْ يُعَلِّمُ
 عَلِمَ-الْأَنْسَانَ مَا لَمْ يَعْلَمُ
 عَلِمَ-الْأَنْسَانَ مَا لَمْ يَعْلَمُ
 عِلِمَ-الْأَنْسَانَ مَا لَمْ يَعْلَمُ
 عِلِمَ-الْأَنْسَانَ مَا لَمْ يُعَلِّمُ
 عِلِمَ-الْأَنْسَانَ مَا لَمْ يَعْلَمُ
 عِلِمَ-الْأَنْسَانَ مَا لَمْ يَعْلَمُ

Based on a language model, a score is assigned to each hypothesis. For example, given a bigram language model, the hypothesis $لَمْ يَعْلَمُ مَا لَمْ يَعْلَمُ$ is assigned a score as follow:

$$P(\mathbf{H}) = P(لَمْ يَعْلَمُ)P(مَا لَمْ يَعْلَمُ)P(الْأَنْسَانَ)P(عَلِمَ-الْأَنْسَانَ)P(مَا لَمْ يَعْلَمُ)P(عَلِمَ-عَلِمَ) \quad (4)$$

Similarly, all possible hypotheses are scored in a similar way. The best path or the most likely diacritized word sequence is assigned to the hypothesis that has the highest score.

Using this statistical scoring method, Arabic diacritics restoration problem can be defined as a problem of choosing a word sequence $\hat{\mathbf{H}}$ with the *maximum language model score* given undiacritized word sequence \mathbf{W} :

$$\hat{\mathbf{H}} = \arg \max_{\mathbf{H}} P(\mathbf{H}|\mathbf{W}) \quad (5)$$

A brute force approach to score all hypotheses may be computationally expensive. An efficient algorithm to find the most likely hypothesis is discussed in [1].

3 SMOOTHING

The maximum likelihood (ML) estimation of the stochastic language model parameters is based on Equation (3). This estimation is problematic since it assigns zero probabilities to unseen n -grams in the training data. Smoothing aims to handle this problem by taking some probability mass from the observed n -gram and distribute it to the unseen n -grams [11], [12].

Several smoothing techniques can be expressed in one of the following two forms:

$$P_{\text{Interp}}(w_i|w_{i-n+1}^{i-1}) = \alpha(w_i|w_{i-n+1}^{i-1}) + \gamma(w_{i-n+1}^{i-1})P_{\text{Interp}}(w_i|w_{i-n+2}^{i-1}) \quad (6)$$

$$P_{\text{bo}}(w_i|w_{i-n+1}^{i-1}) = \begin{cases} \alpha(w_i|w_{i-n+1}^{i-1}) & \text{if } c(w_{i-n+1}^i) > 0, \\ \gamma(w_{i-n+1}^{i-1})P_{\text{bo}}(w_i|w_{i-n+2}^{i-1}) & \text{otherwise} \end{cases} \quad (7)$$

where Equation (6) describes the so-called interpolated models and Equation (7) describes the backoff models. $\alpha(w_i|w_{i-n+1}^{i-1})$ is the discounted probability distribution and $\gamma(w_{i-n+1}^{i-1})$ is a backoff weight depends on the n -gram history w_{i-n+1}^{i-1} and ensures the distribution is properly normalized. The interpolated models always utilize the lower order distribution whereas the backoff model utilizes it only when the $c(w_{i-n+1}^i) = 0$

A. Katz smoothing

Katz smoothing [13], is a method that combines the Good-Turing discounting formula [14] along with a backoff model Equation(7) to obtain a reliable estimate for the unseen words.

Let n_r represent the number of n -grams that occur r times, i.e.

$$n_r = |w_{i-n+1} \dots w_i | c(w_{i-n+1} \dots w_i) = r| \quad (8)$$

According to Good, for any n -gram that occurs r times, we should discount it, pretending that it occurs $r^*(r)$ times where

$$r^* = (r + 1) \frac{n_{r+1}}{n_r} \quad (9)$$

The discounted probability $\alpha(w_i|w_{i-n+1}^{i-1})$ is given by

$$\alpha(w_i|w_{i-n+1}^{i-1}) = d(r) \frac{c(w_{i-n+1}^i)}{c(w_{i-n+2}^i)} \quad (10)$$

where

$$d(r) = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \quad (11)$$

and the backoff weight $\gamma(w_{i-n+1}^{i-1})$ for the Katz smoothing is given by:

$$\gamma(w_{i-n+1}^{i-1}) = \frac{1 - \sum_{w_i:r>0} P_{\text{bo}}(w_i|w_{i-n+1}^{i-1})}{1 - \sum_{w_i:r>0} P_{\text{bo}}(w_i|w_{i-n+2}^{i-1})} \quad (12)$$

B. Absolute Discounting

Absolute discounting [15] subtracts a fixed constant D from the ML estimate to assign probability for the unseen words. A possible D can be selected for each n -gram order. $\alpha(w_i|w_{i-n+1}^{i-1})$ and $\gamma(w_{i-n+1}^{i-1})$ for the interpolated form are given by:

$$\alpha(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{0, c(w_{i-n+1}^i) - D\}}{\sum_{w_i} c(w_{i-n+1}^i)} \quad (13)$$

$$\gamma(w_{i-n+1}^{i-1}) = \frac{D}{\sum_{w_i} c(w_{i-n+1}^i)} N_{1+}(w_{i-n+1}^{i-1}*) \quad (14)$$

where $N_{1+}(w_{i-n+1}^{i-1}*)$ can be computed as follow:

$$N_{1+}(w_{i-n+1}^{i-1}*) = |\{w_i : c(w_{i-n+1}^i w_i) > 0\}| \quad (15)$$

The notation N_{1+} means the number of unique words following history w_{i-n+1}^{i-1} amongst n -grams that are found to occur 1 times in the training data.

C. Kneser-Ney smoothing

Kneser-Ney smoothing is the most commonly used technique in speech recognition and machine translation fields. It is originally a backoff method and computes the backoff distribution in a novel way. It was shown that a modified version of the Kneser-Ney smoothing method can lead to better results in [11]. $\alpha(w_i|w_{i-n+1}^{i-1})$ for the interpolated form of the modified Kneser-Ney is given by:

$$\alpha(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) - D(c(w_{i-n+1}^i))}{c(w_{i-n+1}^{i-1})} \quad (16)$$

The count-specific paramters are defined as

$$D(c) = \begin{cases} 0 & \text{if } c = 0, \\ D_1 & \text{if } c = 1, \\ D_2 & \text{if } c = 2, \\ D_{3+} & \text{if } c \geq 3 \end{cases} \quad (17)$$

The distribution must sum to 1, so the $\gamma(w_{i-n+1}^{i-1})$ is computed as:

$$\gamma(w_{i-n+1}^{i-1}) = \frac{D_1 N_1(w_{i-n+1}^{i-1}*) + D_2 N_2(w_{i-n+1}^{i-1}*) + D_{3+} N_{3+}(w_{i-n+1}^{i-1}*)}{c(w_{i-n+1}^{i-2})} \quad (18)$$

where $N_2(w_{i-n+1}^{i-1}*)$ and $N_{3+}(w_{i-n+1}^{i-1}*)$ are defined similar to $N_1(w_{i-n+1}^{i-1}*)$.

The discount parameters are computed from the count-of-count n_r from the training data as follows:

$$Y = \frac{n_1}{n_1 + n_2} \quad (19)$$

$$D_1 = 1 - 2Y\left(\frac{n_2}{n_1}\right) \quad (20)$$

$$D_2 = 1 - 2Y\left(\frac{n_3}{n_2}\right) \quad (21)$$

$$D_{3+} = 1 - 2Y\left(\frac{n_4}{n_3}\right) \quad (22)$$

Table 2: The properties of the training and test data of the Tashkeela corpus.

Property	Training data	Test data
Number of words	52500084	1902145
Out of Vocabulary (OOV)	-	6459

Table 3: Tashkeela dictionary properties.

Property	Word count
unique diacritized words	794456
unique undiacritized words (keys)	419857

4 EXPERIMENTS

Automatic diacritics restoration experiments were carried out on the Arabic vocalized text corpus: Tashkeela [16]. The corpus is collected using automatic web crawling methods and it is free. It is collected from Islamic religious heritage books and it contains 6,149,726 words.

In this work, the undiacritized Arabic words and sentences were removed from the corpus. The corpus needs further text normalization steps which are beyond the scope of this work. For example, the word `الأنسان` appears in corpus in two forms due to manual data entry errors. These two words should be normalized to one word. Correcting these mistakes may improve the statistics gathered from the data. The corpus was divided into training and test sets. The properties of the training and test data are summarized in Table 2. The system dictionary was built from the training data and its properties are detailed in Table 3.

The language models were built using the SRILM toolkit [17]. Several model orders were investigated. The model with order 2 contains 794,479 monograms and 10,847,630 bigrams. The correct word count and the diacritization Word Error Rate (WER) are shown in Table 4. The measure WER2 is computed by removing the last diacritic of the words (ignoring the case ending diacritic). It can be seen that most diacritization errors are related to the syntax of Arabic language.

Applying smoothing techniques improve dramatically the results. For models with order 2, the interpolated versions of the modified Kneser-Ney and absolute discounting have similar performance. This performance is slightly better than the backoff Katz method. The parameters `-gt1min`, `-gt1max`, `-gt2min`, `-gt2max` were investigated for the Katz’s method and are detailed in Table 4. The SRILM default tuning parameters were applied for the modified Kneser-Ney method. In addition, D parameter for absolute discounting was investigated. Table 5 shows the results for models with order 3 and 4. The interpolated version of the absolute discounting method leads to the best results.

5 CONCLUSIONS

In dynamic programming based Arabic diacritics restoration, the possible word sequences with diacritics are assigned scores using statistical n -gram language models. The maximum likelihood estimate for these models leads to poor

Table 4: Diacritization performance on the Tashkeela database (model order = 2).

Smoothing method	Correct	WER	WER2
Smoothing is disabled [1]	744589	61.0%	35.6%
Katz (3,7,3,7)	1662367	12.6%	4.6%
Katz (1,1,1,1)	1692458	11.0%	4.6%
Katz (1,1,1,7)	1723104	9.4%	3.6%
Katz (1,1,1,3)	1722909	9.4%	3.6%
I-Modified Kneser-Ney	1725637	9.2%	3.6%
Absolute Discounting ($D = 0.1$)	1722708	9.4%	3.7%
Absolute Discounting ($D = 0.9$)	1716873	9.3%	3.5%
Absolute Discounting ($D = 0.5$)	1725427	9.2%	3.6%

Table 5: Diacritization performance on the Tashkeela database.

Smoothing method	Order	Correct	WER	WER2
Katz (default parameters)	3	1729605	9.0%	3.4%
I-Modified Kneser-Ney	3	1721121	9.5%	3.6%
Absolute Discounting ($D = 0.5$)	3	1731803	8.9%	3.4%
Katz (default parameters)	4	1729438	9.0%	3.5%
I-Modified Kneser-Ney	4	1720832	9.5%	3.7%
Absolute Discounting ($D = 0.5$)	4	1732175	8.9%	3.4%

results due to the data sparse problem. Smoothing techniques aim to provide reliable probability estimate and may lead to better results.

In this paper, we provide empirical study using different smoothing techniques commonly used in speech recognition and machine translation fields. The empirical results show that applying smoothing techniques dramatically improve the diacritization accuracy. The interpolated versions of the absolute discounting and the Modified Kneser-Ney method show similar performance for models with order 2. For higher order models (i.e. order 3 and 4), the interpolated version of the absolute discounting method leads to the best results.

References

- [1] Y. Hifny, “Restoration of Arabic diacritics using dynamic programming,” *submitted to COLING*, 2012.
- [2] Y. Gal, “An HMM approach to vowel restoration in Arabic and Hebrew,” in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2002, pp. 66–73.
- [3] R. Nelken and S. M. Shieber, “Arabic diacritization using weighted finite-state transducers,” in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2005.
- [4] I. Zitouni, J. S. Sorensen, and R. Sarikaya, “Maximum entropy based restoration of Arabic diacritics,” in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 2006, pp. 577–584.
- [5] N. Habash and O. Rambow, “Arabic diacritization through full morphological tagging,” in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, ser. NAACL-Short ’07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 53–56. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1614108.1614122>
- [6] M. Alghamdi and Z. Muzaffar, “Kacst Arabic diacritizer,” in *The First International Symposium on Computers and Arabic Language*, 2007.
- [7] K. Shaalan, H. M. Abo Bakr, and I. Ziedan, “A hybrid approach for building Arabic diacritizer,” in *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, 2009.
- [8] M. Rashwan, M. Al-Badrashiny, M. Attia, S. Abdou, and A. Rafea, “A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 166 – 175, 2011.
- [9] Y. Hifny and et al., “ARABTALK: An implementation for arabic text to speech system,” in *Proceedings of the 4th conference of the Egyptian Society of Language Engineering (ESLE)*, October 2003.
- [10] D. Vergyri and K. Kirchhoff, “Automatic diacritization of Arabic for acoustic modeling in speech recognition,” in *COLING 2004 Computational Approaches to Arabic Script-based Languages*, 2004.
- [11] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Harvard Univ., Computer Science Group, Cambridge, MA, Tech. Rep. TR-10-98, August 1998.
- [12] J. T. Goodman, “A bit of progress in language modeling,” Microsoft Research, Tech. Rep. MSR-TR-2001-72, 2001.

- [13] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400–401, March 1987.
- [14] I. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3 and 4, pp. 237–264, 1953.
- [15] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependences in stochastic language modeling," *Computer, Speech, and Language*, vol. 8, pp. 1–38, 1994.
- [16] T. Zerrouki, "Tashkeela: Arabic vocalized text corpus," 2011. [Online]. Available: <http://aracorpus.e3rab.com>
- [17] A. Stockle, "SRILM: the SRI language modeling toolkit." in *Proc. ICSLP*, 2002.

Higer Order n -gram Language Models for Arabic Diacritics Restoration

Yasser Hifny

Univeresity of Helwan, Egypt

yhifny@fci.helwan.edu.eg

Abstract

Dynamic programming based Arabic diacritics restoration aims to assign diacritics to Arabic words. The technique is purely statistical approach and depends only on an Arabic corpus annotated with diacritics. The possible word sequences with diacritics are assigned scores using statistical n -gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm. In previous work [1], the assigned scores are based on a bigram stochastic language model and the decoder was restricted to this model. Using higher order n -gram language may lead to better diacritization accuracy. In this work, we extend the dynamic programming decoding algorithm to support higher order language models. Preliminary results on a public domain corpus show that dynamic programming decoding based on higher order n -gram models can lead to better results than bigram models.

Index Terms: Arabic diacritics restoration, dynamic programming, statistical language modeling, smoothing

1 INTRODUCTION

Arabic diacritics restoration aims to assign diacritics to Arabic words. Over the last few yeas, many systems to restore Arabic diacritics were developed [2],[3],[4],[5],[6],[7],[8]. When the diacritics are present, the Arabic script provides enough information about the correct pronunciation and the meaning of the words. In some applications like Arabic text to speech, diacritics are necessary to get the correct pronunciation [9]. Moreover, diacritics help to get the reference transcription for speech recognition systems [10].

In [1], an algorithm to restore Arabic diacritics using a dynamic programming approach was presented. The possible word sequences with diacritics are assigned scores using statistical n -gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm. The described algorithm is closely related to the Viterbi algorithm which is used to find the most likely sequence in the hidden Markov models. However, in our formulation we do not have hidden states as in Hidden Markov Models (HMMs). The presented technique is purely statistical approach and depends only on an Arabic corpus annotated with diacritics.

In previous work, the assigned scores are based on a bigram stochastic language model and the decoder was restricted to this model [1]. Using higher order n -gram language models may lead to better diacritization accuracy. In this work, we extend the dynamic programming decoding algorithm to support higher order language models. Higher order language models can be easily estimated using a language modeling toolkit. However, the search algorithm based on static lattices presented in [1] cannot be used to decode higher order n -gram language models. The new search algorithm will depend on dynamic lattices where the scores of different paths will be computed on the run time. Hence, the arcs scores can depend on the decoded history and higher order n -gram language models can be supported.

In Section 2, a mathematical formulation of the restoration of Arabic diacritics is described. The new search algorithm based on dynamic lattices is presented in 3. Sections 4 and 5 give experimental results on a public domain task and conclusions.

2 PROBLEM FORMULATION

A fundamental problem in natural language processing is to give a score to a sentence hypothesis or sequence of words. The assigned score for a sentence can be used to disambiguate between different solutions. Large vocabulary speech recognition systems (LVCSR) use a n -gram language model, which gives an approximate probability score of an

Table 1: Some possible word diacritics for an Arabic sentence.

يعلم	لم	ما	الانسان	علم
يَعْلَمُ	لَمْ	مَا	الْأَنْسَانُ	عَلِمَ
يُعْلَمُ			الْأَنْسَانُ	عِلْمٌ
يَعْلَمُ			الْأَنْسَانِ	عَلِمَ
يَعْلَمُ				عُلْمٌ

allowable word sequence $\mathbf{W} = w_0 w_1 \dots w_m$ in the recognition task. This probability score is calculated by accumulating local scores using $n - 1$ Markov chains over the word sequence and is given by

$$P(\mathbf{W}) = \prod_{i=1}^m P(w_i | w_{i-n+1}^{i-1}) \quad (1)$$

For bigram language model, This probability score is given by

$$P(\mathbf{W}) = \prod_{i=1}^m P(w_i | w_{i-1}) \quad (2)$$

and the $P(w_i | w_{i-1})$ is estimated by

$$P(w_i | w_{i-1}) = \frac{c(w_i, w_{i-1})}{c(w_{i-1})} \quad (3)$$

where $c(w_i, w_{i-1}), c(w_{i-1})$ can be computed from a large training corpus.

A. Diacritics restoration problem

In the context of Arabic restoration problem, the diacritized Arabic word sequence can be assigned a probability score based on a language model. This language model must trained on a large corpus of diacritized Arabic text. For example, Table 1 shows some possible diacritics alternative for the undiacritized sentence:

علم الانسان ما لم يعلم

Some of the possible hypotheses for this undiacritized sentence are:

عَلِمَ الْاَنْسَانَ مَا لَمْ يَعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يَعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ
 عِلْمٌ الْاَنْسَانَ مَا لَمْ يُعْلَمُ

Based on a language model, a score is assigned to each hypothesis. For example, given a bigram language model, the hypothesis $\text{عَلِمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ}$ is assigned a score as follow:

$$P(\mathbf{H}) = P(\text{عَلِمَ})P(\text{الْإِنْسَانَ}|\text{عَلِمَ})P(\text{الْإِنْسَانَ}|\text{مَا لَمْ})P(\text{مَا لَمْ})P(\text{لَمْ}|\text{يَعْلَمْ}) \quad (4)$$

Similarly, all possible hypotheses are scored in a similar way. The best path or the most likely diacritized word sequence is assigned to the hypothesis that has the highest score.

Using this statistical scoring method, Arabic diacritics restoration problem can be defined as a problem of choosing a word sequence $\hat{\mathbf{H}}$ with the *maximum language model score* given undiacritized word sequence \mathbf{W} :

$$\hat{\mathbf{H}} = \arg \max_{\mathbf{H}} P(\mathbf{H}|\mathbf{W}) \quad (5)$$

A brute force approach to score all hypotheses may be computationally expensive. An efficient algorithm to find the most likely hypothesis is discussed in the following section.

3 DYNAMIC DECODING

An efficient algorithm to find the most likely hypothesis is required to solve the search problem. If the average number of possible alternative diacritics per word is N and the sequence has L words, then the complexity of the search problem is related to $O(N^L)$. Hence, a brute force approach to solve the search problem does not scale.

Fortunately, it is possible to solve the search problem with complexity related to $O(NL)$ using a dynamic programming algorithm [1]. In this work, we extend the dynamic programming decoding algorithm to support higher order language models. The new search algorithm will depend on dynamic lattices where the scores of different paths will be computed on the run time. Hence, the arcs scores can depend on the decoded history and higher order n -gram language models can be supported.

The algorithm follows the following steps:

1. A dictionary for undiacritized words (keys) and diacritized word (values) is generated from the training corpus.
2. For an input sequence W_1, W_2, \dots, W_n , a lattice is created using the dictionary. Each node represents a possible diacritics option. For OOV input word (i.e. a word did not appear in the dictionary), a node is created with the undiacritized word. For example, Figure 1 shows a lattice for the Arabic sentence $\text{علم الانسان ما لم يعلم}$.
3. The nodes in the adjacent words W_n, W_{n-1} are connected by arcs, where n is a discrete time index. The initial nodes of the lattice at $n = 0$ are assigned a score computed as a log monogram probability of the diacritized word identifier of the node.
4. At $n = 1$, each arc (transition) has a score computed as a log probability of the word identifier of the node given the word identifier of the previous node (bigram probability score).
5. The most likely *partial* path is computed as in the Viterbi algorithm [11]. Using backtrack, it is possible to the history word sequence for each node.
6. For $n=2$ to $n=$ sentence length, each arc (transition) has a score computed as a log probability of the word identifier of the node given the history word sequence of the previous node. Go to step 5.

The presented algorithm is different from the algorithm described in [1] since the arcs scores are computed at each step using partial history computed as in Viterbi algorithm. We refer to this method as *dynamic lattice search*.

4 EXPERIMENTS

Automatic diacritics restoration experiments were carried out on the Arabic vocalized text corpus: Tashkeela [12]. The corpus is collected using automatic web crawling methods and it is free. It is collected from Islamic religious heritage books and it contains 6,149,726 words.

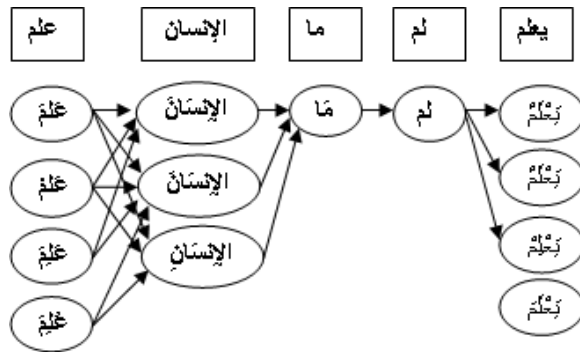


Figure 1: Generated lattice for the the undiacritized sentence علم الانسان ما لم يعلم.

Table 2: The properties of the training and test data of the Tashkeela corpus.

Property	Training data	Test data
Number of words	52500084	1902145
Out of Vocabulary (OOV)	-	6459

In this work, the undiacritized Arabic words and sentences were removed from the corpus. The corpus needs further text normalization steps which are beyond the scope of this work. For example, the word `الانسان` appears in corpus in two forms due to manual data entry errors. These two words should be normalized to one word. Correcting these mistakes may improve the statistics gathered from the data. The corpus was divided into training and test sets. The properties of the training and test data are summarized in Table 2. The system dictionary was built from the training data and its properties are detailed in Table 3.

The language model was built using the SRILM toolkit [13]. The maximum likelihood (ML) estimation of stochastic language model parameters is based on Equation (3). This estimation is problematic since it assigns zero probabilities to unseen n -grams in the training data. Smoothing aims to handle this problem by taking some probability mass from the observed n -gram and distribute it to the unseen n -grams [14], [15]. Applying smoothing techniques improve dramatically the results [16]. In this work, absolute discounting smoothing method was chosen as it leads to the best results. D parameter for absolute discounting was fixed to 0.5.

Several model orders were investigated. The model with order 2 contains 794,479 monograms and 10,847,630 bigrams. The correct word count and the diacritization Word Error Rate (WER) are shown in Table 4. The measure WER2 is computed by removing the last diacritic of the words (ignoring the case ending diacritic). Language model with order 4 leads to the best results. In addition, WER2 measure shows that most diacritization errors are related to the syntax of Arabic language.

5 CONCLUSIONS

In dynamic programming Arabic diacritics retraction, the possible word sequences with diacritics are assigned scores using statistical n -gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm.

In this paper, we develop a new search algorithm that supports higher order n -gram language models. The new search algorithm will depend on dynamic lattices where the scores of different paths will be computed on the run time. Hence, the arcs scores can depend on the decoded history and higher order n -gram language models can be supported. Preliminary results on a public domain corpus show that dynamic programming decoding based on higher order n -gram

Table 3: Tashkeela dictionary properties.

Property	Word count
unique diacritized words	794456
unique undiacritized words (keys)	419857

Table 4: Diacritization performance on the Tashkeela database.

Model order	Correct	WER	WER2
2	1725427	9.2%	3.6%
3	1731803	8.9%	3.4%
4	1732175	8.9%	3.4%

models can lead to better results than bigram models.

References

- [1] Y. Hifny, “Restoration of Arabic diacritics using dynamic programming,” *submitted to COLING*, 2012.
- [2] Y. Gal, “An HMM approach to vowel restoration in Arabic and Hebrew,” in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2002, pp. 66–73.
- [3] R. Nelken and S. M. Shieber, “Arabic diacritization using weighted finite-state transducers,” in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2005.
- [4] I. Zitouni, J. S. Sorensen, and R. Sarikaya, “Maximum entropy based restoration of Arabic diacritics,” in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 2006, pp. 577–584.
- [5] N. Habash and O. Rambow, “Arabic diacritization through full morphological tagging,” in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, ser. NAACL-Short ’07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 53–56. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1614108.1614122>
- [6] M. Alghamdi and Z. Muzaffar, “Kacst Arabic diacritizer,” in *The First International Symposium on Computers and Arabic Language*, 2007.
- [7] K. Shaalan, H. M. Abo Bakr, and I. Ziedan, “A hybrid approach for building Arabic diacritizer,” in *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, 2009.
- [8] M. Rashwan, M. Al-Badrashiny, M. Attia, S. Abdou, and A. Rafea, “A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 166 – 175, 2011.
- [9] Y. Hifny and et al., “ARABTALK: An implementation for arabic text to speech system,” in *Proceedings of the 4th conference of the Egyptian Society of Language Engineering (ESLE)*, October 2003.
- [10] D. Vergyri and K. Kirchhoff, “Automatic diacritization of Arabic for acoustic modeling in speech recognition,” in *COLING 2004 Computational Approaches to Arabic Script-based Languages*, 2004.
- [11] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. of IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [12] T. Zerrouki, “Tashkeela: Arabic vocalized text corpus,” 2011. [Online]. Available: <http://aracorpus.e3rab.com>
- [13] A. Stockle, “SRILM: the SRI language modeling toolkit.” in *Proc. ICSLP*, 2002.
- [14] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Harvard Univ., Computer Science Group, Cambridge, MA, Tech. Rep. TR-10-98, August 1998.
- [15] J. T. Goodman, “A bit of progress in language modeling,” Microsoft Research, Tech. Rep. MSR-TR-2001-72, 2001.
- [16] Y. Hifny, “Smoothing techniques for arabic diacritics restoration,” *submitted to ESOLE*, 2012.

IAN: An Automatic tool for Natural Language Analysis

Sameh Alansary^{*1}, Magdy Nagy^{**2}, Noha Adly^{**3}

** Department of Phonetics and Linguistics, Faculty of Arts, Alexandria University
El Shatby, Alexandria, Egypt*

**Bibliotheca Alexandrina, Alexandria, Egypt*

¹Sameh.alansary@bibalex.org

*** Computer and System Engineering Dept., Faculty of Engineering*

Alexandria University, Egypt.

²magdy.nagi@bibalex.org

³noha.adly@bibalex.org

Abstract— This paper presents IAN as an automatic tool for Natural language Analysis discussing and describing the tool in detail. The paper examines the tool from two aspects: the linguistic framework of IAN explaining the approach and theory adopted for analysis in this paper; the constituency-based approach, and the formal framework explaining the basic concepts behind the process of building the grammar and the types of grammar rules within IAN. Finally, it describes with examples how IAN can work as a linguistic analyser on all levels of linguistic analysis; tokenization, as well as morphological, syntactic and semantic analysis.

1 INTRODUCTION

Due to the importance of analyzed corpora, so many attempts have been carried out to analyze corpora morphologically and syntactically, examples are the International Corpus of English [1], the Corpus of German Newspaper Texts [2], the Czech National Corpus [3], the Prague Dependency Treebank [4], the Quranic Arabic Corpus [5], etc.. Several levels of analysis should be performed on a corpus in order to be used in practical natural language processing systems; morphological analysis, syntactic analysis, lexical analysis and semantic analysis. Morphological and syntactic analyses of corpora are more common than semantic analysis. Semantic analysis is helpful in understanding the underlying meaning of the document; it provides additional information in the form of metadata to process the document in an intelligent way. Semantic analysis is important for applications such as information extraction, question answering, machine translation and summarization.

The UNL (The Universal Networking Language) [6] is a semantically-based interlingua designed to break language barriers between people. The UNDL Foundation in cooperation with Bibliotheca Alexandrina have started an initiative for building a tool for linguistic analysis called IAN; the Interactive Analyzer. IAN is the natural language text analysis tool, it is an integrated environment for Natural language analysis. It works for all languages and on all levels of linguistic analysis with any linguistic theory. It simply employs the analysis grammar rules to analyze input and finally generate output through tokenization, morphological, syntactic and semantic analysis. IAN is the second generation text analysis tool following the first generation tool, the Enconverter [7]. UNL expressions are the final output of IAN. UNL expressions can also be obtained manually through the UNL editor tool [8]. In the UNL editor, semantic analysis of a natural language sentence can be reached in a single step. IAN is the automated version of the UNL editor, it takes the input sentence through all levels of analysis in order to finally output the UNL expression; the semantic representation of the natural language input.

This paper elaborates on the paper in [9], it is concerned with presenting and explaining IAN as an automatic tool for natural language analysis. It is divided into three sections; section 2 discusses the linguistic framework behind the design of IAN, section 3 is a detailed explanation accompanied with screenshots illustrating how IAN works, section 4 presents IAN as a linguistic analyzer. Finally, Section 5 concludes the paper.

2 IAN AS A TOOL FOR LINGUISTIC ANALYSIS

IAN [10] is an acronym for Interactive ANalyser. It is a natural language analysis system that represents natural language sentences as morphologically analyzed sentences, syntactic trees and semantic networks in the UNL format. In its current release, it is a web application developed in Java and available at the UNLdev¹, the application's interface is shown in figure (1). It is a part of the UNLdev which includes many other applications such as Eugene, SEAN, UNL editor. IAN users should be registered in the UNLweb in order to log in.

¹ <http://dev.undlfoundation.org/index.jsp>

IAN includes a grammar for natural language analysis and operates semi-automatically; word sense disambiguation is still carried out by the language specialist, however, the system can filter the candidates using an optional set of disambiguation rules. Syntactic processing is done automatically using the natural language analysis grammar, however, syntactic ambiguities are signaled to the user who may backtrack and choose a different syntactic path. In any case, human interaction is always optional and is used to improve the results. In case of no human intervention, the system simply outputs the most likely alternative, which is the one corresponding to the highest priority in the lexicon and in the grammar.

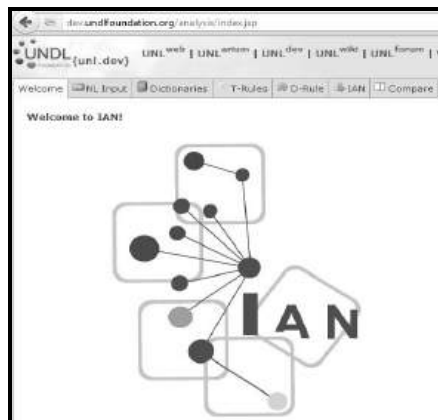


Figure 1: The interface of “IAN”

As a universal engine, IAN must be supplied with the following files in order to operate; each file is uploaded through the interface in figure (1):

- The input natural language document, from the “NL input” tab an input file can be uploaded, created, deleted, renamed, downloaded or shared with another user as in figure (2).

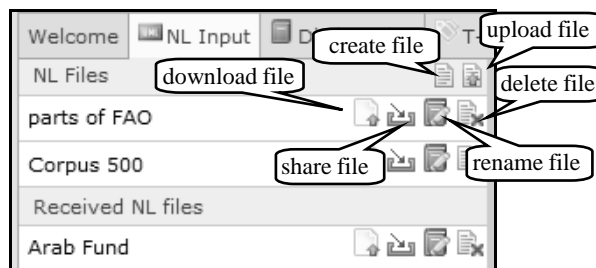


Figure 2: : The “NL tab” commands

- The NL-UNL (analysis) dictionary: it is a lexical database where UWs² are mapped with natural language entries, along with the corresponding features to be provided according to the UNL Dictionary Specs [11]. From the “dictionaries” tab in figure (3), users can control the dictionary files by the use of “control files commands”; he/she can create, upload, delete, download files...etc. The user can also search the dictionary via the user dictionary lookup tool by selecting either the headword, attribute or UW from the combo box as a search query. Other actions are also possible; users can select stop rules, delete rules, add rules,... etc.,

² UWs stand for Universal Words which are the vocabulary of the UNL and stand for language-independent concepts.

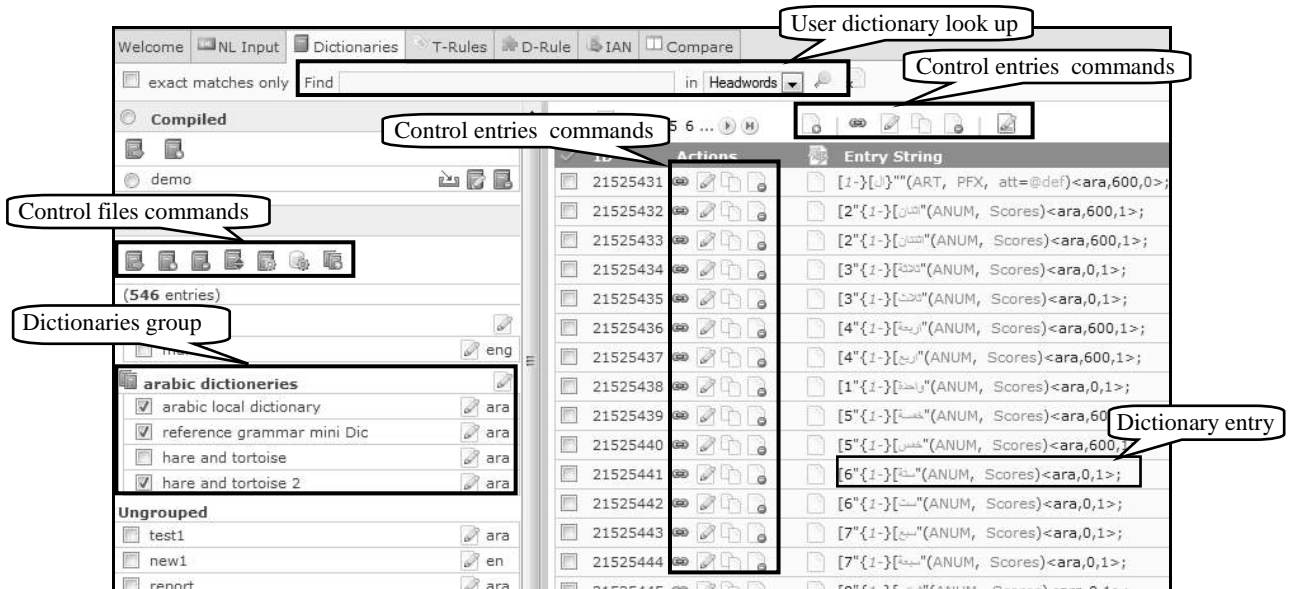


Figure 3: The “dictionaries tab” commands

- The NL-UNL (analysis) grammar: There are two types of grammar rules; transformation rules and disambiguation rules. IAN users can tackle each type; tackling transformation rules is through the “T-Rules” tab and disambiguation rules form through the “D-Rules” tab. Similar to the “Dictionaries tab”, users can control the rule files via the “control files commands”; he/she can create, upload, delete, download files,...etc. Users can also search for a rule by its string, action, condition or the rule ID. Other actions are also possible; users can stop rules, delete rules, add rules,... etc.

There are two other tabs in IAN’s interface; “IAN” and “compare” tabs. In the “IAN” tab in figure (4), the natural language text selected from the sentences navigation pane is processed by the selected dictionary and the selected rule files. From the output control pane, users can control the time the engine takes to process the selected sentences. Also, some options in the right pane are responsible for stopping processing, deleting the trace, exporting or flagging this trace. Users can control if they want the tokenization output only or wants to choose the word sense (WSD) manually. The output and trace appears in the area under the output control pane.

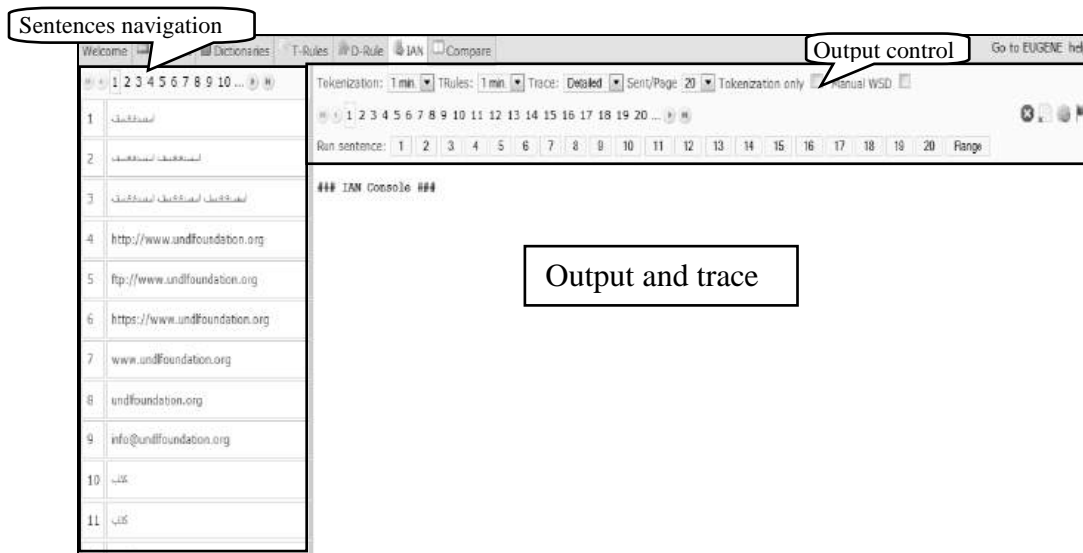


Figure 4: The “IAN” tab panes

The final tab is the “compare” tab which is able to compare the two final results as in figure (5).

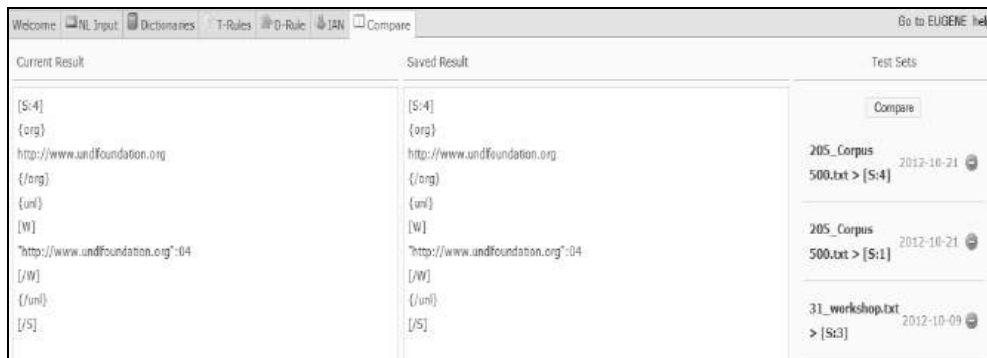


Figure 5: The “compare” tab

3 THE LINGUISTIC FRAMEWORK

IAN is a flexible linguistic description environment; it can provide a linguistic description for a Natural language text using any linguistic theory. On the text segmentation level, IAN can segment and tokenize a sentence from any point of view; for example; “the university of Alexandria” can be considered a single unit or four units: “the”, “university”, “ of “ and “Alexandria”. IAN enables users to control such segmentations.

On the syntactic level, any sentence can be analyzed using either the constituency based approach as shown in figure (6) or the dependency based approach as shown in figure (7).

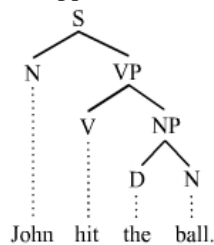


Figure 6: Constituency-based tree

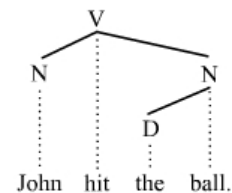


Figure 7: Dependency-based tree

IAN’s grammar environment allows both approaches for writing grammars using the rule types discussed in section 4. For the paper in hand, the used approach is the constituency based approach using the x bar theory [12].

The X-bar theory postulates that all human languages share certain structural similarities, including the same underlying syntactic structure. It stipulates that all the major phrase types are structured in the same way, as shown in figure (8):

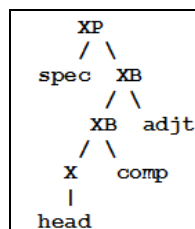


Figure 8: The general configuration of X-bar tree

In figure 7 above, *X* is the head, the nucleus or the source of the whole syntactic structure. The letter *X* is used to signify an arbitrary lexical category (part of speech). When analyzing a specific utterance, specific categories are assigned. Thus, the *X* may become a *N* for noun, a *V* for verb, a *J* for adjective, or a *P* for preposition. *comp* (i.e., complement) is an internal argument, i.e., a word, phrase or clause which is necessary to the head to complete its meaning (e.g., objects of transitive verbs).

adjt (i.e., adjunct) is a word, phrase or clause which modifies the head but which is not syntactically required by it. Removing an adjunct would leave a grammatically well-formed sentence. *spec* (i.e., specifier) is an external argument, i.e., a word, phrase or clause which qualifies (determines) the head. *XB* (X-bar) is the general name for any of the intermediate projections derived from *X*. Finally, *XP* (X-bar-bar, X-double-bar, X-phrase) is the maximal projection of *X*.

4 THE FORMAL FRAMEWORK

UNL is a language for computers that has the same properties of human languages; it has all the components corresponding to that of a natural language. It is composed of words expressing concepts called "Universal words", also referred to as UWs that are inter-linked with each other to form the UNL expressions of sentences. These links, called "relations", specify the role of each word in a sentence. The subjective meanings intended by the author are expressed through "attributes" [13]. Every language has its own grammar which describe and govern the linguistic behavior of structures of that language. UNL as a language for computers should have a grammar that describes language in a way computers can understand; this way is the UNL formal grammar. This section will discuss and explain the basic concepts behind the building of UNL grammar and the types of grammar rules.

A. Basic Definitions

UNL grammar expresses the meaning of sentences using some basic constituents; those constituents are nodes and relations. A node is the most elementary unit in the grammar. It corresponds to the notion of "lexical item", to be represented by dictionary entries. At the surface level, a natural language sentence is considered a list of nodes, and a UNL graph a set of relations between nodes. Nodes are enclosed between (parentheses), and may contain any of the following:

- A string, to be represented between "quotes". As for ("ing").
- A headword, to be represented between [square brackets], which expresses the original value of the node in the dictionary, as for ([book]);
- A UW, to be represented between [[double square brackets]], which expresses the UW value of the node, as for ([[book(icl>document)]]);
- A feature or set of features, which express the features of the node, as for (NUM);
- An index, preceded by the symbol %, which is used to reference the node, as for (%x);

In order to form a UNL graph, nodes are inter-related by relations. Inside each relation, nodes are isolated by a semicolon (;). In the UNL framework, there can be three different types of relations explained in table (1):

TABLE 1: THE DIFFERENT TYPES OF RELATIONS

Relation type	Definition	Form	format
Linear relations (L)	express the surface structure of natural language sentences.	binary	L(X;Y), or (X)(Y)
Syntactic relations	express the deep (tree) structure of the natural language sentences, they are not predefined, due to the free use of syntactic theory.	n-ary	rel(X;Y)
Semantic relations	Express the structure of UNL graphs, they constitute a predefined and closed set that stated in the UNL specs [14].	binary	rel(X;Y)

Nodes may contain one or more relations. In this case, they are said to be "hyper-nodes", and represent scopes or sub-graphs. The same as regular nodes, hyper-nodes contain a string, a headword, a UW, an index and features. When a hyper-node is deleted, all its internal relations are deleted as well. (1) is an examples of a hyper-node:

- (1) (("X")("Y")) - a hyper-node containing a linear relation between the nodes ("X") and ("Y").

Relations may have relations as arguments. In this case, they are said to be "hyper-relations", as in (2):

- (2) XP(XB(X;Y);Z) - a syntactic relation XP between the syntactic relation XB(X;Y) and the node Z.

B. Types of rules

In the UNL Grammar, there are two basic types of rules; transformation rules and disambiguation rules [15].

1) *Transformation rules*: Used to generate UNL graphs out of natural language sentences. They follow the general formalism in (3); where the left side α is a condition statement, and the right side β is an action to be performed over α .

$$(3) \alpha := \beta;$$

For IAN, there are five types of transformation rules that can be used in analyzing Natural language sentences on all linguistic levels, these types will be discussed in detail in the following. In brief, the original Natural language (NL) sentence is supposed to be preprocessed by the LL rules in order to become an ordered list. Next, the resulting list structure is parsed by the LT rules so as to unveil its surface syntactic structure, which is a tree. The tree structure is further processed by the TT rules in order to expose its inner organization, the deep syntactic structure, which is more suitable for semantic interpretation. Then, this deep syntactic structure is projected into a semantic network by the TN rules. The resultant semantic network is then post-edited by the NN rules in order to comply with UNL standards and generate the UNL Graph.

IAN automatically puts the Natural language input after dictionary look up in the form of a list of binary linear relations; for example, if the input is “X Y Z”, IAN will put it in the form in (4):

(4)

<SHEAD> L(X:01, " ":02) L(" ":02,Y:03) L(Y:03, " ":04) L(" ":04,Z:05) <STAIL>
--

The binary linear relations in (4) are enclosed inside “<SHEAD>” and “<STAIL>”; “SHEAD” expresses the sentence head (the beginning of input processing) while “<STAIL>” expresses the sentence tail (the end of input processing). The input “X Y Z” does not only include the nodes “X”, “Y” and “Z”, it also includes two blank spaces “ ”; thus, the input includes five nodes. The digits “:01”, “:02”, “:03”, “:04” and “:05” beside each node in (4) are IDs assigned automatically by IAN to each node in the list according to its order in the input. There are two blank spaces in the input, however, each has a different ID; “:02” for the second node and “:04” for the fourth node in the list.

Linear relations in (4) will be transformed into a syntactic (SYN) or semantic relation (SEM) with the help of the transformation rule types indicated in table 2. Table 2 shows each rule type, its function, properties and the syntax of writing the rule in the Example field.

TABLE 2: THE TYPES OF TRANSFORMATION RULES

Rule type	function	properties	Example
LL	Pre-editing the natural language sentence and preparing the input for the syntactic module.	Addition	(n1):=(n1)(n2);
		Deletion	(n):= - (n); or (n):=;
		Replacement	(n1):=(n2);
		Merging	(n1)(n2):=(n1&n2);
LT	Parsing the list structure into a tree structure.	Addition	(n1)(n2):=+SYN(n1;n2);
		Replacement	(n1)(n2):=SYN(n1;n2);
TT	Revealing the deep structure out of the surface structure.	Addition	SYN1(n1;n2):=+SYN2(n3;n4);
		Deletion	SYN1(n1;n2):=- SYN1(n1;n2); or SYN1(n1;n2):=;
		Replacement	SYN1(n1;n2):=SYN2(n3;n4);
TN	Deriving a semantic network out of a syntactic tree.	Addition	SYN (n1;n2):=+SEM (n3;n4);
		Replacement	SYN (n1;n2):= SEM(n1;n2);
TT	Post-editing the semantic network derived from the syntactic module	Addition	SEM1(n1;n2):=+SEM2(n3;n4);
		Deletion	SEM1(n1;n2):=- SEM1(n1;n2); or SEM(n1;n2):=;
		Replacement	SEM1(n1;n2):=SEM2(n3;n4);

2) *Disambiguation rules*: Used to improve the performance of transformation rules by constraining their applicability. They follow the general formalism in (5); where the left side (α) is a statement and the right side (P) is an integer from 0 to 255 that indicates the probability of occurrence of “ α ”.

$$(5) \alpha=P;$$

Disambiguation Rules are optional and may be used to: prevent wrong lexical choices, provoke best matches and check the consistency of the graphs, trees and lists. There are three types of disambiguation rules: list, tree and Network disambiguation rules.

- a) **List disambiguation rules:** apply over the natural language list structure to constrain the application of both tree-to-list (TL) and list-to-list (LL) transformation rules. They are also used for word selection. They have the following format in (6):

$$(6) (A)(B)=P;$$

- b) **Tree disambiguation rules:** apply over the intermediate tree structure to constrain the application of list-to-tree (LT), network-to-tree (NT) and tree-to-tree (TT) transformation rules. They have the following format in (7):

$$(7) SYN(A;B)=P;$$

- c) **Network disambiguation rules:** apply over the network structure of UNL graphs to constrain the application of tree-to-network (TN) and network-to-network (NN) transformation rules. They have the following format in (8):

$$(8) SEM(A;B)=P;$$

5 IAN AS A LINGUISTIC ANALYZER

IAN is an integrated environment for natural language analysis, it can analyze any Natural language on all linguistic levels from the surface level to the deep level. It can work as a tokenizer, morphological, syntactic and semantic analyzer. Grammar rules (section 2) are used for analysis, each type used in a certain level of analysis; disambiguation rules for tokenization, list-to-list rules are used for morphological analysis, list-to-tree and tree-to-tree for syntactic analysis and finally tree-to-network and network-to-network for semantic analysis. Tokenization and morphological analysis are dictionary design based processes; the language specialist is free to use a stem, word, or root based dictionary and design rules that suite his/her dictionary. The dictionary of the paper in hand is word-based. The following sub-section will discuss how IAN functions in each level of analysis.

Each level of analysis depends on the previous one; therefore, the syntactic analysis depends on the morphological analysis. In other words, the output of the morphological analysis is the input of the syntactic analysis and so on, this is shown in figure (9). The direction of arrows in figure (9) indicates the order of the levels of linguistic analysis; for example, the tokenization level takes place before the morphological analysis. In the following sub-sections both Arabic and English examples are used; Arabic examples are used to illustrate the morphological analysis due to the morphological richness of the language.

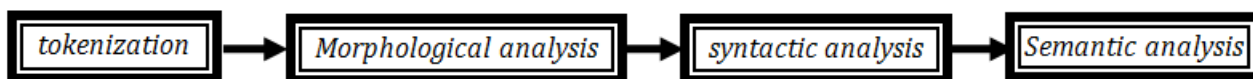


Figure 9: The order of linguistic analysis levels

A. IAN as a Tokenizer

Tokenization in general is the process of breaking up a text into units called tokens. In UNL terms, it is the process of segmenting the input into nodes. A token may be word, phrase, symbol, or other meaningful element. The list of tokens may be the input for further processing such as parsing. In IAN, the tokenization of the natural language input follows the following eight principles:

- 1- Tokenization is a dictionary-based process; the system tries to match the strings of the natural language input against the entries existing in the dictionary. In case it does not succeed, the string is treated as a temporary entry. There are no predefined token: spaces and punctuation signs have to be inserted in the dictionary in order to be treated as non-temporary entries. For instance, if the dictionary contains entries like [foundation], [un] and [or], the input will be tokenized as [www.] [un] [dl] [foundation] [.] [or] [g].
- 2- The tokenization algorithm goes from left to right. The system tokenizes the leftmost entries first. If the dictionary contains only two entries: [un] and [nd], with the same length and with the same frequency, the string "www.undlfoundation.org" will be tokenized as [www.][un][dlfoundation][.org], instead of [www. u][nd][lfoundation][.org], because [un] appears before [nd].
- 3- The tokenization algorithm tries to match first the longest string possible to entries in the dictionary. If the dictionary contains only two entries: [John], [Dr.] and [Dr. J], the string "Dr. John" will be tokenized as [Dr. J][ohn], instead of [Dr.][John], because [Dr. J] is longer than the string [Dr.].

- 4- The tokenization algorithm observes the frequency of the entries found in the dictionary; the most frequent entries have priority). The system observes the frequency numbers specified in the dictionary. If the dictionary contains only two entries: [un] and [nd], but the frequency of [nd] is higher than the frequency of [un], the string "www.undlfoundation.org" will be tokenized as [www. u][nd][lfoundation][.org], instead of [www.][un][dlfoundation][.org].
- 5- The tokenization algorithm observes the order of the entries in the dictionary (the system selects the first to appear in case of the same frequency). The system observes the order defined in the dictionary. If the dictionary contains only two entries: [un] and [nd],but [nd] appears first in the dictionary, the string " www.undlfoundation.org " will be tokenized as [www.u][nd][lfoundation][.org], instead of [www.][un][dlfoundation][.org]. .
- 6- The tokenization algorithm is case-insensitive, except in case of regular expressions. The string "a" will be matched by both [a] and [A], but the entry [/a/] will match only the string "a".
- 7- The tokenization algorithm assigns the feature TEMP (temporary) to the strings that were not found in the dictionary. In case of the absence of a dictionary entry, the system will consider the string " www.undlfoundation.org " a temporary UW and assigns it the feature TEMP.
- 8- The tokenization algorithm blocks tokens or sequences of tokens prohibited by disambiguation rules. If the dictionary contains entries such as [un], [foundation] and [or], the string "www.undlfoundation.org" will be tokenized as [www.] [un] [dl] [foundation] [.] [or] [g], which is incorrect, since the whole string should be treated as a single temporary entry. In order to block this sequence, the disambiguation rules in (9) prevent the engine from considering any word such as [foundation] a node unless it occurs after a blank space. Similarly, the disambiguation rule in (10) informs IAN that the prefix [un] with the feature (PFX) cannot be followed by a blank, and that the conjunction [or] with the feature (C) must be followed by a blank space by the disambiguation rule in (11).

(9) (POS)(^BLK)=0;

(10) (PFX)(BLK)=0;

(11) (C)(^BLK)=0;

The tokenized output from the tokenization process will be the input of the next process; the morphological analysis process.

B. IAN as Morphological analyzer

After tokenization,, the segmentation of the sentence into words or morphemes can take place. The Morphological analysis in IAN is responsible for transforming the different word categories (tokens) into the UNL format. Arabic is a highly inflectional language; for example, a single verb may have 146 inflectional forms such as the verb “أعطى” in figure (10):

The question here is, how can IAN and the analysis grammar express the inflected forms of this verb in UNL format?.

PA5&3P5&SNG&MCL&ACV= أَعْطَى	PA5&2PP&FEM&ACV&PLR= أَعْطَيْتَ	PR5&3PP&DUA&FEM&ACV&JUS= تَعْطِيَانِ
PSV&PA5&SNG&3P5&MCL= أَعْطَى	PSV&PA5&2PP&FEM&PLR= أَعْطَيْتَ	PR5&3PP&DUA&FEM&PSV&JUS= تَعْطِيَانِ
PR5&NOM&3P5&SNG&MCL&ACV= يَعْطِيَانِ	PR5&NOM&2PP&FEM&ACV&PLR= تَعْطِيَانِ	PR5&3PP&PLR&FEM&ACV&JUS= يَعْطِيَانِ
PSV&PR5&NOM&SNG&3P5&MCL= يَعْطِيَانِ	PSV&PR5&NOM&2PP&FEM&PLR= تَعْطِيَانِ	PR5&3PP&PLR&FEM&PSV&JUS= يَعْطِيَانِ
PA5&3PP&MCL&DUA&ACV= أَعْطَى	PA5&1P5&ACV&SNG= أَعْطَى	PR5&2P5&SNG&MCL&ACV&JUS= تَعْطِيَانِ
PSV&PA5&3PP&MCL&DUA&ACV= يَعْطِيَانِ	PSV&PA5&1P5&SNG= أَعْطَى	PR5&2P5&DUA&MCL&PSV&JUS= تَعْطِيَانِ
PA5&3PP&MCL&PLR&ACV= أَعْطَى	PSV&PR5&NOM&1P5&SNG= أَعْطَى	PR5&2P5&PLR&MCL&ACV&JUS= تَعْطِيَانِ
PSV&PA5&3PP&MCL&PLR= أَعْطَى	PA5&1PP&ACV&PLR= أَعْطَى	PR5&2P5&PLR&MCL&PSV&JUS= تَعْطِيَانِ
PR5&NOM&3PP&MCL&PLR&ACV= يَعْطِيَانِ	PSV&PA5&1PP&PLR= أَعْطَى	PR5&2P5&SNG&FEM&ACV&JUS= تَعْطِيَانِ
PSV&PR5&NOM&3PP&MCL&PLR= يَعْطِيَانِ	PR5&NOM&1PP&ACV&PLR= تَعْطِيَانِ	PR5&2P5&SNG&FEM&PSV&JUS= تَعْطِيَانِ
PA5&3PP&SNG&FEM&SNG= أَعْطَى	PR5&3P5&SNG&FEM&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&ACV&JUS= تَعْطِيَانِ
PSV&PA5&3PP&SNG&FEM&SNG= يَعْطِيَانِ	PR5&3P5&SNG&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&PSV&JUS= تَعْطِيَانِ
PR5&NOM&3P5&SNG&FEM&ACV= تَعْطِيَانِ	PR5&3PP&DUA&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&PSV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&3P5&SNG&FEM&SNG= تَعْطِيَانِ	PR5&3PP&DUA&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&ACV&ACC= تَعْطِيَانِ
PA5&3PP&FEM&ACV&DUA= أَعْطَى	PR5&3PP&DUA&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&PSV&ACC= تَعْطِيَانِ
PSV&PA5&3PP&FEM&DUA= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&PSV&ACC= تَعْطِيَانِ
PR5&NOM&3PP&FEM&ACV&DUA= تَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&ACV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&3PP&FEM&DUA= يَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&PSV&ACC= تَعْطِيَانِ
PA5&3PP&FEM&ACV&PLR= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&ACV&ACC= تَعْطِيَانِ
PSV&PA5&3PP&FEM&PLR= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&PSV&ACC= تَعْطِيَانِ
PR5&NOM&3PP&FEM&ACV&PLR= يَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&PSV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&3PP&FEM&PLR= يَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&ACV&ACC= تَعْطِيَانِ
PA5&2P5&MCL&ACV&SNG= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&ACV&ACC= تَعْطِيَانِ
PSV&PA5&2P5&MCL&SNG= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&PSV&ACC= تَعْطِيَانِ
PR5&NOM&2P5&MCL&ACV&SNG= تَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&ACV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&2P5&MCL&SNG= تَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&PSV&ACC= تَعْطِيَانِ
IMP&2P5&MCL&ACV&SNG= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&ACV&ACC= تَعْطِيَانِ
PA5&2PP&DUA&MCL&ACV= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&PSV&ACC= تَعْطِيَانِ
PSV&PA5&2PP&DUA&MCL= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&ACV&ACC= تَعْطِيَانِ
PR5&NOM&2PP&DUA&MCL&ACV= تَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&PSV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&2PP&DUA&MCL= يَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&ACV&ACC= تَعْطِيَانِ
IMP&2PP&DUA&MCL&ACV= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&PSV&ACC= تَعْطِيَانِ
PA5&2PP&DUA&MCL&PLR= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&ACV&ACC= تَعْطِيَانِ
PSV&PA5&2PP&DUA&MCL&PLR= يَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&PSV&ACC= تَعْطِيَانِ
PR5&NOM&2PP&DUA&MCL&PLR= تَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&ACV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&2PP&DUA&MCL&PLR= يَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&PSV&ACC= تَعْطِيَانِ
IMP&2PP&MCL&ACV&SNG= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&ACV&ACC= تَعْطِيَانِ
PA5&2P5&FEM&ACV&SNG= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&PSV&ACC= تَعْطِيَانِ
PSV&PA5&2P5&FEM&SNG= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&ACV&ACC= تَعْطِيَانِ
PR5&NOM&2P5&FEM&ACV&SNG= تَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&PSV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&2P5&FEM&SNG= تَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&ACV&ACC= تَعْطِيَانِ
IMP&2P5&FEM&ACV&SNG= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&SNG&FEM&PSV&ACC= تَعْطِيَانِ
PA5&2PP&DUA&FEM&ACV= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&ACV&ACC= تَعْطِيَانِ
PSV&PA5&2PP&DUA&FEM= أَعْطَى	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&FEM&PSV&ACC= تَعْطِيَانِ
PR5&NOM&2PP&DUA&FEM&ACV= تَعْطِيَانِ	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&ACV&ACC= تَعْطِيَانِ
PSV&PR5&NOM&2PP&DUA&FEM= يَعْطِيَانِ	PR5&3PP&PLR&MCL&ACV&ACC= يَعْطِيَانِ	PR5&2P5&DUA&MCL&PSV&ACC= تَعْطِيَانِ
IMP&2PP&DUA&FEM&ACV= أَعْطَى	PR5&3PP&PLR&MCL&PSV&ACC= يَعْطِيَانِ	PR5&2P5&PLR&MCL&ACV&ACC= تَعْطِيَانِ

Figure 10: The inflected forms of the Arabic verb "أعطى"

In Arabic verbs, the subject may be expressed in the verb form itself as a suffix; for example the suffix "وا" attached to the verb form "كتبوا" "they write" expresses the plural subject "هم" "they", the suffix "ن" attached to the verb form "كتبن" "they write" expresses the plural feminine subject "هن" "they". The subject may also be expressed without any suffixes attached to the verb form as in "كتب" "write", the subject here is the hidden singular pronoun "هو" "he". The inflectional forms in figure (10) express the both the concept of the verb itself and the concept of the subject.

Moreover, in the dictionary, grammatical attributes are assigned to the each verb form, these grammatical attributes provide information about the person, gender, number of the subject and also information about the voice and tense of the verb. Thus, in order to UNLize any Arabic verb, the morphological analysis rules should interpret the grammatical attributes assigned to the verb form and transform them into UNL format. For example, the Arabic verb "كتبوا" "they write" should be analyzed as shown in figure (11):



Figure 11: The morphological analysis of the verb "كتبوا"

The linguistic description shown in figure (11) can be realized through the grammatical attributes assigned to the verb "كتبوا" in the dictionary, these are shown in figure (12).

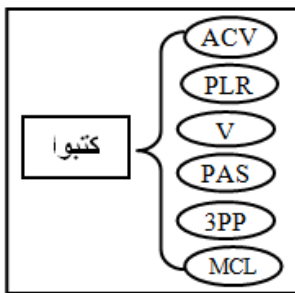


Figure 12: The grammatical attributes assigned to "كتبوا" in the dictionary

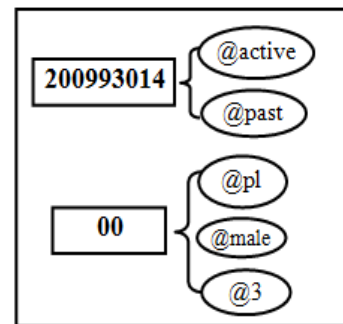


Figure 13: The UNL attributes of "كتبوا"

Morphological analysis rules are able to interpret each grammatical attribute in figure (12) and transform them into the suitable UNL attributes. They identify firstly the concept of the verb "كتب" from the dictionary which is expressed in the ID number

“200993014”, and then generate the attributes of the verb; for example, “@past” refers to the present tense (PAS) and “@active” refers to the active voice (ACV). Secondly, the morphological analysis rules make the concept of subject explicit and expresses it in UNL format as “00” and assigns it the necessary UNL person, number and gender attributes: @3, @pl and @male, respectively, as shown in figure (13).

Finally, the two independent concepts in figure (13) are represented in UNL format (shown in figure (14)) which is the output of the morphological analysis using IAN. Users of IAN can control the format of concepts; whether he/she wants to use Arabic words instead of IDs in order to better understand the output of morphological analysis.

```
[S:1]
{org}
كتبوا
{/org}
{unl}
[W]
200993014:01.@active.@past
[/W]
[W]
00:02.@3.@male.@pl
[/W]
{/unl}
[/S]
```

Figure 14: The UNL expression of the verb “كتبوا”

C. IAN as a Syntactic Analyzer

Syntactic analysis in IAN is the third level of linguistic analysis; the natural language sentence should be tokenized and morphologically analyzed before being subject to syntactic analysis. The two main advantages of IAN in relation to syntactic analysis are: first, it does not depend on a specific approach in functioning as discussed in section (3); it only depends on the set of rules describing the behavior of the language, as discussed in section (4). Second, IAN rules can output the surface and deep structures of the sentence, depending on the user’s purpose; the surface structure is the output of ordinary syntactic analyzers, on the other hand, the deep structure output is the input for semantic analysis.

The following two sub-sections will describe how both levels of syntactic analysis can be achieved using the X-bar theory; the first level is the surface structure analysis using the LT rule type and the second level is the deep structure analysis using the TT rule type.

- 1) *surface structure analysis (LT)*: In this type of analysis, small constituents or trees are constructed for small phrases (usually noun phrases) of the sentence and then combined to form a bigger tree gradually until the whole sentence is analyzed as a tree.

(12)

- a. (N, ^NB, %n) := (%n, NB);
- b. (J, %adjc)(NB, %n) := (NB(N, %n, J, +adjc), NB);
- c. (DET, %d)(NB, %n) := (NP(NB, %n, DET, %d, +spec), NP);
- d. (V, %v)(NP, %np) := (VB(V, %v, NP, %np, +comp), VB);
- e. (AUX, %I)(VB, ^VP, %vb)(STAIL) := (VP(%v, +empty, +spec)O
- f. (AUX, %I)(VB, %vb) := (IB(AUX, %I, VB, %vb, +comp), IB);
- g. (NP, %np)(IB, %ib) := (IP(IB, %ib, NP, %np, +spec), IP);

For instance, when analyzing a sentence like “the clever boy will understand the lesson”, rules start to link the nodes of the smallest constituents in the sentence. First, using the rule in (12a), IAN will project the nouns “boy” and “lesson” to the intermediate constituent “NB” as they are heads in noun phrases (figure (15)). Modifiers of nouns will be linked to the intermediate constituent “NB” in the following step; the adjective “clever” which precedes the “NB” will be considered its modifier, thus, the rule in (12b) will regard “clever” as an adjunct for the “NB” “boy”. Consequently, the feature +adjc is assigned to “clever”, and they are linked to a bigger “NB” as shown in figure (16).

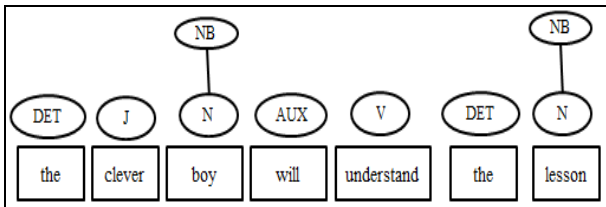


Figure 15: Projecting the head nouns from “NB”

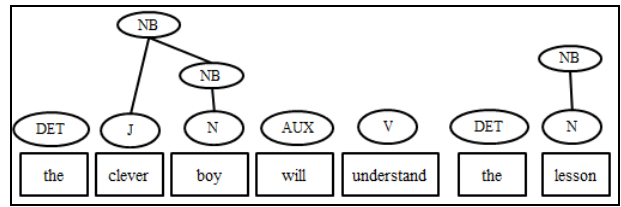


Figure 16: The syntactic tree for “clever boy”

As there are no other adjectives or modifiers for the head nouns, the “NB” of “clever boy” will be linked with the determiner (DET) “the”; the specifier of the “NP”, by the rule (12c) to form the maximal projection “NP” as in figure (17). As shown in figure (18), the second noun phrase “the lesson” will be constructed by the rule (12c) in the same manner as “the clever boy”.

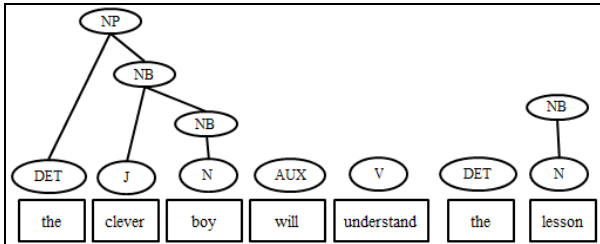


Figure 17: The syntactic tree for “the clever boy”

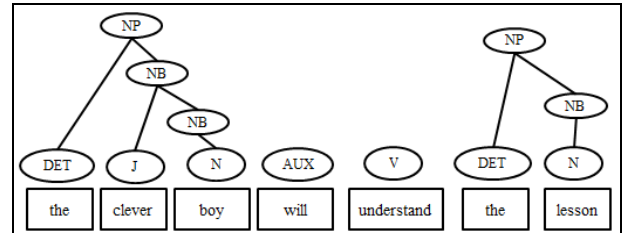


Figure 18: The syntactic tree for “the lesson”

A bigger intermediate constituent, the “VB” will be constructed by the rule (12d) as shown figure (19). This constituent links the NP “the lesson” with the head of the verbal phrase “understand”. Rule (12d) considers the “NP” after the verb a complement (i.e +comp assigned to the “NP”) to the verb “understand”, this is shown in figure (19) . The intermediate constituent VB will be projected to the maximal constituent “VP” as in figure (20).

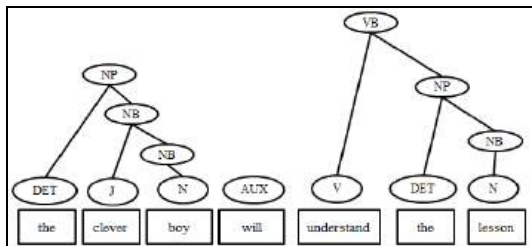


Figure 19: The syntactic tree for “understand the lesson”

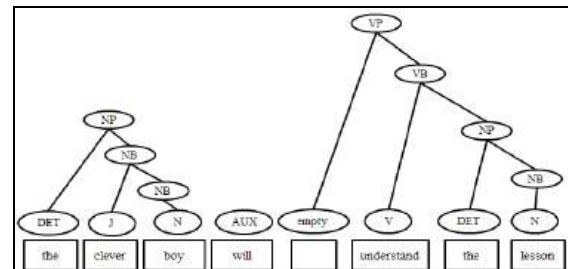


Figure 20: The syntactic tree for “understand the lesson” and reservation for the specifier position by the “empty” node

The “VB” does not have any adjuncts or specifiers; thus, it will be projected directly to the “VP” by the rule (12e). The position of the “VP” will be left empty; rule (12e) created a new node in the position of the specifier and assigned it the features (+spec) and (+empty). The auxiliary (AUX) “will” will be linked with the maximal projection “VP” “understand the lesson” to the intermediate constituent “IB” by the rule (12f) as shown in figure (21). The “VP” will be considered the complement of the auxiliary “will” which is the head of the inflectional phrase (I) since any tensed verb should exist inside an “IP”. The biggest tree will be formed by combining the “IB” “will understand the lesson” with the noun phrase “the clever boy” which is the specifier of the inflectional phrase “IP” as shown in figure (22).

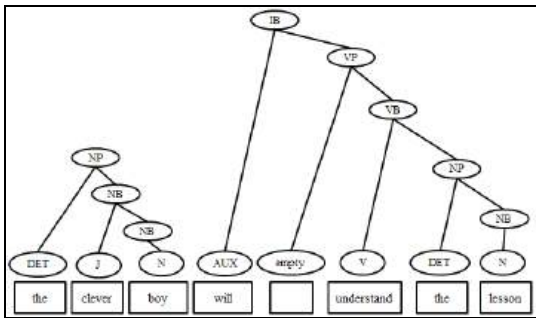


Figure 21: The syntactic tree for “will understand the lesson”

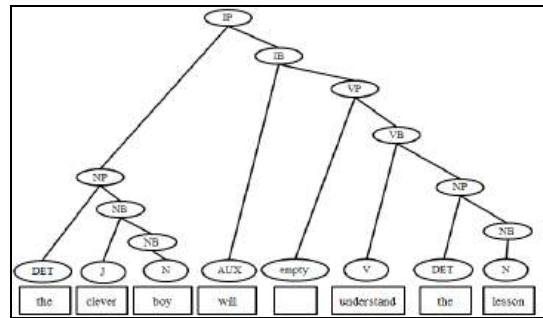


Figure 22: The syntactic tree for “the clever boy will understand the lesson”

Figure (23) is the surface syntactic structure output of IAN for the natural language sentence “the clever boy will understand the lesson” reached by applying the X-bar theory. This syntactic representation is sufficient to be an output of the syntactic analysis stage, otherwise, it can be the input for a deeper syntactic analysis; the deep structure syntactic analysis is discussed in section (C2), reaching the semantic network is discussed in section (d).

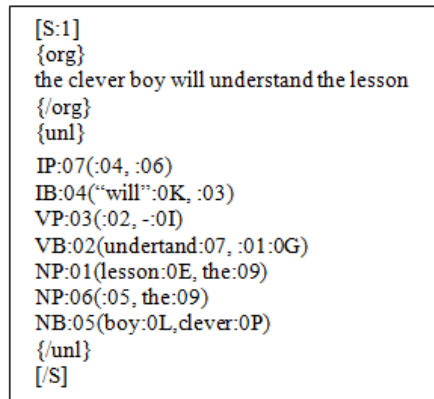


Figure 23: IAN’s output as a surface syntactic tree structure

2) *Deep structure analysis (TT)*: In this type of analysis, the input is the surface structure tree and the output is the head-driven structure of the sentence. For example, by using the output of the previous LT analysis as an input for TT analysis, the “NP” “the clever boy”, which is the specifier of the verb “understand”, will be connected with the intermediate constituent “IB” “will understand the lesson” by an “IP”. TT rules will derive the deep structure tree in figure (24) from the surface structure in figure (22).

- (13)
- a. $(IP(IB (;VP(e)); \%s \wedge e)) := IP(IB (;VP(\%s)); \%e + e);$
 - b. $(e) := ;$

TT rules restore the dependency relations between constituents which were isolated in the surface structure. There are four procedures in the restoration process: the first is movement; in the example in hand, using the rule in (13-a) the specifier of the “IP” “the clever boy” will be moved to the position of the specifier of the “VP” which was left empty in figure (16). The position of the “IP” specifier is left empty after movement, thus, the second procedure will take place; i.e., the deletion of empty nodes. Empty nodes in the tree become useless and are, thus, deleted by the rule in (13-b). The deletion of a node related to another one leads to the deletion of the relation between them. Hence, the “IP” relation will be deleted from the tree as shown in figure (24).

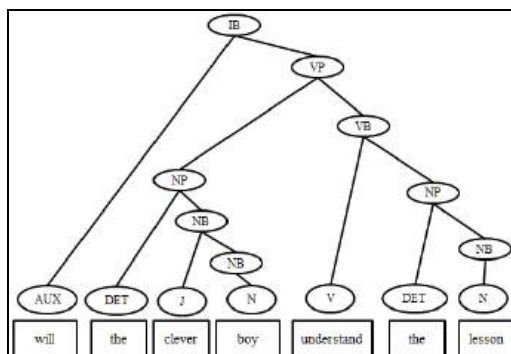


Figure 24: The deep syntactic tree structure after the movement of the specifier and the deletion of empty nodes

After reaching the deep structure, the third procedure ; de-arborisation, will de-arborise the syntactic tree structure in figure (24) into head-driven structures.

(14)

- a. $(IB(I, \%i; VP(VB(\%v; \%comp), \%x; NB(N, \%n; \%adjc), \%spec))) = (IC(\%i; \%v)VP(VB(\%v; \%comp), \%x; NB(N, \%n; \%adjc), \%spec));$
- b. $VP(VB(\%v; \%comp), \%x; NB(N, \%n; \%adjc), \%spec) = ((VB(V, \%v; \%comp), rel = \%x)(NB(N, \%n; \%adjc), rel = \%spec)VS(V, \%v; N, \%n));$

Rule (14a) will de-arborise the intermediate projection “IB” to the smaller constituent “VP” and the syntactic role “IC”; “VP” is the complement of inflectional phrase while “IC” is composed of “will” and the verb “understand” as shown in figure (25). Then, the maximal constituent VP between the intermediate constituent VB “understand the lesson” and the noun phrase “the clever boy” will be de-arborised into the syntactic role VS “verb specifier”, the intermediate constituent “VB” and the intermediate constituent “NB”. The “VS” is composed of the head of the verb phrase “understand” and the head of the noun phrase “boy” by the rule (14b), the intermediate constituent “VB” is composed of the head of the verb phrase “understand” and the head of the noun phrase “lesson” and, finally, the intermediate constituent “NB” is composed of the noun “boy” and the adjective “clever” as shown in figure (26).

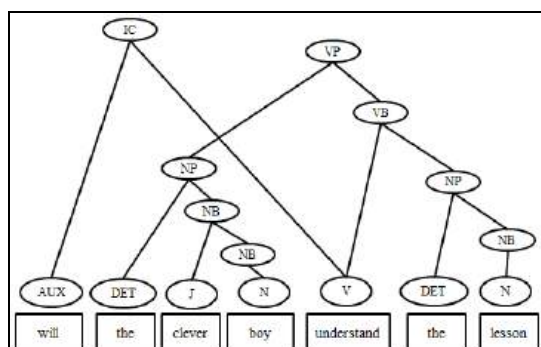


Figure 25: The de-arborisation of the “IB” to two smaller constituents “VP” and “IC”

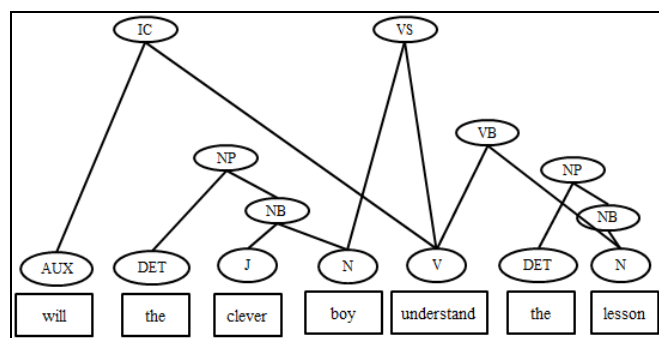


Figure 26: : The de-arborisation of the “IB” to two smaller constituents “VB” and “VS”

After de-arborisation, the fourth procedure will take place; Re-categorization. This procedure re-categorizes the resulting intermediate constituents from the de-arborisation process; in other words, maps each constituent to its syntactic role.

(15)

- a. $(VB(V, \%v; \%n, comp)) = VC(\%v; \%n);$
- b. $(NB(N, \%n; N, \%n2, spec)) = NS(\%n; \%n2);$
- c. $(NB(N, \%n; J, \%j, adjc)) = NA(\%n; \%j);$

The intermediate constituent VB will be mapped to the VC “verb complement” role between the head of the intermediate constituent “understand” and the head of the noun phrase “lesson” as in figure (27) by the rule (15-a). Then, the maximal projection NP between the intermediate constituent NB “clever boy” and the determiner “the” will be mapped to the syntactic role NS “noun specifier” between the head of the noun phrase “boy” and the determiner “the” by the rule (15-b) as shown in figure (28). The same rule (15-b) will apply to map the NP between the noun “lesson” (the head of the noun phrase) and the determiner “the” to NS as shown in figure (29).

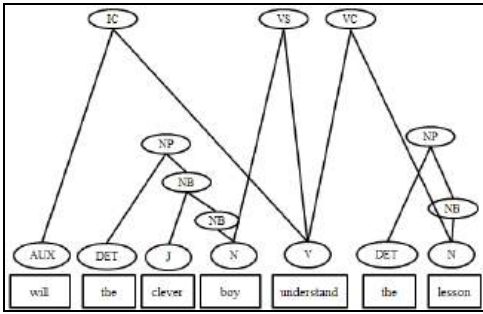


Figure 27: The re-categorisation of the “VB” to be “VC”

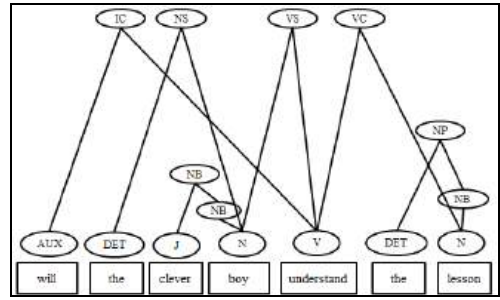


Figure 28: The re-categorisation of the “NB” to be “NS”

Then, the intermediate constituent NB will be mapped to NA “noun adjunct” between the head of the NB “boy” and the adjective “clever” by the rule (15c) as shown in figure (30).

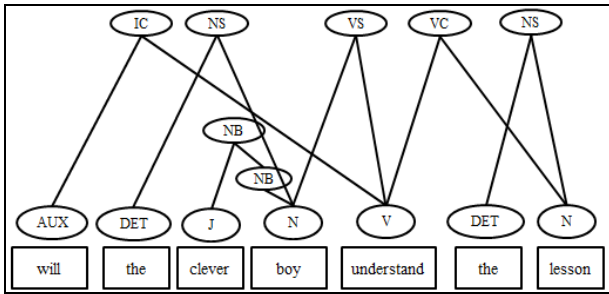


Figure 29: : The re-categorisation of the “NB” to be “NS”

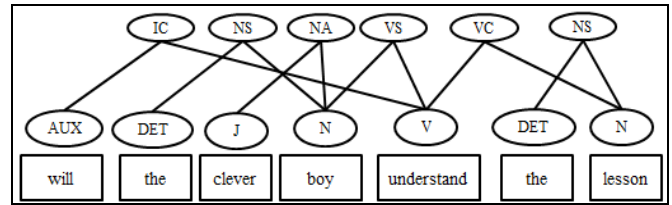


Figure 30: The re-categorisation of the “NB” to be “NA”

The output of IAN in figure (31) is the machine representation of figure (30) which represents the syntactic structure of the sentence.

```
[S:1]
{org}
the clever boy will understand the lesson
{/org}
{uml}
IC(will:08,understand:09)
VC(understand:09,lesson:0D)
NS(lesson:0D,the:06)
VS(understand:09,@future,boy:05)
NA(boy:05,clever:03)
NS(boy:05,the:04)
{/uml}
[S]
```

Figure 31: IAN’s output as a head-driven structure

D. IAN as a Semantic Analyzer

IAN is considered a semi-automatic tool only in semantic analysis; it allows for human intervention. The users of IAN can choose the suitable word-sense in order to improve the results in case the engine’s choice was inappropriate. To analyze the sentence semantically, its deep structure must be revealed first; therefore, the input of the semantic analyzer is the syntactic network in figure (31); the head driven structure.

- (16)
- VS(%v;%spec):=agt(%v;%spec);
 - VC(%v;%n,comp):=obj(%v;%comp);
 - NA(%n;%j,adjt):=mod(%j;%n);
 - NS(%n;%spec,DET,att):=(%n,att=%spec);
 - IC(%v;%I,att):=(%v,att=%I);

The semantic analyzer maps the syntactic roles with their equivalent semantic relations or UNL attributes [16] . The head-driven structure in figure (31) contains six syntactic roles; VS, VC, IC, NA, and two NSs. Since the verb “understand” is a

transitive verb that requires an agent and an object, the syntactic roles VS and VC will be respectively mapped to agent (agt) and object (obj) semantic relations by the rules (16a) and (16b). Then, the NA between the noun “boy” and the adjective “clever” will be mapped to a modifier (mod) relation by the rule (16c). Finally, the two NSs between “the” and “boy”, and “the” and “lesson” will be omitted and an attribute “@def” will be assigned to the two nouns “boy” and “lesson” by the rule (16d). The complete UNL graph or the final semantic representation for the present example is shown in figure (32). IAN’s final output is shown in figure (33).

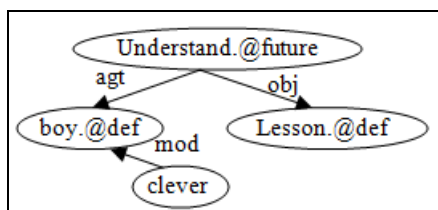


Figure 32: The UNL graph for “the clever boy will understand the lesson”.

```

[S:1]
{org}
the clever boy will understand the lesson
{/org}
{unl}
agt(understand:09.@future, boy:05.@def)
obj(understand:09.@future, lesson:0D.@def)
mod(boy:05.@def, clever:03)
{/unl}
[S]
  
```

Figure 33: IAN’s output for the semantic network.

6 CONCLUSIONS

This paper presented IAN as a pioneering effort for providing a tool for full natural language analysis. IAN has three main advantages; the first is its ability to analyze natural language texts morphologically, syntactically and semantically, the second is its language-independency, and the third is its adaptability to any linguistic approach or theory in analyzing languages. This paper is an invitation to all people around the world to register on the UNL web and test IAN and participate in its development and improvement by building their own language resources.

REFERENCES

- [1] G. Nelson, *The design of the corpus*. In: Greenbaum, S. (ed) *Comparing English Worldwide: the International Corpus of English*. Oxford: Clarendon Press, 27-35, 1996.
- [2] M. Kupietz , C. Belica , H. Keibel and A. Witt, *The German Reference Corpus DeReKo: A primordial sample for linguistic research*. In: Calzolari, N. et al. (eds.): *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC 2010)*, 2010.
- [3] K. Kučera, *The Czech National Corpus: principles, design, and results*. In: *Literary and Linguistic computing* 17(2), 245-257, 2002.
- [4] A. Bohmova, J. Hajic, E. Hajicova, and B. Hladka, *The Prague Dependency Treebank: three-Level Annotation Scenario*, In: Abeille, A. (ed) *Treebanks: Building and Using Syntactically Annotated Corpora*. Dordrecht: Kluwer, 2001.
- [5] K. Dukes and T. Buckwalter, *A Dependency Treebank of the Quran using Traditional Arabic Grammar*. In *Proceedings of the 7th International Conference on Informatics and Systems (INFOS)*. Cairo, Egypt, 2010.
- [6] H. Uchida and M. Zhu. *UNL2005 for Providing Knowledge Infrastructure*, in *Proceedings of the Semantic Computing Workshop (SeC2005)*, Chiba, Japan, 2005.
- [7] H. Uchida, *UNL: Universal Networking Language – An Electronic Language for Communication, Understanding, and Collaboration*, UNU/IAS/UNL Center, Tokyo, Japan, 1996.
- [8] S. Al-ansary, *UNL Editor: An Annotation tool for Semantic Analysis*, In *Proceedings of the 11th International Conference on Language Engineering*, Cairo, Egypt, 2011.
- [9] S. Al-ansary, M. Nagi, N. Adly, *UNL+3: The Gateway to a Fully Operational UNL System*, In *Proceedings of the 10th International Conference on Language Engineering*, Cairo, Egypt, 2010.
- [10] IAN application: <http://dev.undlfdoundation.org/analysis/login.jsp>,
- [11] UNL dictionary specs: http://www.unlweb.net/wiki/Dictionary_Specs
- [12] N.Chomsky, *Remarks on nominalization*. In: R. Jacobs and P. Rosenbaum (eds.) *Reading in English Transformational Grammar*, 184-221. Waltham: Ginn, 1970.
- [13] Z. Meiying, U. Hiroshi, *UNL annotation*. UNL Center,,UNDL Foundation website 2003 specifications and manuals, 2003.
- [14] The UNL semantic relations: <http://www.unlweb.net/wiki/Relations>
- [15] UNL grammar specs: http://www.unlweb.net/wiki/Grammar_Specs
- [16] The UNL attributes: <http://www.unlweb.net/wiki/Attributes>

A Baseline Speech Recognition System for Levantine Colloquial Arabic

Mohamed Elmahdy^{*1}, Mark Hasegawa-Johnson^{**2}, Eiman Mustafawi^{*3}

**Qatar University, Qatar*

¹mohamed.elmahdy@qu.edu.qa

³eimanmust@qu.edu.qa

***University of Illinois, USA*

²jhasegaw@illinois.edu

Abstract— The Arabic language is characterized by the existence of many different colloquial varieties that significantly differ from the standard Arabic form. In this paper, we propose a state-of-the-art speech recognition system for Levantine Colloquial Arabic (LCA). A fully continuous context dependent acoustic model was trained using 50 hours of speech from the BBN DARPA Babylon corpus. Pronunciation modeling was initially grapheme-based due to the absence of diacritic marks in transcriptions. Acoustic model parameters have been optimized including number of senones and Gaussians. In order to improve speech recognition accuracy, a cross-lingual hybrid acoustic and pronunciation modeling approach is proposed, where a MSA phoneme-based acoustic model is adapted using a small amount of LCA speech data. The adapted AM was then combined with the initial grapheme-based model to create a hybrid acoustic model.

1 INTRODUCTION

Arabic language is the largest still living Semitic language in terms of number of speakers. Around 250 million persons are using Arabic as their first native language and it is the 6th most widely used language based on the number of first language speakers.

Modern Standard Arabic (MSA) is currently considered the formal Arabic variety across all Arabic speakers. MSA is used in news broadcast, newspapers, formal speech, books, movies subtitling, and whenever the target audience or readers come from different nationalities. Practically, MSA is not the natural spoken language for native Arabic speakers. MSA is always a second language for all Arabic speakers. In fact, dialectal (or colloquial) Arabic is the natural spoken variety of Arabic in everyday life.

The majority of previous work in Arabic Automatic Speech Recognition (ASR) has focused on MSA whilst relatively little work has focused on dialectal Arabic. A significant problem in Arabic speech recognition is the existence of many different Arabic dialects. Every country has its own dialect and usually there exist different dialects within the same country. Dialects can be classified into two groups: Western Arabic and Eastern Arabic. Western Arabic can be sub-classified into Moroccan, Tunisian, Algerian, and Libyan dialects, while Eastern Arabic can be sub-classified into Egyptian, Gulf, Damascus, and Levantine.

The different Arabic dialects are only spoken and not formally written and significant phonological, morphological, syntactic, and lexical differences exist between the dialects and the standard form. This situation is called Diglossia and it has been discussed in [1, 2].

In this work, we are proposing an ASR system for Levantine Colloquial Arabic (LCA). LCA Arabic is the dialect of Arabic spoken by people in Lebanon, Syria, Jordan, and Palestine. Since the majority of dialectal Arabic speech resources are usually provided with graphemic transcriptions lacking diacritic marks, we have initially created a grapheme-based ASR system, where the phonetic transcription is approximated to be the word letters rather than the actual pronunciation.

In order to improve speech recognition accuracy, a cross-lingual hybrid acoustic and pronunciation modeling approach is proposed. First, a MSA phoneme-based acoustic model was trained. The MSA AM was then adapted using Maximum Likelihood Linear Regression (MLLR) followed by Maximum A-Posteriori (MAP) with a little amount of LCA speech data that has been phonetically transcribed. Afterwards, the adapted phonemic AM and the initial graphemic AM are fused together in order to create the hybrid model.

2 LEVANTINE COLLOQUIAL ARABIC SPEECH CORPUS

The BBN/AUB DARPA Babylon Levantine Arabic Corpus [6] is a corpus of controlled spontaneous speech. The corpus is recorded from subjects having Levantine colloquial Arabic as their native language. The subjects were from Lebanon, Syria, Jordan, and Palestine.

The corpus was recorded using a close-talking, noise-cancelling, headset microphone at 16kHz sampling rate. The subjects in the corpus were responding to refugee/medical questions like: (Where is your pain?, How old are you?, etc.), and were playing the part of refugees. Each subject was given a part to play, that prescribed what information they were to give in response to the questions. However, they were advised to express themselves naturally, in their own way, in Arabic. To avoid priming subjects to give their answer with a particular Arabic wording, the parts were given in English rather than Arabic.

The total number of recorded speakers is 164 with a vocabulary size of 14.7K unique words. The lexicon consists of only words in the graphemic form without phonetic transcription. Furthermore, we could not find any accurate pronunciation lexicon to map words to corresponding phonemes. Actually, this case is normal for all the Arabic dialects since they are only spoken and not written. That is why it is difficult to find a standard way to estimate the correct phonemes sequence for a given dialectal word.

The LCA corpus was sub-divided into a training set of ~50 hours (136 speakers) and a testing set of ~10 hours (28 speakers).

3 LANGUAGE MODELING

The language model is a statistical backoff tri-gram model with Kneser-Ney smoothing. The language model has been trained with the transcriptions of the LCA corpus training set (~230K words). The vocabulary size of the train set was found to be 13.6K unique words. The evaluation of the language model against the transcriptions of the testing set resulted in an OOV rate of 1.8%, tri-grams hits of 74.3%, and perplexity of 34 (entropy of ~5.1 bits) as shown in Table I. Language modeling in this research was carried out using the CMU-Cambridge Statistical Language Modeling Toolkit [3, 10].

TABLE I
LANGUAGE MODEL PROPERTIES AND EVALUATION AGAINST THE TRANSCRIPTIONS OF THE TESTING SET

Training words	230K
Vocabulary	13.6K
OOV	1.8%
Perplexity	34
Tri-gram hits	74.3%

4 SYSTEM DESCRIPTION

Our system is a GMM-HMM architecture based on the CMU Sphinx engine [4, 5]. Acoustic models are all fully continuous density context-dependent tri-phones with 3 states per HMM trained with MLE. The feature vector consists of the standard 39 MFCC coefficients. During acoustic model training, linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) were applied to reduce dimensionality to 29 dimensions. This was found to improve accuracy as well as recognition speed. Decoding is performed in multi-pass, a fast forward Viterbi search using a lexical tree, followed by a flat-lexicon search and a best-path search over the resulting word lattice.

5 GRAPHEME-BASED ACOUSTIC MODELING

Grapheme-based acoustic modeling (also known as graphemic modeling) is an acoustic modeling approach where the phonetic transcription is approximated to be the word letters rather than the exact phonemes sequence. Short vowels and germinations are assumed to be implicitly modeled in the acoustic model. Each letter was mapped to a unique model resulting in a total number of 36 base units (letters in the Arabic alphabet). In this case, pronunciation modeling is a straightforward process, so for any given word, pronunciation modeling is done by splitting the word into letters. In this case, each word is associated with only one graphemic pronunciation variant.

The 50 hours LCA training set was used for acoustic modeling. The acoustic model consists of both context-independent (CI) and context-dependent (CD) phones. During decoding, CI models are used to compute likelihood for tri-phones that have never been seen in the training set. The graphemic acoustic model consists of 108 CI states. In order to determine the optimized number of CD tied-states (senones) and the number of Gaussians per state, several acoustic models have been created with varying number of senones (500 to 4500) and varying number of Gaussians (4 to 128). For each setting, decoding was performed over the testing set as shown in Table II.

The optimized number of senones and Gaussians per state were found to be 2000 and 64 respectively, resulting in an absolute WER of 30.5 % as shown in Table II.

TABLE III
WORD ERROR RATE (WER) % ON THE LCA TEST SET, OPTIMIZING THE TOTAL NUMBER OF TIED-STATES (TS) AND
GAUSSIAN DENSITIES

Gaussians	Tied states (TS)								
	500	1000	1500	2000	2500	3000	3500	4000	4500
4	44.1	41.8	41.7	40.1	38.6	39.7	38.8	40.2	38.5
8	39.8	38.1	37.3	36.0	36.2	36.4	34.7	36.1	33.3
16	38.2	35.7	34.7	34.8	33.5	32.4	32.6	33.2	32.9
32	36.3	34.4	33.1	33.4	32.4	33.6	33.6	32.3	34.5
64	34.5	33.1	32.8	30.5	32.4	33.1	32.4	32.8	34.1
128	32.8	32.1	31.3	32.0	33.3	34.3	35.5	37.8	40.2

In order to improve accuracy, we have applied linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) to reduce dimensionality from 39 to 29 dimensions. This was found to decrease WER with 1.3% relative as shown in Table III. Since we are training a graphemic model, cross-phone training was applied and this was found to further decrease WER to 28.8% absolute (-5.6% relative to the baseline) as shown in Table III.

TABLE IIIII
WER (%) ON THE LCA TEST SET, COMPARING FEATURE TRANSFORMATION AND CROSS-PHONE TRAINING RELATIVE TO THE BASELINE AM

	WER	Relative
Baseline	30.5%	-
Feature transform (FT)	30.1%	-1.3%
FT + Cross-phone	28.8%	-5.6%

6 HYBRID ACOUSTIC MODELING

Hybrid acoustic modeling is performed by two independent acoustic model trainings (phonemic model and another graphemic model). Afterwards, the two models are fused together into one hybrid model [8].

Since we could not obtain large amounts of phonetically transcribed speech data for Levantine Arabic, we have adopted a cross-lingual approach to train the phonemic AM using MSA speech data in a similar way as described in [9]. The MSA phonemic AM was trained using 62 hours of speech data from two corpora provided by ELRA:

- The NEMLAR Broadcast News Speech Corpus (~40 hours) [11].
- The NetDC Arabic BNSC (Broadcast News Speech Corpus) (~22 hours).

Both corpora are provided with fully vocalized transcriptions, and therefore grapheme-to-phoneme is almost a one-to-one mapping. The phonemic MSA AM was then adapted using MLLR adaption followed by MAP along with 100 utterances from the LCA train set. Phonetic transcriptions for these 100 utterances were manually prepared.

Phonemic pronunciation variants were generated using the LDC Standard Arabic Morphological Analyser (SAMA) [7]. Pronunciation modeling was hybrid by generating a graphemic variant along with all possible phoneme variants generated by the morphological analyzer. Decoding results of the LCA test set shows that we can achieve an improved accuracy of 28.4% absolute WER with a 1.3% relative reduction compared to the baseline. It was noticed that many pronunciation variants that are generated from the morphological analyzer differ from the actual LCA pronunciation only in vowels. e.g. /i/ realized as /e/. That is why our next step is to normalize vowels in the output from the morphological analyzer as well as the MSA phonemic AM.

TABLE IVV
WER (%) ON THE LCA TEST SET, COMPARING GRAPHEMIC AND HYBRID ACOUSTIC MODELING

	WER	Relative
Graphemic AM	28.8%	-
Hybrid AM	28.4%	-1.3%

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an ASR system for Levantine Colloquial Arabic (LCA). First, a grapheme-based acoustic model was trained using 50 hours of LCA speech data. The optimized number of senones and Gaussians densities were found to be 2000 and 64 respectively. Batch decoding of the test set (10 hours) resulted in a WER of 30.5%. Feature transformation using LDA and MLLT was applied to reduce dimensionality resulting in WER reduction of 1.3% relative. Cross-phone training was found to add further reduction in WER achieving 5.6% relative to the baseline.

A cross-lingual phonemic acoustic model for LCA was prepared by adapting existing MSA acoustic model with little LCA speech data along with manually prepared phonetic transcription. A hybrid acoustic model was created by combining the graphemic and the cross-lingual phonemic models. Hybrid acoustic modeling resulted in a further reduction in WER of 1.3% relative (28.4% absolute).

For future work, the proposed hybrid approach will be extended and evaluated with the other Arabic colloquials (e.g. Egyptian, Iraqi, Gulf, etc.).

ACKNOWLEDGMENT

This publication was made possible by a grant from the Qatar National Research Fund under its National Priorities Research Program (NPRP) award number NPRP 09-410-1-069. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Qatar National Research Fund.

We would like also to acknowledge the Linguistic Data Consortium (LDC) and the European Language Resources Association (ELRA) for providing us with the required speech and text resources to conduct this research.

REFERENCES

- [1] A.S. Kaye, "Modern Standard Arabic and the Colloquials", *Lingua* 24, pp. 374-391, 1970.
- [2] C. Ferguson, "Diglossia", *Word* 15, pp. 325-340, 1959.
- [3] Carnegie Mellon University-Cambridge, CMU-Cambridge Statistical Language Modeling toolkit, <http://www.speech.cs.cmu.edu/SLM/toolkit.html>
- [4] Carnegie Mellon University Sphinx, Speech Recognition Toolkit, <http://cmusphinx.sourceforge.net/>
- [5] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices", In *Proceedings of ICASSP*, vol. 1, pp. 185-188, 2006.
- [6] J. Makhoul, B. Zawaydeh, F. Choi, and D. Stallard, "BBN/AUB DARPA Babylon Levantine Arabic Speech and Transcripts", *Linguistic Data Consortium(LDC)*, LDC Catalog No.: LDC2005S08, 2005.
- [7] M. Maamouri, D. Graff, B. Bouziri, S. Krouna, A. Bies, and S. Kulick, "LDC Standard Arabic Morphological Analyzer (SAMA) Version 3.1", *Linguistic Data Consortium(LDC)*, LDC Catalog No.: LDC2010L01, 2010.
- [8] M. Elmahdy, M. Hasegawa-Johnson, and E. Mustafawi, "Hybrid Pronunciation Modeling for Arabic Large Vocabulary Speech Recognition", *Qatar Foundation Annual Research Forum*, 2012.
- [9] M. Elmahdy, R. Gruhn, and W. Minker, "Cross-Lingual Acoustic Modeling for Dialectal Arabic Speech Recognition", In *Proceedings of Interspeech*, pp. 873-876, 2010.
- [10] P. Clarkson, and R. Rosenfeld, "Statistical Language Modeling Using the CMU-Cambridge Toolkit", In *Proceedings of ISCA Eurospeech*, 1997.
- [11] The Nemlar project, <http://www.nemlar.org/>

المعجم العربي الحديث

أحمد محمد متولي
amt@sakhr.com

مصطفى رمضان
msr@sakhr.com

حمدي سليمان مبارك
hamdys@sakhr.com

قسم أبحاث اللغة العربية
شركة صخر لبرامج الحاسب
القاهرة - جمهورية مصر العربية

ملخص

كانت العرب من أسبق الأمم إلى النشاط المعجمي، وقد سار تطور التأليف المعجمي مسيرة طبيعية، إذ تنوعت طرائق اللغويين كثيرًا في بدايات هذا النوع من التأليف، حتى وصلوا إلى هذا النمط الطبيعي السهل في ترتيب ألفاظ اللغة، وذلك لأن السهولة والبساطة هما الكمال بعينه، ولهذا فهما من أصعب الأشياء وأرقاها، لكن هذه المعاجم غدت بعيدة عن مقتضيات العصر وما تتطلبه وسائل البحث الحديثة من سهولة ووضوح وقرب مأخذ؛ ولذا تقدم صخر تجربتها في بناء معجم جديد يتعامل مع اللغة العربية الحديثة محاولة أن تلبي احتياجات متحدثي العربية وطلابها من خلال أسلوب جديد ارتأت فيه السهولة والوضوح في تقديم المعنى والمثال والبيانات اللغوية المختلفة للكلمات العربية والمصطلحات الحديثة.

الكلمات المفتاحية

(المعجم، مدخل معجمي، مدونة لغوية، بيانات صرفية، بيانات نحوية، بيانات معجمية، تراكيب اصطلاحية، متلازمات)

مقدمة

يقول أحد الباحثين " المعجم هو المضمار الذي تتمثل فيه علاقة اللفظ بالمعنى، وهو المرجع الذي يستوي في الحاجة إليه الناشئ والمتعلم والباحث المنقب، وتنوع المعاجم لدى الأمة وتجدها من حين إلى آخر وذيوع استعمالها بين الأفراد دليل على حيوية هذه الأمة وحيوية لغتها، وعلى هذا فالمعجم أداة من أدوات الثقافة الهامة والمرآة التي تعبر عن مستوى الارتقاء الثقافي في مجتمع ما فإذا كان هذا المعجم محشواً بمعلومات غير دقيقة، أو إذا كان يفتقر إلى النظام والمنهجية والدقة فهذا يعني أننا ندخل كل هذه المساوئ في عقول أبنائنا، وهذا يعني من جهة ثانية أننا نقدم صورة مشوهة عن ذواتنا وثقافتنا وترائنا".

إن التطور العلمي والحضاري والمعرفي الذي وصل إليه العالم اليوم، والتقدم الهائل في وسائل البحث وأدواته، يضاف إلى ذلك الحاجة الشديدة إلى اختصار الوقت والحرص عليه؛ أمام التسارع الكبير في

الاكتشافات والاختراعات، والزيادة الهائلة فيما ينشر في العالم اليوم، وحاجة الباحثين إلى متابعة ما أمكن منه، كل ذلك وغيره يتطلب إعادة النظر في المعاجم، والعمل على إعداد معجم عربي عصري؛ يواكب النهضة المعاصرة، وفي بمتطلباتها، ويعين الباحثين، ولا يستهلك وقتاً طويلاً عند الرجوع إليه، ويشتمل على جميع الألفاظ، ويعتمد منهجاً دقيقاً في إيراد المعاني وترتيب الألفاظ، ويفيد من التطور التقني لتسهيل عملية الرجوع إليه.

واقع المعجمية العربية

يقول: أ. د. محمد حلمي هُلَيْل في بحوث الاجتماع الثاني لخبراء المعجم الحاسوبي التفاعلي للغة العربية، الرياض 2008: "تعاني معاجمنا العربية التي يرجع إليها ابن اللغة أو متعلمها أو مترجمها - سواء لفهم العربية أو الكتابة بها أو ترجمتها إلى لغات أخرى - من قصور شديد حتى أننا لا يمكننا بحال من الأحوال أن نسميها معاصرة أو حديثة هذا إذا ما قارناها بالمعجمات في اللغات الحية الأخرى مثل الإنجليزية والفرنسية والألمانية" ولقد تجلت الصعوبات التي تواجه المعجمية العربية في النقاط الآتية:

1. اختيار المداخل وترتيبها: سرد المداخل المعجمية - سواء أكانت قديمة أم حديثة بدون تفريق بينها - أو النص على قدم أو حداثة الاستخدام، فنجد في معظم المعاجم - على تفاوت بينها - بعض المواد والمداخل القديمة المهجورة التي لا يكاد يستعملها أحد.
2. غياب المشتقات الخاصة بالكلمة حيث إن المعاجم العربية كلها لا تأتي على الكلمات المشتقة من الجذور مع تصريفاتها وأشكالها إذ تفترض أن المستخدم يعرف قواعد الاشتقاق والتصريف والحقيقة أنه ليس كذلك في الوقت الحاضر.
3. ترتيب المعاني: في داخل المادة الواحدة حسب شهرة استخدام المعنى وكثرة وروده.
4. عدم ربط المعنى بالمشتقات المستخدمة منه :

المعنى	المشتقات المستخدمة
قَتَلَ: الإنسان ونحوه: أماته	قَتَلَ، قُتِلَ، يَفْتُلُ، يُفْتَلُ، أَقْتَلُ، قَتَلْتُ، مَقْتُلٌ، قَاتِلٌ، مَقْتُولٌ، قَتِيلٌ، قَتَالٌ، قَتَلَةٌ
قَتَلَ: الموضوع بحثاً: درسه من جميع جوانبه	قَتَلَ، قُتِلَ، يُفْتَلُ

5. عدم التركيز على المترادفات والمتضادات وإهمالها أثناء تعريف المعنى ، مع أن لها تأثيرا كبيرا في تبيان المعنى وتوضيحه كما أن المترادفات يتم استغلالها في طرق البحث المتقدمة.
6. هناك قصور كبير في تغطية الكثير من ألفاظ الحضارة والمصطلحات العلمية وأسماء النباتات والحيوانات وما شابهها والكثير من المعاني المستحدثة مثل (دَوْلَرَة، فَوْتَرَة، عَوْرَبَة، تَرْيِيف، أُخُوْنَة، خَوْصَصَة) .
7. عدم ذكر المعاجم لكثير من السمات النحوية والصرفية والدلالية الضرورية للكلمة أو التنويه عليها مثل الإعراب والبناء، والصرف وعدمه، والميزان الصرفي، وقابلية الكلمة للتعريف والتذكير، والتأنيث، والسوابق واللواحق وصيغ التصغير والنسب.
8. عدم النص بصورة وافية على حروف الجر التي ترتبط دلاليا بالكلمة
مثل : سَعَى فلان إلى كذا : عمل له
حروف الجر (إلى / لـ / في / على / نحو / وراء)
تَحَدَّثَ فلان عن كذا : تكلم عنه
حروف الجر (في / عن / حول / على)
9. تجاهل البيانات الخاصة باستخدام المعنى الحقيقي أو المجازي :

حقيقي	نَسَجَ: الثوب حاكه
مجازي	نَسَجَ: الأمر / كذا على منوال : أَلْفَه
حقيقي	تَشَمَّمَ: الشيء : شمه في مهل
مجازي	تَشَمَّمَ: الخبر : تطلبه والتمس معرفته
حقيقي	صَفَّى: كذا: أزال عنه شوائبه" صفى الماء "
مجازي	صَفَّى: كذا : أنهاه " صفى مشاكله "
مجازي	صَفَّى: فلان خصومه وأعداءه : قتلهم .

10. عدم ربط كل معنى من معاني المعجم بسياقات طبيعية من المكنز العربي الحديث المتمثل في(الكتب والصحف والمجلات وغير ذلك من المصادر المنتشرة على الإنترنت) بالإضافة إلى النصوص التراثية

(القرآن والحديث والأمثال والشعر) إن وجدت بحيث تتضح المعاني والتراكيب التي تستخدم فيها، وأساليب استخدامها.

11. غياب المعلومات الصوتية :

حيث إن المعلومات الصوتية ضرورية عندما يُخشى اللبس أو التحريف أو عندما تكون الكلمات حوشية غريبة أو عندما يكون للكلمة عدة أنواع من الشكل للمعنى ذاته أو للدلالة على معانٍ مختلفة.

المعجم العربي الحديث ضرورة يفرضها الواقع اللغوي

وأمام هذا الواقع اللغوي تقدم صخر معجمها الجديد الذي تضع فيه خبرتها الطويلة التي تتجاوز ربع قرن من الزمان في المعالجة الآلية للغة العربية أملة أن يلبي حاجة المستخدم العربي بمختلف توجهاته قارئاً ومتعلماً و مترجماً.

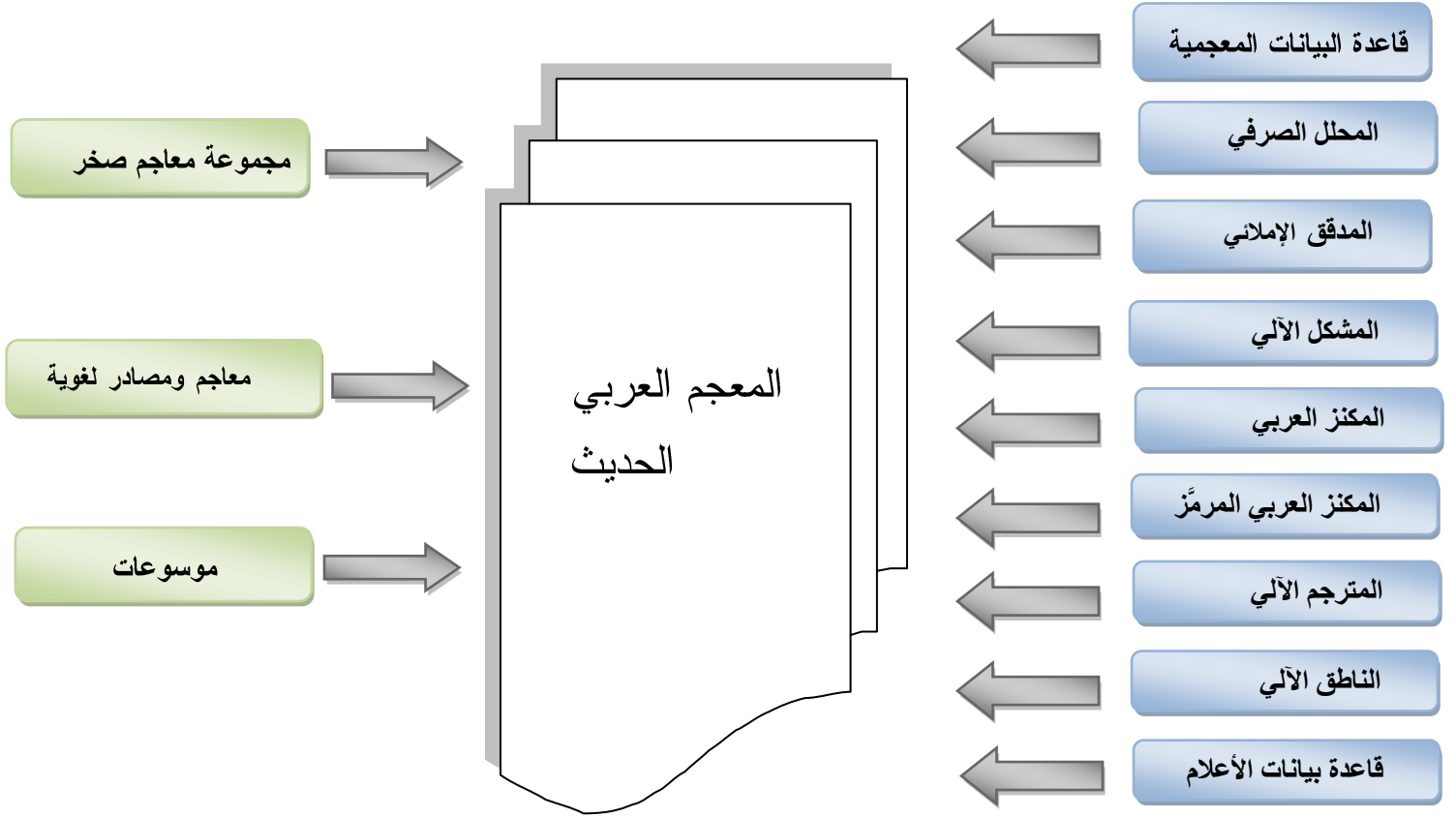
مصادر المادة المعجمية

يعتمد المعجم العربي الحديث في بناء مادته على اللغة المستخدمة، معتمداً في ذلك على المدونات اللغوية. "المدونة" أو (المكنز) في الألسنية الوصفية الحديثة هي مجموعة معينة من النصوص المكتوبة أو المنطوقة تؤخذ كأساس لبناء المعجم، وغايتها تحقيق توازن وكفاية في التوزيع النسبي للنصوص على النحو الآتي:

- التنوع الموضوعي بحيث تمثل مختلف ميادين المعرفة العامة المتوفرة بالعربية.
- التنوع المكاني بحيث تغطي معظم الأقطار التي تتحدث هذه اللغة.

أدوات تطوير المعجم العربي الحديث

وضعت صخر خبرتها اللغوية الطويلة في معالجة اللغة العربية ممثلة في مجموعة من الأدوات والتقنيات الحديثة التي تنفرد بها على المستوى الإقليمي والعالمي في سبيل إخراج معجمها المتميز في أكمل صورة لتلبية حاجة مستخدميه على اختلاف ثقافتهم ودرجة إتقانهم للغة، تمثلت هذه الأدوات فيما يلي:



- قاعدة البيانات المعجمية هي نتاج عمل أبحاث عشرين عاما اعتمدنا في بنائها على العديد من المعاجم العربية الحديثة مثل "المعجم الوسيط، المعجم العربي الأساسي، المورد، المنجد، معجم اللغة العربية المعاصرة... الخ"، بعد استبعاد المعاني التي لا ترد في الكتابات الحديثة.
- مدونة لغوية مجمعة من مصادر متنوعة (كتب، موسوعات، مقالات صحفية منشورة في مجلات، وصحف، ووكالات أنباء) بعض من هذه المدونة تم ترميزه على المستوى الصرفي والنحوي (المكنز المرمر) حوالي سبعة ملايين كلمة مرمره صرفيا ونحويا.

مكونات المعجم

يشتمل المعجم العربي الحديث على حوالي ستين ألف مدخل - قابلة للزيادة لمزامنة التطور اللغوي - ، تم تحديدها حسب تردها إحصائياً في كل حرف، وهذه المداخل نوعان:

1. مداخل معجمية مفردة عبارة عن كلمة واحدة.

2. مداخل معجمية مركبة من أكثر من كلمة، وهي نوعان:

1- المتلازمات

2- المسكوكات

- **المتلازمات:** تجمع معجمي من كلمتين أو أكثر جرت العادة على تلازمهما وتكرر حدوثهما، مع احتفاظ كل منهما بدلالته، ويمكن تصنيف المتلازمات بالصورة التالية: الصفة والموصوف، المضاف والمضاف إليه.

- **صفة وموصوف:** بث مباشر، المقالة الافتتاحية، سكين حاد... إلخ

- **مضاف ومضاف إليه:** وكالة إعلان، قسم الشرطة، غلاف الكتاب... إلخ

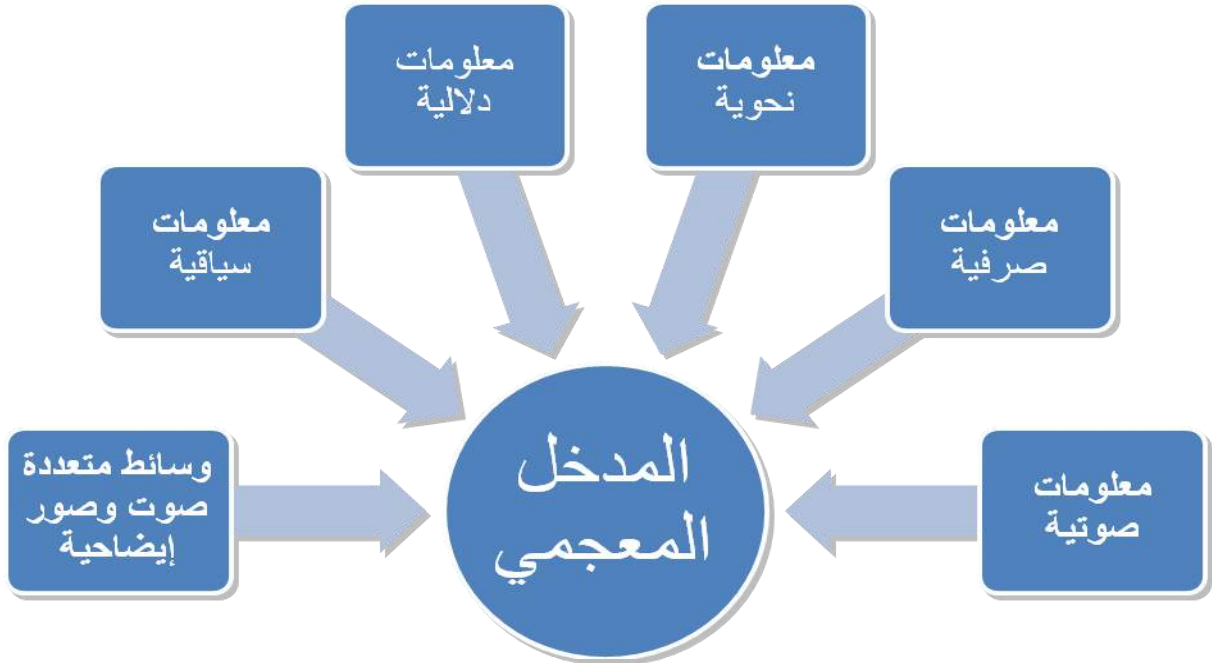
- **المسكوكات:** كلمتان أو أكثر اصطلاح على استخدامهما للدلالة على معنى معين قد يكون على علاقة بدلالة المفردات أو قد يكون له علاقة له بدلالة المفردات، وهي نوعان مسكوكات اسمية أو فعلية.

- مسكوكات اسمية: الذهب الأبيض، ملك الغابة، القفص الذهبي... إلخ

- مسكوكات فعلية: لقي حتفه، رجع بخفي حنين، ضرب أخماسا في أسداس... إلخ

المعلومات اللغوية والمعجم:

إن معرفة مجموع مفردات اللغة أو معجمها، تقتضي الإحاطة بمجموعة من المعلومات عن هذه المفردات، مثل خصائصها الصوتية والصرفية والتركيبية والدلالية ومن هذا المنطلق حرصنا في المعجم أن نضمنه حزما من البيانات اللغوية المتنوعة صرفية ونحوية ودلالية... إلخ تعرّف الكلمة وتمييزها مبنى ومعنى لتلبية الاحتياجات المتنوعة للمستخدم متخصصا وغير متخصص.



المعلومات الصوتية:

يتم نطق الكلمة نطقا واضحا، مع توصيف الكلمة صوتيا حسب الألفبائية الصوتية المعتمدة، مع ذكر مخرج كل حرف وخصائصه الصوتية.

المعلومات الصرفية:

مجموعة الخصائص الصرفية المميزة لمبنى الكلمة وهي كالتالي:

- ❖ الجذر: أصل الكلمة ويكون ثلاثيا أو رباعيا، ويتم كتابة حروفه بصورة متفرقة (ك ت ب، ز ل ز ل).
- ❖ الميزان الصرفي: وضع الموازين الصرفية للكلمات العربية (فَعَلَ، فَاعِلٌ، مَفْعُولٌ، فَعَائِلٌ...إلخ)
- ❖ قسم الكلم، وتم حصر أقسام الكلم المستخدمة في أربعة أقسام رئيسة هي (اسم، فعل، صفة، كلمة وظيفية) ولكل قسم من هذه الأقسام الرئيسية فروع ثانوية تدرج تحتها، فعلى سبيل المثال:
 - اسم: يندرج تحته (اسم ذات، اسم آلة، اسم عدد، اسم علم، اسم مكان، اسم زمان...إلخ).
 - فعل: يندرج تحته (فعل ماض مجرد، فعل ماض مزيد)
 - صفة: يندرج تحتها (اسم الفاعل، اسم المفعول، الصفة المشبهة، صيغة المبالغة...إلخ)
- ❖ الجمود والاشتقاق (فعل متصرف، فعل جامد، اسم مشتق، اسم جامد)
- ❖ الصحة والاعتلال (صحيح، مهموز، مقصور، منقوص، معتل مثال، معتل أجوف، معتل لفيف مفروق...إلخ)
- ❖ مشتقات الفعل (المصدر، اسم الفاعل، اسم المفعول، المصدر الميمي...إلخ)
- ❖ تصريف الفعل (المضارع، الأمر، الماضي المجهول، المضارع المجهول)
- ❖ قابلية الكلمة للسوابق واللواحق.
- ❖ جمع التكسير بأنواعه مثل جمع القلة أو الكثرة، صيغة منتهى الجموع، جمع الجمع.

المعلومات النحوية:

مجموعة الخصائص النحوية المميزة للكلمة وهي كالتالي:

- ❖ الإعراب والبناء.
- ❖ قابلية الكلمة للتوين.
- ❖ اللزوم والتعدي (فعل لازم، فعل متعد بحرف، فعل متعد، فعل متعد لمفعولين)
- ❖ رصد حروف الجر المتعلقة بالكلمة سواء أكانت فعلا أو اسما (تجنى على، تمسك بـ، كتاب عن ، حول، خريطة لـ ، بـ...إلخ)

المعلومات الدلالية:

مجموعة من الخصائص الدلالية المميزة للكلمة وهي كالتالي:

- ❖ النوع (مذكر ، مؤنث، مذكر ومؤنث)
- ❖ العقلانية وغير العقلانية (عقل، غير عقل)
- ❖ حقيقة المعنى ومجازيته (حقيقي، مجازي)
- ❖ المستوى العُمريّ لاستخدام المعنى (مبتدئ، متوسط، متقدم - متخصص-)
- ❖ تأصيل المعنى (دخيل، مُحدَث، مُعَرَّب، مجمعيّ، قرآني)
- ❖ التصنيف الموضوعي لمعنى الكلمة (طبي، سياسي، اقتصادي...إلخ)
- ❖ الترجمة الإنجليزية والفرنسية للمدخل من قاموسي:
"كولينز" للترجمة الإنجليزية، و "لاروس" للترجمة الفرنسية.

المعلومات السياقية:

الخصائص السياقية للكلمة وتتمثل في النقاط الآتية:

- ❖ الكلمات المصاحبة، والتي تشمل رصد لسلوك الكلمة السياقي مع غيرها من المفردات، وذلك من خلال علاقتي الإضافة، والوصف.

مثال: كلمة (أسلوب)

علاقة الإضافة : أسلوب حياة، أسلوب الكاتب، أسلوب المعيشة، أسلوب التعامل...إلخ

علاقة الوصف: أسلوب حاد، أسلوب عنيف، أسلوب قذر، أسلوب متحضر...إلخ

- ❖ ذكر المترادفات والمتضادات الخاصة بكل معنى، واعتمدنا فيها على قاعدة البيانات المعجمية الخاصة

بصخر، بالإضافة إلى بعض المصادر التي تفردت بمعالجة هذا الموضوع مثل:

- المكنز الكبير د/ أحمد مختار عمر.
- نجعة الرائد الشيخ/ إبراهيم اليازجي.
- قاموس الأضداد د/ عائدة الدقرمنجي.
- المكنز العربي المعاصر د/ محمود إسماعيل صيني وآخرين.

الوسائط المتعددة (الصور):

يقول الدكتور علي القاسمي في كتابه "علم اللغة وصناعة المعجم" "نحن نرى أن الشواهد الصورية ذات فائدة بالغة ويجب أن يجري العمل على استعمالها بصورة منتظمة وثابتة، فالشواهد الصورية تساعد القارئ على فهم مضمون المقابل اللفظي، وتعزز ما يقرأ وتعمق فهمه لمعنى المقابل اللفظي"

منهجية ومعايير صياغة المعنى والأمثلة:

المعنى والأمثلة هما العنصران الأساسيان لبناء أي معجم ولهذا سنوليها شيئاً من التفصيل والشرح:

أولاً: المعنى

يمثل المعنى الغرض الأساسي من بناء المعجم، وبه يحكم على مدى إفادة المعجم والقيام بوظيفته الأساسية ألا وهي إزالة إبهام الكلمة، **وقد تم اختيار المعاني حسب أولويات ترددها إحصائياً**، ولقد اعتمدنا في صياغة المعنى عدداً من المعايير:

❖ مراعاة أركان التعريف المنطقي بأن يكون جامعاً مانعاً بذكر صفات الشيء المعرفة له النافية لاشتراك شيء آخر معه.

▪ المنشار: آلة لها نصل حديدي ذو أسنان مدببة يستخدم لقطع الخشب ونحوه.

❖ اعتمدنا في تعريف الكلمات قليلة الشهرة والاستعمال على مرادفها الأكثر شيوعاً وشهرة منها.

▪ الوغى: الحرب.

▪ الهامة: الرأس.

▪ الناطور: الحارس.

❖ اعتمدنا التعريف بالضد في بعض الحالات مثل:

▪ الأبيض: لون من الألوان عكس الأسود.

▪ الليل: عكس النهار.

❖ الشرح بالألفاظ بسيطة واضحة يسهل على المستخدم أيّاً كان مستواه التعليمي أو الثقافي أن يفهمها.

❖ البعد عن الألفاظ الغامضة المبهمة في التعريف مثل كلمات على شاكلة "معروف، شهير... إلخ"

ثانياً: الأمثلة

المثال له أهمية كبيرة في توضيح المعنى.

ويقدم المعجم نوعين من الأمثلة، أمثلة من القرآن الكريم، أو الحديث الشريف أو الشعر العربي، وأمثلة أخرى من الصحف والمجلات ومواقع الأخبار واعتمدنا في رصدها على محرك جهيئة للأخبار - أداة حاسوبية لجمع الأخبار من المصادر الإخبارية-

وقد حرصنا على المعايير الآتية في انتقاء الأمثلة:

- أن يكون المثال بلغة بسيطة سهلة لا غموض فيها.
- لا يزيد عن خمس عشرة كلمة.
- تحديد الكلمة مناط التمثيل.
- الحرص على بيان مصدره.

الربط بين معاني الكلمة قديما وحديثا:

يربط المعجم بين القديم والحديث وكيفية تناول القدماء لهذه المادة المعجمية، وتمثل ذلك في الربط بين المعجم العربي الحديث وثلاثة معاجم قديمة:

- القاموس المحيط للفيروز آبادي.
- تاج العروس للزبيدي.
- لسان العرب لابن منظور.

وختاما بعد هذا العرض الموجز للمعجم العربي الحديث، نستطيع أن نقول إن المعجم الوافي الذي يلبي متطلبات الصناعة المعجمية بالمعنى العلمي الدقيق هو المعجم الذي ينهض على خطة محكمة قابلة للتنفيذ لأنها تنطلق من قواعد ومبادئ تراعي روح العربية وروح العصر، والجهد المطلوب لتحقيق كل ذلك كبير وشاق، وعلينا أن نتابع العمل من حيث انتهى من قبلنا لا أن نبدأ من جديد، ولا نملك أخيراً إلا أن نقول: إن الحلم هو الحقيقة التي ستأتي بتضافر جهودنا و عملنا المستمر .

ملاحق البحث

ملحق (1)

أدوات بناء المعجم العربي الحديث

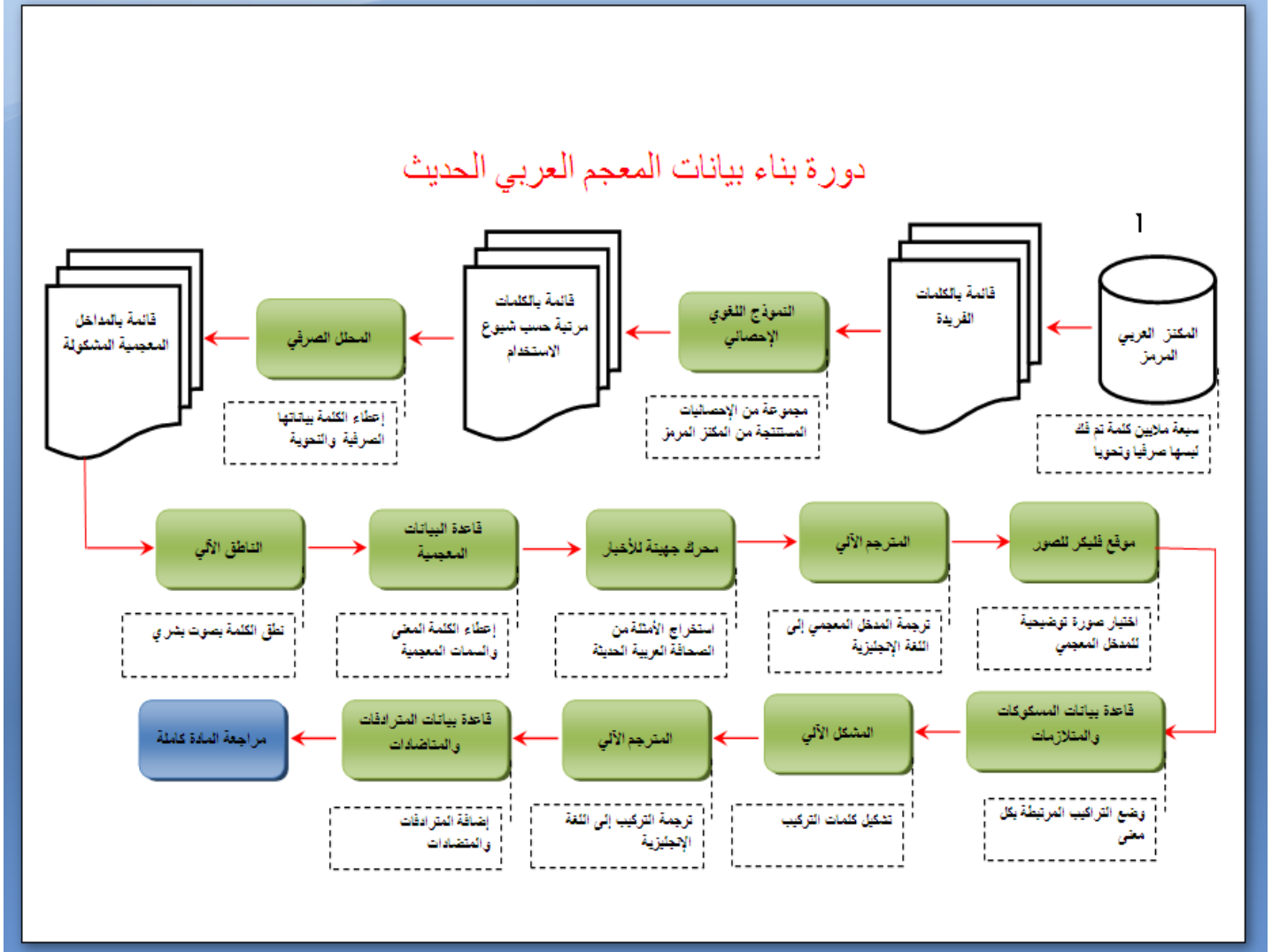


أدوات بناء المعجم العربي الحديث



ملحق (2)

دورة بناء المعجم العربي الحديث



ملحق (3)

صورة من المعجم العربي الحديث

المعجم العربي الحديث

عين بحث

صخر Sakhr Software

ء ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي

عين^١ عين^٢

المعاني

1.4 عضو الإبصار في الإنسان والحيوان **عَيُونُ/ أَعْيُنُ**



Eye Collins
œil Larousse

احتتمت بمستشفى **العيون** بجدة فعاليات الدورة المكثفة في مراجعة طب **العيون**،
[جريدة البلاد - السعودية]

الكلمة في المعاجم التراثية

- لسان العرب
- القاموس المحيط
- تاج العروس

صرفية نحوية دلالية

اسم ذات
- الجذر الثلاثي **ع ي ن**
- ميزان **فعل**

المسكوكات المتلازمات

- **العين** السحرية
- إنسان **العين**

المترادفات المتضادات

الباصرة

توصيات البحث

- ❖ الاهتمام بالمعجم العربي لما يمثله من أهمية عظيمة لمتحدثي العربية ودارسيها.
 - ❖ الاهتمام برصد الألفاظ الجديدة ومتابعتها بصورة دائمة، وإضافتها للمعجم العربي.
 - ❖ عمل ورش عمل من المتخصصين في كل فن من الفنون لعرض الألفاظ الخاصة بهم وتقديم المفهوم والتعريف الخاص بكلمات هذا الفن.
 - ❖ الدعوة إلى الاستفادة من استخدام الوسائل الإلكترونية في الكشف عن معاني الكلمات الجديدة والقديمة على السواء لسهولة البحث من خلالها.
- "وآخر دعوانا أن الحمد لله رب العالمين"

المصادر والمراجع

- بحوث الاجتماع الثاني لخبراء المعجم الحاسوبي التفاعلي للغة العربية.
- الجانب اللغوي للمعجم الحاسوبي للغة العربية، د / محمود بن إسماعيل صالح
- علم اللغة وصناعة المعجم الدكتور علي القاسمي.
- المعجم بين الواقع والطموح د/ نادية حسكور، مجلة مجمع اللغة العربية بدمشق، المجلد (78) الجزء الثالث.

آفاق اللغويات للسانيات الحاسوبية : مصادرها النظرية وغاياتها العملية

د.نبيل على

nabilalii@gmail.com

باتت اللغة تحتل موقع القلب على الخريطة الجيومعرفية ، وقد أقامت اللغة علاقات تبادلية مع معظم فروع المعرفة ، كان من آخرها علوم الحاسوب وتكنولوجيا المعلومات ليظهر إلى الوجود ثنائى التفاعل المعرفى المتمثل فى اللغويات الحاسوبية .

وشتان بينهما ،فى حين يسود الطابع النظرى اللغويات الحاسوبية تنطلق من اللغه اساسا وتتخذ من الحاسوب ونظم المعلومات أداة لعمليات التحليل والتوليد اللغوية ، وفى المقابل يسود الطابع التطبيقى الحاسوبيات اللغوية التى تلجأ لمعرفة اللغويات الإحصائية والرياضية والمنطقية من أجل تأصيل الأسس التى تبنى عليها نظم حوسبة اللغة ونظم المعلومات من قبيل نظم البحث فى النصوص وتصميم نظم الإعراب الآلى وتحويل النصوص إلى مقابلة المنطوقه .

وغنى عن القول أن التقدم فى الحاسوبيات اللغوية رهن بما يتم إنجازه على الصعيد اللغوى /الحاسوبى ،وهو الأمر الذى يتطلب مداومه اكتشاف آفاق جديدة وتحديد مصادرها النظرية وغاياتها التطبيقية

تشمل قائمة المصادر :

- فلسفة المعرفة (الابستمولوجيا)
- علم النفس المعرفى
- علم النفس الثقافى
- علم الاجتماع المعرفى
- علم الاجتماع الثقافى
- التاريخ الرقمى
- نظرية الأدب والنقد
- علم الخطاب
- علم المخ

– علم اللغويات البيولوجيه

تسعى الدراسة إلى تحديد شبكة العلاقات التي تربط بين اللغة وفروع المعرفة عموماً
وشق المعنى على وجه الخصوص

يتم تناول الأغراض العملية للأفاق الاستكشافية المذكورة من ثلاث جهات نظر هي :

– توسيع وتعميق التطبيقات العملية لحوسبة اللغة ونظم المعلومات .

– إقتراح قائمة بتطبيقات مستحدثه .

– إقتراح خطة موضوعية لتأصيل اللغويين الحاسوبيين.

A Corpus-based Approach for the Automatic Development of UNL Grammars

Sameh Alansary

*Department of Phonetics and Linguistics, Faculty of Arts, Alexandria University
El Shatby, Alexandria, Egypt*

Bibliotheca Alexandrina, Alexandria, Egypt
Sameh.alansary@bibalex.org

Abstract— This paper aims to present the process of the automatic extraction of the tree-to-tree (TT) rules which constitute the most intricate phase in developing the UNL analysis and generation grammars. During analysis, tree-to-tree rules are used to reveal the deep syntactic structure out of the surface syntactic structure; while in generation, they are used to transform the deep syntactic structure into a surface structure. In this paper, we present a method to extract automatically TT rules using an annotated corpus. The availability of large syntactically annotated corpora such as the Penn Tree Bank presents us with the opportunity to automatically build broad coverage grammars. Experiments have proved that grammars that are dependent on the context can be more effective than context-free grammars. We have automatically constructed the TT rules using the Penn Tree Bank version 2.

1 INTRODUCTION

In the beginning of 2009 a new system came to light in language processing called the UNL+3; UNL+3 is the latest development to the first generation of UNL (more information about the earlier system can be found at <http://www.undl.org/unlsys/unl/unl2005/>). UNL+3 offers improvements to the existing infrastructure, however, it changes some of the core fundamentals of UNL. The change encompasses the linguistic components (Universal Words, Relations, Attributes and Features) as well as the non-linguistic ones (tools, engines and applications). Also, the structure of the linguistic resources (grammars, dictionaries, corpora.etc.) that handle these components has been drastically changed (more information about UNL+3 can be found <http://www.unlweb.net/wiki/>).

Natural language sentences and UNL graphs are supposed to convey the same amount of information in different structures: whereas the former arranges data as an ordered list of words, the latter organizes it as a hyper-graph. In that sense, translating from natural language into UNL and from UNL into natural language is ultimately a matter of transforming lists into networks and vice-versa. The UNDL Foundation's generation and analysis tools; Eugene and IAN, assume that such transformation should be carried out progressively, i.e., through a transitional data structure: the tree, which can be used as an interface between lists and networks. Accordingly, the UNL Grammar utilizes seven different types of rules: LL, TT, NN, LT, TL, TN and NT.

In analysis (NL-UNL), list-to-list or List Processing (LL), list-to-tree or surface-structure Formation (LT), tree-to-tree or syntactic processing (TT), tree-to-network or deep-structure formation (TN) and network-to-network or semantic processing (NN) rules are used. The Natural language original sentence (NL) is supposed to be preprocessed by the LL rules in order to become an ordered list. Next, the resulting list structure is parsed with the LT rules so as to unveil its surface syntactic structure which is syntactic tree. The tree structure is further processed by the TT rules in order to expose its inner organization; the deep syntactic structure which is more suitable to semantic interpretation. Then, this deep syntactic structure is projected into a semantic network by TN rules. The resultant semantic network is then post-edited by the NN rules in order to comply with UNL standards and finally generate the UNL Graph.

In generation, the five types of rules used are the network-to-network or semantic processing (NN), network-to-tree or deep-structure Formation (network-to-tree), tree-to-tree or syntactic processing (TT), tree-to-list or surface-structure formation (TL) and list-to-list or list processing (LL) rules. The UNL graph is preprocessed by the NN rules in order to become a more easily tractable semantic network. The resulting network structure is converted by the NT rules into a syntactic structure that is still distant from the surface structure, since it is directly derived from the semantic arrangement. This deep syntactic structure is subsequently transformed into a surface syntactic structure by the TT rules. The surface syntactic structure undergoes many other changes according to the TL rules, which generates a NL-like list structure. This list structure is finally realized as a natural language sentence by the LL rules.

The tree-to-tree rules (TT) phase is a common phase in both the analysis and generation grammars; they are used for processing trees, both in analysis and in generation. During analysis, these rules are used to reveal the deep structure out of the surface structure; however, in generation, they are used to transform the deep syntactic structure into a surface structure.

The TT is the most challenging phase in the UNL grammar, hence, it was necessary to think about a way to automate it. For this purpose, empirical rather than introspective data is required in order to express the authenticity of the language; a corpus-based approach is selected to be the base of the automatic extraction of the tree-to-tree module.

Corpus-based studies provide a means for handling large amounts of language and keeping track of the contextual factors. The availability of resources such as the Treebank¹ has a great impact on the investigations that depend on this kind of studies, and has paved the way for the automatic updating of grammar which first emerged in the nineties. Moreover, the availability of the Treebank caused us to wonder whether it is possible to extract grammar rules from surface structures using the Treebank.

This paper aims to build an application that extracts the grammatical rules needed to build the component responsible for transferring the surface syntactic structure into deep structure and vice versa; the so-called TT module referred to earlier. The examples used in this paper are derived from the English Penn tree-bank, however, this does not imply that the process or the application are only applicable to English; English was mainly chosen for the purpose of illustration and clarity. Section 2 discusses the design and implementation of application and the design of the data selected for the application. Section 3 discusses how the output of this application fits within the other modules in the UNLdev environment discusses the challenges. Section 4 evaluates the results of the generated and analyzed sentences. And finally section 5 is a conclusion and a survey of the future work.

2 AUTOMATIC UNL GRAMMAR EXTRACTOR

An application was built to automatically extract TT rules for both the analysis and generation processes. Its main objective is to allow UNL grammar developers to establish a more empirical and authentic grammar by means of massive analyzed data that cover the main syntactic structures of a language.

2.1 CORPUS COMPILATION

The Treebank or the parsed corpus is a text annotated with syntactic structure commonly represented as a tree structure. Treebanks can be annotated manually or semi-automatically. Examples of the available treebanks are the BulTreeBank, the Penn Treebank and the Quranic Arabic Dependency Treebank.....etc.

The corpus used in the development of our application is derived from the English Penn Treebank. The English Penn Treebank annotates phrase structures²; for example, the syntactic analysis for 'peter killed Mary' following the Penn Treebank notation, is shown in figure 1 and may be represented by labeled brackets as shown in figure 2.

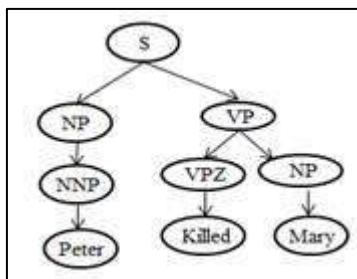


Figure (1) Penn treebank analysis for " peter loves Mary "

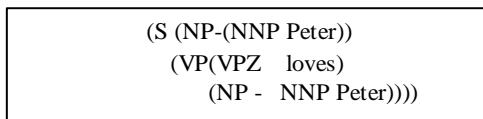


Figure (2): Penn treebank analysis for " peter loves Mary "

¹ A bank of texts that are annotated syntactically and are commonly represented as tree structures; hence the name Treebank.
² more information about the Penn Treebank can be found at <http://www.cis.upenn.edu/~treebank/>

120 structures have been chosen from the Penn Treebank to constitute the data that will be used in the development of our application. These structures are divided into 113 full sentences and 10 phrases. The phrases include noun phrases (NPs), verb phrases (VPs) and adverbial phrase (APs). The following are five types of noun phrases:

1. Proper nouns
2. Pronouns,
3. Det + noun,
4. Det + adj + noun
5. Det + adj + adj + noun.

VPs include verbs in the past tense, Verbs in present tense and auxiliaries + verb. While APs include adverbs only. The structures of the 113 sentences are mainly generated from the rules (1),(2),(3) and (4):

- (1) S -> NP + VP
- (2) NP -> DET + (Adj) + Noun
- (3) VP -> V + (NP) + (AP)
- (4) AP -> Adverb

Out of these rules, it is possible to generate 113 sentences since the structures between brackets are optional. It is possible to generate from these rules a numerous amount of structures such as in table 1:

Prpoernoun + aux + verb	Det + adj + noun + aux + verb +Pronoun + ADV
Pronoun + aux + verb	Det + adj + noun + aux + verb + Det + noun + ADV
Det + noun + aux + verb	Det + adj + noun + aux + verb +Det + adj + noun + ADV
Det + adj + noun + aux + verb	Det + adj + noun + aux + verb Det + adj + adj + noun + ADV
Det + adj + adj + noun +aux + verb	Det + adj + adj + noun + aux + verb +Pronoun
Prpoernoun + aux + verb + adv	Det + adj + adj + noun + aux + verb + Det + noun
Pronoun + aux + verb + adv	Det + adj + adj + noun + aux + verb +Det + adj + noun
Det + noun + aux + verb + adv	Det + adj + adj + noun + aux + verb Det + adj + adj + noun
Det + adj + noun + aux + verb + adv	Det + adj + adj + noun + aux + verb +Pronoun + ADV
Det + adj + adj + noun +aux + verb + adv	Det + adj + adj + noun + aux + verb + Det + noun + ADV
Prpoernoun + aux + verb +Pronoun	Det + adj + adj + noun + aux + verb +Det + adj + noun + ADV
Prpoernoun + aux + verb + Det + noun	Det + adj + adj + noun + aux + verb Det + adj + adj + noun + ADV
Prpoernoun + aux + verb +Det + adj + noun	Prpoernoun + verb
Prpoernoun + aux + verb Det + adj + adj + noun	Pronoun + verb
Prpoernoun + aux + verb +Pronoun + ADV	Det + noun + verb
Prpoernoun + aux + verb + Det + noun + ADV	Det + adj + noun + verb
Prpoernoun + aux + verb +Det + adj + noun + ADV	Det + adj + adj + noun + verb
Prpoernoun + aux + verb Det + adj + adj + noun + ADV	Prpoernoun + verb
Pronoun + aux + verb +Pronoun	Pronoun + verb + adverb
Pronoun + aux + verb + Det + noun	Det + noun + verb +adverb
Pronoun + aux + verb +Det + adj + noun	Det + adj + noun + verb +adverb
Pronoun + aux + verb Det + adj + adj + noun	Det + adj + adj + noun + verb +adverb
Pronoun + aux + verb +Pronoun + ADV	Prpoernoun + aux + verb +adverb
Pronoun + aux + verb + Det + noun + ADV	Pronoun + aux + verb + adverb
Pronoun + aux + verb +Det + adj + noun + ADV	Det + noun + aux + verb +adverb
Pronoun + aux + verb Det + adj + adj + noun + ADV	Prpoernoun + verb + Prpoernoun
Det + noun + aux + verb +Pronoun	Prpoernoun + verb + Pronoun
Det + noun + aux + verb + Det + noun	Prpoernoun + verb + Det + noun
Det + noun + aux + verb +Det + adj + noun	Prpoernoun + verb + Det + adj + noun
Det + noun + aux + verb Det + adj + adj + noun	Prpoernoun + verb + Det + adj + adj + noun
Det + noun + aux + verb +Pronoun + ADV	Prpoernoun + verb + Prpoernoun + adv
Det + noun + aux + verb + Det + noun + ADV	Prpoernoun + verb + Pronoun+ adv
Det + noun + aux + verb +Det + adj + noun + ADV	Prpoernoun + verb + Det + noun+ adv
Det + noun + aux + verb Det + adj + adj + noun + ADV	Prpoernoun + verb + Det + adj + noun + adv
Det + adj + noun + aux + verb +Pronoun	Prpoernoun + verb + Det + adj + adj + noun + adv
Det + adj + noun + aux + verb + Det + noun	Pronoun + verb + Prpoernoun
Det + adj + noun + aux + verb +Det + adj + noun	Pronoun + verb + Pronoun
Det + adj + noun + aux + verb Det + adj + adj + noun	Pronoun + verb + Det + noun

Det + adj + noun + verb + Det + adj + adj + noun + adv	Pronoun + verb + Det + adj + noun
Det + adj + adj + noun + verb + Prpoernoun	Pronoun + verb + Det + adj + adj + noun
Det + adj + adj + noun + verb + Pronoun	Pronoun + verb + Prpoernoun + adv
Det + adj + adj + noun + verb + Det + noun	Pronoun + verb + Pronoun+ adv
Det + adj + adj + noun + verb + Det + adj + noun	Pronoun + verb + Det + noun+ adv
Det + adj + adj + noun + verb + Det + adj + adj + noun	Pronoun + verb + Det + adj + noun + adv
Det + adj + adj + noun + verb + Prpoernoun + adv	Pronoun + verb + Det + adj + adj + noun + adv
Det + adj + adj + noun + verb + Pronoun+ adv	Det + noun + verb + Prpoernoun
Det + adj + adj + noun + verb + Det + noun+ adv	Det + noun + verb + Pronoun
Det + adj + adj + noun + verb + Det + adj + noun + adv	Det + noun + verb + Det + noun
Det + adj + adj + noun + verb + Det + adj + adj + noun + adv	Det + noun + verb + Det + adj + noun
Det + adj + noun + verb + Prpoernoun	Det + noun + verb + Det + adj + adj + noun
Det + adj + noun + verb + Pronoun	Det + noun + verb + Prpoernoun + adv
Det + adj + noun + verb + Det + noun	Det + noun + verb + Pronoun+ adv
Det + adj + noun + verb + Det + adj + noun	Det + noun + verb + Det + noun+ adv
Det + adj + noun + verb + Det + adj + adj + noun	Det + noun + verb + Det + adj + noun + adv
Det + adj + noun + verb + Prpoernoun + adv	Det + noun + verb + Det + adj + adj + noun + adv
Det + adj + noun + verb + Pronoun+ adv	
Det + adj + noun + verb + Det + noun+ adv	
Det + adj + noun + verb + Det + adj + noun + adv	

Table (1): The structures can be generated from rules (1),(2),(3) and (4)

2.2 DEEP-STRUCTURE RULE EXTRACTOR

The main objective of this application is to allow UNL grammar developers to establish a more empirical and authentic grammar. This can be achieved by means of massive analyzed data that cover the syntactic structures of a language. The application is designed to be flexible and interactive for the grammar developer. Firstly, the user should determine the kind of structure to be handled; whether a full sentence or a certain phrase. The user has to select whether the required rules are for the generation process or the analysis process, since TT rules are not the same in both processes.

After selecting the structure to be dealt with, the application reaches the processing phase which is composed of two levels; the phrase level and the sentence level; according to the structure of the input, the engine will decide which level is suitable. For each level there is a set of conditions and decisions that will be discussed later in sub section 2.3. Figure (3) is a simple diagram that illustrates how the input is processed in both levels.

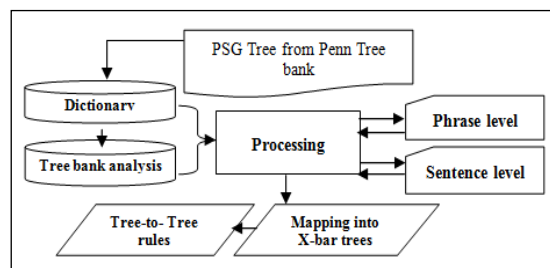


Figure (3): a simple diagram for algorithm of the application

After the grammar developer selects the structure to be processed, the next step is for the engine to determine the corresponding phrase structure; in other words, the application would search its memory for the PSG structure most appropriate to this sentence according to Penn Treebank analysis. This task would be done with the help of a dictionary that contains words and their part of speech. The processing phase itself would be accomplished according to the type of the structure; whether the input is a phrase or a full sentence. As mentioned before, the main role of tree-to-tree rules is transforming the deep syntactic structure into a surface syntactic structure according to the principles of X-bar theory as the adopted theory in the UNL framework. Thus, there must be an intermediate phase before extracting TT rules from the annotated sentence since the annotated sentences are parsed according to Phrase Structure Grammar Structure PSG grammar. This intermediate process will map the annotated input of the Penn Treebank from PSG analysis to X-bar. This mapping is a challenging task since it is not one-to-one mapping process. The challenges of this phase will be discussed latter in section 4.

Our application depends mainly on a set of situations and decisions; in other words, it makes a specific decision to convert a branch of the PSG analysis into X-bar analysis according to the set of conditions present. As mentioned before, mapping from PSG to X-bar is not one-to-one; you cannot simply convert the branch named NP in PSG into a branch named NB in X-bar.

To demonstrate this, we are going to take the sentence "she kicked the ball strongly" as an example. The previous flowchart in figure (3) will explain the steps followed in mapping process. Firstly, the PSG input for this sentence would be as shown in figure (4):

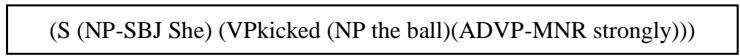


Figure (4): The Penn treebank analysis for "she kicked the ball strongly"

Now we have four situations and four decisions accordingly as illustrated in figures (5), (6), (7), (8). The first decision has to do with the branch named "(S (NP... (VP... ", as shown in figure (4). If a sentence consists of a noun phrase and a verb phrase, then, it begins with (S(NP... (VP... in PSG analysis which should be converted initially to "?P(NP" and "VP" as shown in figure (4) for decision 1. The type of the phrase "?P" will not be defined here since it will be defined later when more information is available.

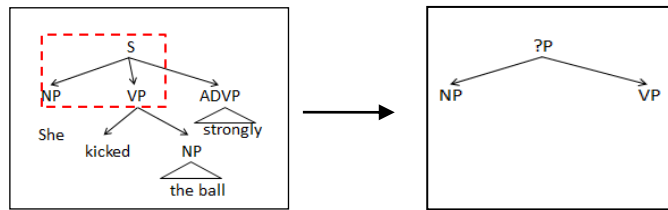


Figure (5): situation 1 and decision 1

The second situation faced here is the existence of the verb phrase and adverb phrase in "(VP ... (ADVP...". As shown in PSG analysis in figure (6), the VP and the ADVP are sisters from the same mother node S. However, in order to convert it to X-bar, the decision was to make the ADVP an adjunct that would be a sister node to the V bar not the VP.

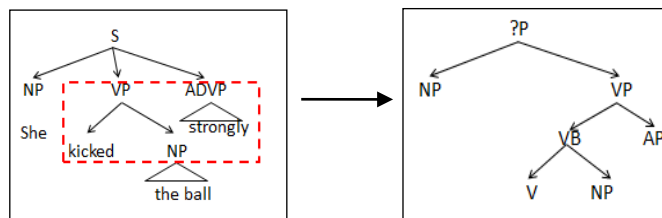


Figure (6): situation 2 and decision 2

The third situation has to do with the inflectional morpheme attached to the verb "kick". According to the dictionary, the verb "kick" is a regular verb in the past tense. This would define the "?P" as being inflectional phrase "IP" as shown in figure number (7).

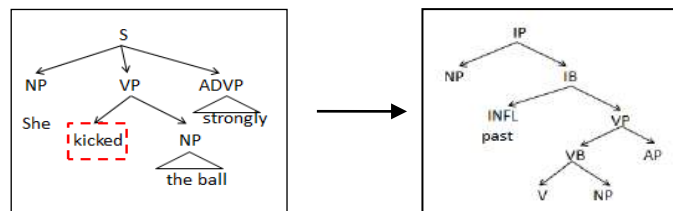


Figure (7) situation 3 and decision 3

Finally, the last decision will be devoted to placing the terminal nodes "she – kick - the - ball - strongly" in their slots. See figure number (8).

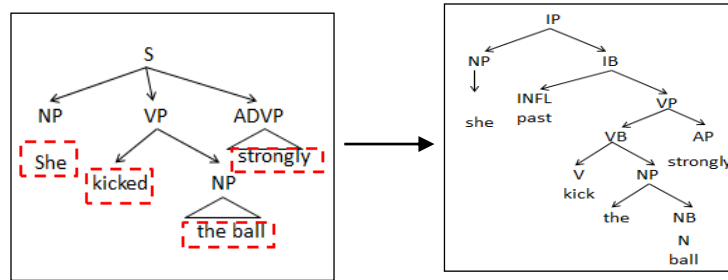


Figure (8) situation 4 and decision 4

Hence, the final X-bar tree corresponding to the PSG tree “(S (NP-SBJ She) (VPkicked (NP the ball)(ADVP-MNR strongly)))” would be represented as: "IP (NP-she IB(INFL-past VP(VB(V-kick ;NP(DET the NB(N-ball)))AP-strongly))", and from this representation the TT rules would be extracted for both analysis or generation.

2.3 REMARKS ON MAPPING FROM PHRASE STRUCTURE GRAMMAR TO X-BAR

The X-bar theory is discussed in almost all modern textbooks of syntax, and it is assumed as a theory of Phrase Structure Grammar. PSG does not have intermediate categories which are larger than a word but smaller than a phrase. Such a category is needed because there are such units that could not be classified neither as a phrase nor as a category. But Phrase Structural Grammar treats such unit in the same manner it treats a category³. X-bar theory was developed in order to provide this intermediate category. The X-bar abstract schema is illustrated in figure (9):

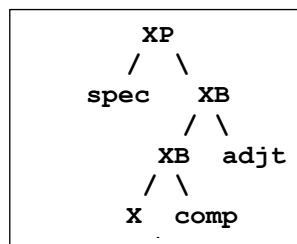


Figure (9) The schema of X bar theory

In figure (9), X is the head of the whole syntactic structure; it is derived (or projected) out of it. The letter X is used to as an arbitrary lexical category. When analyzing a specific utterance, specific categories are assigned, and the X may become an N for noun, a V for verb, an J for adjective, or a P for preposition. The “comp” stands for complement and it is an internal argument which is necessary to the head to complete its meaning. The “adjt” stands for adjunct; it is a word, phrase or clause which modifies the head but which is not syntactically required by it. Finally, the “spec” stands for specifier and it is an external argument which qualifies (determines) the head. XB (X-bar) is the general name for any of the intermediate projections derived from X. XP (X-bar-bar, X-double-bar, X-phrase) is the maximal projection of X.

As mentioned before, X-bar theory is assumed to be a theory of PSG, however, mapping from PSG to X-bar analysis is not a direct endeavor. In contrast with X-bar syntax, Phrase Structure rules cannot distinguish between elements that are subcategorized by the head and those that are optional. For example, in the sentence “A student of physics with a long hair”, it is not possible to determine if the prepositional phrase “of physics” is a complement of the head "student" or an adjunct that is optional. However, in X-bar theory, the complement is a sister of the head, and the constituent formed by a head and its complement are labeled as units named XB (X-bar). As shown in figure (10), in PSG analysis, the two prepositional phrases both originate from the same node NP; on the other hand, in figure (11), according to X-bar analysis, the prepositional phrase "of physics" is a sister node of the head and is its complement, and the prepositional phrase "with long hair" originates from the node NP since it is an optional adjunct.

³<http://epistemic-forms.com/ps-xbar.html>

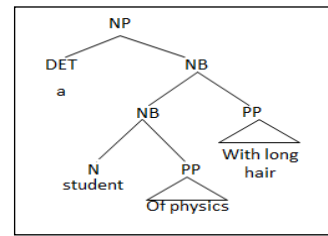
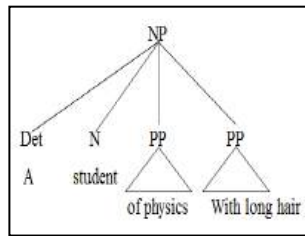


Figure (10) representation of the structure using PSG Figure (11) representation of the structure using X-bar

Another note on PSG analysis is regarding the specifiers. Specifiers in PSG are still part of the constituent XP. X-bar specifiers are also XPs; however, the specifier is daughter to the XP and sister to the XB. Referring back to figures (10) and figure (11), the specifier "a" in PSG analysis is a daughter of the NP but is a sister to N, PP and PP. In X-bar analysis, the specifier is a daughter of NP and a sister to NB.

The final note regarding X-bar theory is the adjuncts. Adjuncts can be sisters to XB and daughters to XB. PSG lacks the concept of distinguishing adjuncts. As shown in figures 4 and 5, the prepositional phrase "with long hair" is well defined in X-bar analysis since it is the sister of the NB and the daughter of another NB. In contrast to PSG, the preposition phrase originates directly from the NP.

All the previous observations make the task of mapping a whole analysis from PSG to X-bar quite challenging. The grammar developer should be fully aware of the concepts of specifiers, complements and adjuncts to the extent that he/she is able to determine whether a certain unit is a complement or an adjunct even if two units are from the same category "PP" and originate from the same mother "NP".

3 TESTING THE APPLICATION

The main goal of this application is to provide a set of automatically generated rules that would constitute the Tree-to-Tree module which is the most important module in the UNL grammar. This experiment was conducted on English as the source language. The sample of sentences was selected from the English Penn Tree Bank version II⁴ as mentioned in section 2. The collected sample was chosen to represent examples of phrases and complete sentences. In order to test the validity of this application, the generated rules (both for generation and analysis process) were supposed to be input into the UNLdev⁵. We must indicate here that special care have been taken to make sure the automatically extracted rules are in the same format as those already within the UNLdev to ensure homogeneity. The UNLdev is a web-based integrated development environment for creating and editing dictionary entries and grammar rules in order to be used in natural language processing applications. The generation rules were added to EUGENE; the engine responsible for producing natural language texts out of UNL documents, while analysis rules were added to IAN; the engine responsible for producing that produces UNL documents out of natural language texts. The generated rules are imported into the UNLdev environment and saved in a separate text file in UTF8 encoding. The user can then upload the imported rules into EUGENE if they are generation rules, or IAN if they were analysis rules.

We will now show the testing of a sample of the generation TT rules that were extracted automatically and uploaded into EUGENE. Figure (12) shows a screen shoot of EUGENE where a group of rules named "Testing Generation" was created as shown in figure (12). Within this group there are five rule-sets; NN, NT, TT, TL and LL. All the rules-set are ready except the TT rule-set, it is still empty so that the grammar developer can add the automatically extracted TT rules to it. Figure (13) shows the same for IAN, where TT rule-set is still empty so that the analysis TT rules would be added to it.

⁴Bracketing Guidelines for Treebank II Style, Penn Treebank Project.

⁵ You can reach it at <http://www.unlweb.net/wiki/UNLdev>

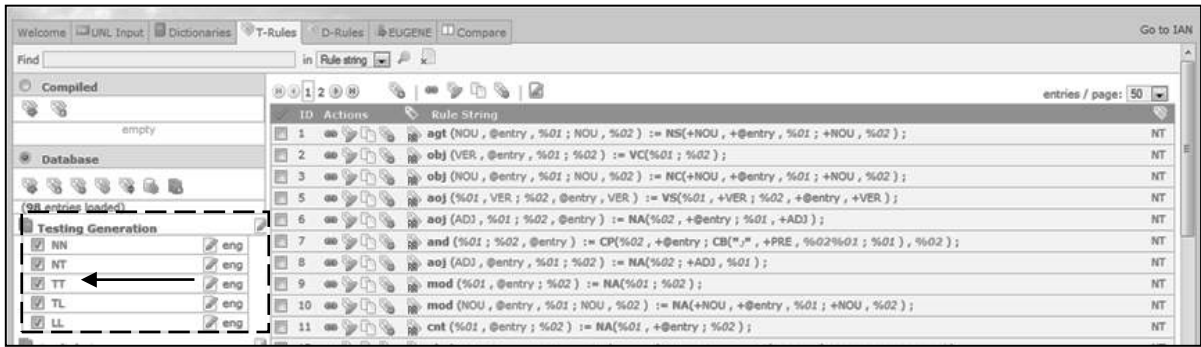


Figure (12) a sample of the grammar in the Eugene engine

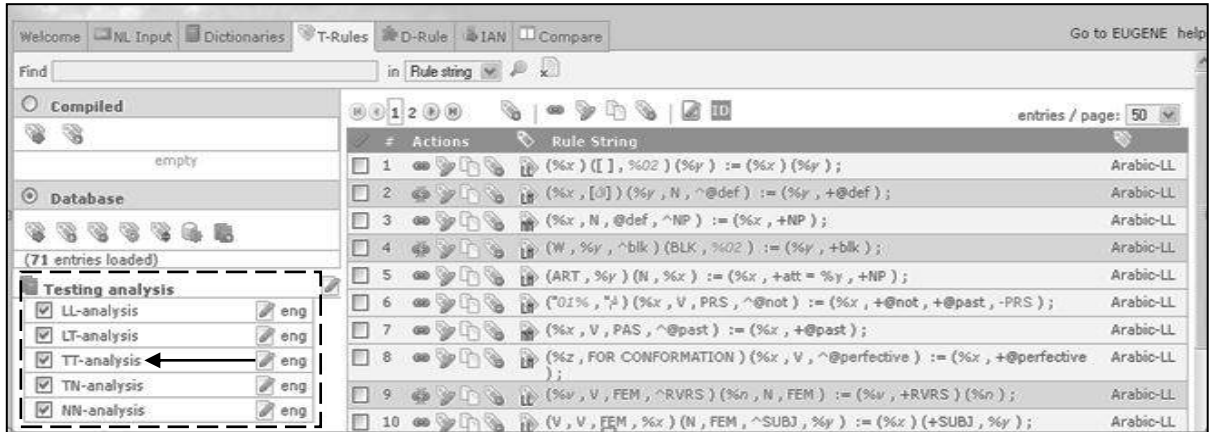


Figure (13): a sample of the grammar in the IAN engine

After preparing these rules-sets in addition to the proper dictionary and the input file to be processed, it is time to test the validity of the automatically generated TT rules. Figure (14) shows the UNL graph that expressing the sentence: "He killed her with a knife in the kitchen yesterday" as an example. In EUGENE, the task is to convert the UNL graph into a natural language string while IAN is supposed to generate the UNL graph for the sentence.

```
[S:8]
{org}
he killed her with a knife in the kitchen yesterday.
{/org}
{unl}
obj(201323958:03.@past, 00:05.@3.@female)
ins(201323958:03.@past, 103623556:0B.@indef)
plc(201323958:03.@past, 103619890:0H.@def)
tim(201323958:03.@past, 115156187:0J)
agt(201323958:03.@past, 00:01.@3.@male)
{/unl}
[/S]
```

Figure (14): The UNL graph of the sentence "He killed her with a knife in the kitchen yesterday"

The NN and NT rules convert the semantic graph in figure (14) into five syntactic branches that are represented in the graph in figure (15):

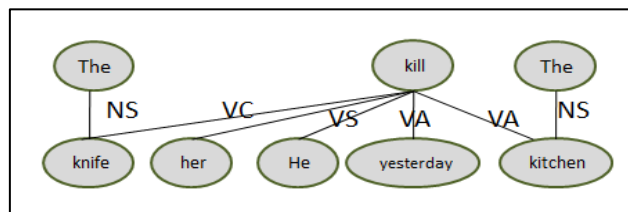


Figure (15) a diagram for the UNL graph

The next step is to add the TT rules that would organize these branches into the surface syntactic structure of the sentence according to the X-bar principles. Finally, the TL and LL rules would list this tree into a natural language sentence. For the analysis process, rules will be added to IAN and the steps will be reversed of processes are reversed in IAN. In the following section, we will test the output of the automatically extracted TT module when being combined with four other manually built modules and using the whole grammar (composed of five modules altogether) to process a number of sentences and save its output to be evaluated later by comparing to the output of another grammar in which all five modules were built manually.

4 EVALUATION

In the UNL System, the F-measure (or F1-score) is the measure of a grammar's accuracy. F-measure is measured by means of considering both the precision and the recall of the grammar to compute the score, according to the formula in figure (16):

$$\mathbf{F\ measure = 2 \times ((precision \times recall) / (precision + recall))}$$

Figure (16) the formula that calculate the F-measure

Precision is the number of correct results divided by the number of all returned results while recall is the number of correct results divided by the number of results that should have been returned. The f-measure can be calculated automatically in the UNLarium⁶. It is possible to compute the F-measure both for analysis where the output is a UNL graph, and in generation where the output is a list of natural language words as shown in figure (17).



Figure (17) A screenshot for F-measure

The result is considered correct when the Levensthein distance between the actual result to be expected and the expected result is less than 30% of the length of the expected result. The Levensthein distance is defined as the minimal number of characters you have to replace, insert or delete to transform a string (the actual output) into another one (the expected output).

For this process, two documents are required. The documents must be in plain text format (.txt) with UTF-8 encoding. One document is for the actual result extracted automatically by our application while the other is for the result obtained from the manually built grammar in the UNLdev. The actual and expected results must be UNL documents in case of analysis, or documents in the same natural language in case of generation and both must not have been post-edited. The table in figure (18) illustrates the F-measure resulting from comparing the output generated from the automatically extracted grammar with that of the manually built grammar.

⁶<http://www.unlweb.net/unlarium/>

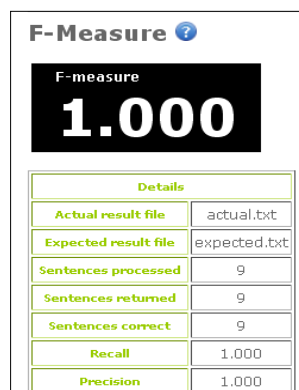


Figure (17): the F-measure comparison result

5 CONCLUSION AND FUTURE WORK

Tree-to-Tree rules are the most complex in UNL grammar and represent the core of the transformation rules in UNL either in generation or analysis. This paper presents a corpus-based application which automatically extracts tree-to-tree rules from syntactically annotated sentences. The process of extraction goes through an intermediate state where the annotated sentences are converted from PSG analysis into X-bar. After the conversion into X-bar, it is possible to use the extracted rules in IAN or EUGENE; the analysis and generation engines in the UNLdev. In the future, the extracted rules can be used in the automatic updating of grammars.

REFERENCES

- [1] András Kornai, Stanford University. The x-bar theory of phrase structure. Hungarian Academy of Sciences, Geoffrey K. Pullum. University of California, Santa Cruz.
- [2] Michelle Sheehan. From Phrase Structure Rules to X-Bar Theory. 18th February 2010
- [3] Joybrato Mukherjee, "Corpus linguistics and English reference grammars", Justus Liebig University, Giessen
Philippe Blache, Marie-Laure Guénot & Tristan van Rullen. A corpus-based technique for grammar development. LPL-CNRS, Université de Provence.
- [4] N. Chomsky. "Remarks on Nominalization" In: Readings in English Transformational Grammar. R. Jacobs and P. Rosenbaum (eds.), pp. 184-221. 1970.
- [5] Noha Adly, Sameh Alansary, "Evaluation of Arabic Machine Translation System based on the Universal Networking Language", in proceedings of 14th International Conference on Applications of Natural Language to Information Systems, (NLDB 2009), Saarland University, Saarbrücken-Germany, June 24 - 26 2009.
- [6] R. Martins, and V. Avetisyan, "Generative and Enumerative Lexicons in the UNL Framework", in proceedings of 7th International Conference on Computer Science and Information Technologies, (CSIT 2009), 28 September - 2 October, 2009, Yerevan, Armenia. 2009.
- [7] http://www.unlweb.net/wiki/Grammar_Specs
- [8] <http://www.unlweb.net/wiki/Relations>
- [9] <http://epistemic-forms.com/ps-xbar.html>
- [10] <http://www.unlweb.net/wiki/UNLdev>
- [11] <http://www.cis.upenn.edu/~treebank/>

Improved Tokenization and POS Tagging for Arabic Text

Michael N. Nawar^{*1}, Mahmoud N. Mahmoud^{*2}, Magda B. Fayek^{*3}

** Computer Engineering Department, Faculty of Engineering, Cairo University
Gamaet El Qahera St. , Giza 12613, Egypt*

[1michaelnawar@ieee.org](mailto:michaelnawar@ieee.org)

[2mah.nabil@ieee.org](mailto:mah.nabil@ieee.org)

[3magdafayek@ieee.org](mailto:magdafayek@ieee.org)

Abstract— In this paper a new technique of stemming and POS tagging for Arabic text is presented. The introduced technique results in significant accuracy improvement compared to previously-known variants of stemming and POS tagging. The proposed method is compared with two other major stemming and POS tagging methods using the same data set. Applying standard evaluation metrics, the proposed stemmer achieves an $F_{\beta=1}$ score of 99.99, and the POS tagger achieves an accuracy of 98.05%. Compared to the highest reported values to our knowledge, the SVM stemmer [1] and the majority POS tagger [2] an error reduction of more than 90% is achieved in the tokenization and 18% and 42% in the POS stage, respectively.

1 INTRODUCTION

Most of Natural language processing (NLP) systems such as information retrieval, text to speech, automatic translation and other use a part-of speech tagger for preprocessing. Supervised methods for part-of-speech (POS) tagging are expensive and time consuming as they depend on manually annotated data. However these methods achieve high results in NLP fields compared to unsupervised methods. Many of the Arabic words are ambiguous in their nature as tag of word can map to a noun, verb or adjective. It is believed that using a statistical approach which makes use of the morphological feature of the Arabic word would result in accurate, efficient and robust tagger that can be used in practical systems. Since both parsing and tagging Arabic words requires a stemming phase, a high accuracy in stemming phase implies a less accumulated error in further phases.

The basic idea of the proposed method is to recognize Arabic tokens and tagging them statistically using the “Conditional Random Field” learning approach by constructing a relevant model and feeding this model with some extra features extracted from the morphological analysis of each Arabic word. This concept is applied in the tokenization, normalization and POS tagging phase.

2 ARABIC NLP AND DATA

There are three main categories of Arabic language; classical – the language of Qur’an, modern standard (MSA) – which is a simplified form of classical that is extracted from news and written documents, and dialectical Arabic which differs from one country to another. One variation of it is the colloquial language which is the daily used language by Egyptians.

In general Arabic has a very rich morphological language where each word can include number, gender, aspect, case, mood, voice, mood, person, and state. The Arabic basic word form can be attached to a set of clitics representing object pronouns, possessive pronouns, particles and single letter conjunctions. Obviously the previous features of Arabic word increase its ambiguity. Generally Arabic stems can be attached three types of clitics order in their closeness to the stem according to the following formula:

{{[proclitic1] {[proclitic2] {Stem [Affix][Enclitic] }}}

Where proclitic1 is the highest level clitics that represent conjunctions and is attached at the beginning such as the conjunction [(و, w, ‘and’), (ف, f, ‘then’)]. Proclitic2 represent particles [(ب, b, ‘with/in’), (ل, l, ‘to/for’), (ك, k, ‘as/such’)]. Enclitics represent pronominal clitics and are attached to the stem directly or to the affix such as pronoun [(ه, h, ‘his’), (هم, hm, ‘their/them’)].

The following is an example of the different morphological segments in the word **وبقدرته** that has the stem (قدر, qdr, power), the proclitic conjunction (و, w, ‘and’), the proclitic particle (ب, b, ‘with/in’), the affix (ات, At, for plural), and the cliticized pronoun (ه, h, ‘his’).

The set of proclitics considered in this work are the particles prepositions {b, l, k}, meaning {by/with, to, as} respectively, the conjunctions {w, f}, meaning {and, then} respectively. Arabic words may have a conjunction and a preposition and a determiner cliticizing to the beginning of a word. The set of possible enclitics comprises the pronouns and (possessive pronouns) {y, nA, k, kmA, km, knA, kn, h, hA, hmA, hnA, hm, hn}, respectively, my (mine), our (ours), your (yours), your (yours) [masc. dual], your (yours) [masc. pl.], your (yours) [fem. dual], your (yours) [fem.pl.], him (his), her (hers), their (theirs) [masc. dual], their (theirs)

[fem. dual], their (theirs) [masc. pl], their (theirs) [fem. pl.]. An Arabic word may only have a single enclitic at the end. We define a token as a (stem + affixes), proclitics, enclitics, or punctuation.

The data used for training and testing the stemmer and the POS tagger is the Arabic Treebank part 1 [1] which consists of 734 news articles (140k words corresponding to 168k tokens after semi-automatic segmentation) covering various topics such as sports, politics, news, etc.

3 RELATED WORK

A lot of the existing systems tend to target a specific application or a POS tag set that is not general enough for different applications. For example Shereen Khoja in (2001) [10] reports preliminary results on a hybrid, statistical and rule based, POS tagger, APT. APT yields 90% accuracy on a tag set of 131 tags including both POS and inflection morphology information. Diab et al. (2007) [1] perform a large-scale corpus-based evaluation of their approach. They use Yamcha SVM classifier based learner for three different tagging tasks: word tokenization, POS tagging and base phrase chunking with a collapsed tag set achieving a $F_{\beta=1}$ score of 99.12 on word tokenization and an accuracy of 96.6% on POS tagging respectively. Diab (2009) [7] extended the work on Diab et al. (2007) to multiple tag set, instead of the PATB (Penn. Arabic Treebank) reduced tag set. Habash and Rambow (2005) [2] use SVM classifier for individual morphological features and an ad-hoc combining scheme for choosing among competing analysis achieving an accuracy of 97.5%. Mansour (2007) [6] port an HMM Hebrew tagger to Arabic yielding to an accuracy of 96.1% for POS tagging. AlGahtani et al. (2009) [4] use transition based learning for the task of POS tagging, achieving an accuracy of 96.9%. Kulick, S. (2010) [5] performs simultaneous tokenization and POS tagging without a morphological analyzer, achieving an accuracy of 95.1% for POS tagging.

It is not a simple matter to compare results with previous work, due to differing evaluation techniques, data sets, and POS tag sets. In this paper, we focus on comparing our results with Diab et al. (2007) (SVM system) and Habash and Rambow (2005) (Majority system); because both of those papers and both of them work on the same range of data PATB (Penn. Arabic Treebank) part1, they report the results based on the PATB reduced tag set, they assume gold tokenization for evaluation of POS results, and the main concern is to report the highest accuracy unlike AlGahtani et al. (2009) and Kulick, S. (2010) where their main concern is the speedup.

4 TOKENIZATION PHASE

In this phase, the classifier takes an input of raw text, without any processing, and assigns each character the appropriate tag from the following tag set {B-PRE1, B-PRE2, B-WRD, I-WRD, B-SUFF, I-SUFF}. Where I denotes inside a segment, B denotes beginning of a segment, PRE1 and PRE2 are proclitic tags, SUFF is an enclitic, and WRD is the stem plus any affixes and/or the determiner Al. Two experiments have been conducted to achieve the final tokenizer: base line and binary feature experiments. The base line experiment is used to check the effect of using a CRF classifier instead of a SVM classifier in the task of tokenization. In the binary feature experiment a new feature has been proposed in addition to the features used in the base line experiment, and the effect of the binary feature in the task of tokenization is checked.

A. Base Line Experiment (CRF-TOK)

This experiment is based on the experiment of (Diab et al., 2007) but instead of using SVM classifier the CRF suite classifier¹ is used. The classifier training and testing data is characterized as follows:

- Input: A sequence of transliterated Arabic characters processed from left-to-right with break markers for word boundaries.
- Context: A fixed-size window of -5/+5 characters centered at the character in focus.
- Features: All characters and previous tag decisions within the context.

B. Binary Feature Experiment (BF-TOK)

A new feature is proposed in this experiment and this feature is added to the feature set in the baseline experiment. BAMA-v2.0 (Buckwalter Arabic morphological analyzer version 2.0) is used to define a binary feature of length 6 where each bit in the feature is mapped to one of the 6 tags in the tokenization tag set. A bit is set if at least one analysis in the morphological analyses of the word, the character is assigned the tag corresponding to the bit.

¹ CRF suite software package available at: <http://www.chokkan.org/software/crfsuite/>

For example the word (وحيد, wHyd) has two possible tokenization schemes: (و\حيد, w\Hyd) or (وحيد, wHyd); then (و, w) could be (B-PRE1 or B-WRD) then in the binary feature of the character there will be 2 bits set which map to B-PRE1 and B-WRD, (ح, H) could be (B-WRD or I-WRD) then in the binary feature of the character there will be 2 bits set which map to B-WRD and I-WRD, (ي, y) and (د, d) could be only (I-WRD) then in the binary feature of the characters there will be only one bit set which map to I-WRD. The following table shows the binary feature of each character of the word (وحيد, wHyd).

TABLE I
TOKENIZATION BINARY FEATURE

Arabic Letter	Transliterated Letter	Binary feature					
		B-PRE1	B-PRE2	B-WRD	I-WRD	B-SUFF	I-SUFF
و	w	1	0	1	0	0	0
ح	H	0	0	1	1	0	0
ي	y	0	0	0	1	0	0
د	d	0	0	0	1	0	0

If the word is not analyzed by the morphological analyzer (out of vocabulary); then all 7 bits of the binary feature will be set.

5 POS TAGGING PHASE

In this phase, the classifier takes an input of tokenized text, and it assigns each token an appropriate POS tag from the Arabic Treebank collapsed POS tags, which comprises 24 tags² as follows: {ABBREV, CC, CD, CONJ+NEG PART, DT, FW, IN, JJ, NN, NNP, NNPS, NNS, NO FUNC, NUMERIC_COMMA, PRP, PRP\$, PUNC, RB, UH, VBD, VBN, VBP, WP, WRB}. Two experiments have been conducted to achieve the final POS tagger. The first experiment is used to check the effect of using a CRF classifier instead of a SVM classifier in the task of tokenization. In the second, the binary feature experiment a new feature has been proposed in addition to the features used in the base line experiment, and the effect of the binary feature in the task of POS tagging is checked.

A. Base Line Experiment (CRF-POS)

This experiment is based on the experiment of (Diab et al., 2007) but instead of using SVM classifier a CRF classifier is used. The classifier training and testing data is characterized as follows:

- Input: A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.
- Context: A window of -2/+2 tokens centered at the focus token.
- Features: Every character N-gram, $N \leq 4$ that occurs in the focus token, the 5 tokens themselves, POS tag decisions for previous tokens within context.

B. Binary Feature Experiment (BF-POS)

A new feature is proposed in this experiment and this feature is added to the feature set in the baseline experiment. BAMA-v2.0 (Buckwalter Arabic morphological analyzer version 2.0) is used to define a binary feature of length 24 where each bit in the feature is mapped to one of the 24 tags in the collapsed POS tag set. A bit is set when its corresponding tag exists in the morphological analysis of a token.

For example the word (كتب, ktb) has 3 different reduced POS tags: VBD means (write), VBN means (be written), and NN means (book); so there will be 3 bits set to one in the binary feature of the (كتب, ktb) word corresponding to VBD, VBN and NN as shown in Table II

TABLE III
POS TAGGING BINARY FEATURE EXAMPLE

Arabic word	Transliterated word	Binary feature					
		VBD	VBN	NN	JJ	NNS	...
كتب	ktb	1	1	1	0	0	0

² The Penn. Arabic Treebank reduced POS tag set and its map to Penn Arabic Treebank POS tag set available at: <http://www.ircs.upenn.edu/arabic/Jan03release/arabic-POSTags-collapse-to-PennPOSTags.txt>

But for the word (يكتب, yktb) it has only one reduced POS tag: VBP which means (write); so there will be only one bit set in the binary feature which map to VBP. If the word is not analyzed by the morphological analyzer (out of vocabulary) like the word (الفالوجة, AlfAlwjp) which is a village in Palestine, then there will be 5 bits set in the binary feature which map to JJ, NN, NNS, NNP, and NNPS.

6 EVALUATION

For the evaluation of these experiments, k-fold algorithm was used by setting the parameter k to five so the Penn Arabic tree bank part1 is randomly partitioned into five portions of equal size. In each iteration of the k- fold algorithm four portions were used for training the model and one portion was used for testing the model. The cross-validation process is then repeated five times (the folds), with each of the k subsamples used exactly once as the testing data. The five results from the folds were averaged to produce the model evaluation. This evaluation scheme was applied for both the tokenization and POS tagging. Then the following performance measures are calculated for each experiment

$$\begin{aligned} \text{macro average precision} &= \frac{1}{n} * \sum_{i=1}^n \text{precision}(\text{tag}(i)) \\ \text{macro average recall} &= \frac{1}{n} * \sum_{i=1}^n \text{recall}(\text{recall}(i)) \\ \text{macro average } F_{\beta=1}(\text{Fmeasure}) &= \frac{1}{n} * \sum_{i=1}^n F_{\beta=1}(\text{tag}(i)) \end{aligned}$$

Where n is the total number of tags.

$$\text{Accuracy} = \frac{\text{number of true results}}{\text{number of true and false results}}$$

Then the proposed method is compared with the SVM based approach [1] and the Majority system [2]. The comparison between the proposed method and the SVM approach and the majority system will be in the accuracy and the $F_{\beta=1}$ of the tokenizer and in the accuracy of the POS tagger, because these are the only performance measures they have reported. The tool used for evaluation is the evaluation tool in the *CRF- Suite* software package.

A. Tokenization Phase Evaluation

The following table compares the different experiments applied to the Tokenization task where the row represents the experiment and the column represents the macro average performance measure.

TABLE IV
TOKENIZATION PHASE EVALUATION

	Precision	Recall	$F_{\beta=1}$	Accuracy	Error
CRF-TOK	0.99835	0.99926	0.99880	99.98%	0.02%
BF-TOK	0.99998	0.99908	0.99952	99.99%	0.01%

The performance of BF-TOK is almost perfect. Comparing BF-TOK to other Arabic tokenizers like: SVM-TOK which has an accuracy of 99.77% and an F score of 99.12; and with the Majority-TOK which has an accuracy of 99.3% and an F score of 99.1; we found that our improved stemmer reduces the error by about 95.65% compared to the SVM-TOK, and 98.57% from the Majority system tokenizer.

B. Pos Tagging Evaluation

The following table compares the different experiments applied to the POS tagging task where the row represents the experiment and the column represents the macro average performance measures.

TABLE IV
POS TAGGING PHASE EVALUATION

	Precision	Recall	$F_{\beta=1}$	Accuracy	Error
CRF-POS	0.83279	0.77210	0.79130	96.10%	3.9%
BF-POS	0.84872	0.81236	0.82695	98.05%	1.95%

The BF-POS is compared with other Arabic POS taggers like: SVM-POS which has an accuracy of 96.6%, and the Majority-POS which has an accuracy of 97.6%. The result was that the proposed POS tagger reduces the error by 42.65% compared to the SVM-POS tagger and by 18.75% compared to the Majority POS tagger.

7 CONCLUSIONS

We have presented an improved stemmer and POS tagger. Using the benchmark data set improvements in both tokenization and POS stages have been reached. First the CRF classifier is used instead of SVM. This resulted in an error reduction by 91.30% in the tokenization stage. Then the new binary feature (BF) extracted from the morphological analyses of the word is added to the feature set. This binary feature is language independent and highly accurate. It resulted in an error reduction by 95.65 % and 18.75% in the tokenization and POS stage, respectively.

To achieve the targeted improvement the proposed system needs extra processing for the extraction of the binary feature. This extra processing could be minimized by using caching techniques in the implementation of the task of binary feature (BF) extraction.

8 FUTURE WORK

The proposed binary feature BF will be tested on other languages like English. In addition, the performance of the Arabic POS tagging system additional features will be developed to further improve the performance. Last but not least, a wider context and more data will be used for testing.

REFERENCES

- [1] M. Diab, K. Hacioglu, and D. Jurafsky: "Automated methods for processing Arabic text: From tokenization to base phrase chunking", in *Antal van den Bosch and Abdelhadi Soudi*, editors, Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer, 2007.
- [2] N. Habash and O. Rambow, "Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop", in *Proc. of the American Association of Computational Linguistic Conference (ACL) Short Papers*, Michigan, USA, 2005.
- [3] N. Habash and O. Rambow, "Morphological analysis and generation for Arabic dialects" in *Proc. of the Workshop on Computational Approaches to Semitic Languages in the American Association of Computational Linguistic Conference (ACL)* , Michigan, USA, 2005.
- [4] S. AlGahtani, W. Black and J. McNaught, "Arabic Part-Of-Speech Tagging Using Transformation-Based Learning", in *Proc. of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April 2009.
- [5] S. Kulick, "Simultaneous Tokenization and Part-Of-Speech Tagging for Arabic without a Morphological Analyzer", in *Proc. of the American Association of Computational Linguistic (ACL) Conference Short Papers*, Uppsala, Sweden, July, 2010.
- [6] S. Mansour, K. Sima'an and Y. Winter, "Smoothing a Lexicon-based POS tagger for Arabic and Hebrew", in *Proc. of the American Association of Computational Linguistic Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*. Prague, Czech Republic, 2007.
- [7] M. Diab, "Second generation tools (AMIRA 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking", in *Proc. of 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, Egypt, April, 2009.
- [8] M. Maamouri, A. Bies, and T. Buckwalter, "The penn arabic treebank : Building a largescale annotated arabic corpus", in *Proc. of NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2004.

[9] E. Tamah, J. L. Al-Shammari, "*Towards an Error-Free Arabic Stemming*", in *Proc. of the American Association of Computational Linguistic (ACL) Conference on Information and Knowledge Management*, New York, NY, USA, 2008.

[10] S. Khoja, P. Garside, and G. Knowles, "*A tagset for the morphosynactic tagging of Arabic*", in *Proc. of Corpus Linguistics*, Lancaster University, Lancaster, UK, March, 2001.

أثر الصرف في بناء المعجم الحاسوبي العربي

أ. د. علي فرغلي
أستاذة اللغويات بآداب القاهرة - أستاذ م - كلية الحاسبات - ج القاهرة
alifarghaly@yahoo.com

أ. د. وفاء كامل فايد
أستاذة اللغويات بآداب القاهرة - أستاذ م - كلية الحاسبات - ج القاهرة
wafkamel@yahoo.com

يوسف أبو عامر
معيد بآداب القاهرة
ysfaboamer@gmail.com

المستخلص

تحتاج نظم المعالجة الآلية للغات الطبيعية - من أكثرها بساطة إلى أكثرها تعقيدًا - إلى معرفة بالمفردات؛ أي إلى معجم يغطي الاستخدام الفعلي لمفردات اللغة، وكذلك المعلومات اللغوية المتعلقة بها، وبالتالي تمثل الإجابة عن السؤال المتعلق بالأسس التي يجب بناء المعجم الحاسوبي العربي عليها، وطبيعة المحتوى اللغوي للمداخل المعجمية أهمية كبيرة قبل الشروع في بناء المعجم ذاته. ويبرز الصرفُ عاملًا مهمًا ومؤثرًا في الإجابة عن هذا السؤال؛ نتيجة للعلاقة الصرفية المعجمية التي تتسم بها اللغة العربية، ولأن اختيار المداخل المعجمية قائم على أسس صرفية بحتة. وتحاول هذه الورقة الإجابة عن هذا السؤال من خلال تحليل اتجاهات درس الصرف الحاسوبي، وتتبع فرضياتها وتأثيرها في بناء المعجم من الزاويتين: اللغوية والحاسوبية. كما تتناول الورقة طبيعة المحتوى الصرفي الذي يُرمز في المعجم الحاسوبي باعتباره مكونًا أساسيًا لبنية الوحدة المعجمية، وذلك من خلال تعرف السمات التصريفية التي تجعل المدخل المعجمي موافقًا للسياق النحوي.

الكلمات المفاتيح

المعجم الحاسوبي - الصرف الاشتقائي - الصرف التصريفي - المعالجة الآلية للغات الطبيعية - الجذر والوزن - الجذع

أولاً: المقدمة

تحتاج نظم المعالجة الآلية للغات الطبيعية إلى معرفة بالمفردات؛ أي إلى معجم يغطي الاستخدام الفعلي لمفردات اللغة، وكذلك المعلومات اللغوية المتعلقة بها، وبالتالي تمثل الإجابة عن السؤال المتعلق بالأسس التي يجب بناء المعجم الحاسوبي العربي عليها، وطبيعة المحتوى اللغوي للمداخل المعجمية أهمية كبيرة قبل الشروع في بناء المعجم ذاته. ويبرز الصرف عاملاً مهماً ومؤثراً في الإجابة عن هذا السؤال؛ نتيجة للعلاقة الصرفية المعجمية التي تتسم بها اللغة العربية، ولأن اختيار المداخل المعجمية قائم على أسس صرفية بحتة. وتحاول هذه الورقة الإجابة عن هذا السؤال من خلال تحليل اتجاهات الدرس الصرف الحاسوبي، وتتبع فرضياتها وتأثيرها في بناء المعجم من الزاويتين: اللغوية والحاسوبية. كما تتناول الورقة طبيعة المحتوى الصرفي الذي يرمز في المعجم الحاسوبي باعتباره مكوناً أساسياً لبنية الوحدة المعجمية، وذلك من خلال تعرف السمات التصريفية التي تجعل المدخل المعجمي موافقاً للسياق النحوي.

ثانياً: المعجم ودراسات الصرف العربي الأولية

اعتبرت الدراسات الأولية للصرف العربي الجذر الثلاثي مع وزن الكلمة الوجدتين الصرفيتين اللتين تتحكمان في بنية اللغة [2]؛ فكلمة "كَتَبَ kataba" هي الشكل السطحي الذي يتحلل إلى الجذر الثلاثي (ك ت ب ktb) والوزن (فَعَلَ faʕala)

فالجذور – تبعاً لهذا الاتجاه في التحليل الصرفي- هي الأصول اللغوية للكلمات؛ أي الحروف الأصلية التي تُشتق منها الكلمات، ومعظم الكلمات العربية ثلاثية الأصل، وهناك عدد قليل رباعي، أو خماسي. ويمكن القول إن جذر الكلمة هو حروفها مجردة من الزوائد وغير مقترنة بصيغة، ويحمل الجذر المعنى الأساسي للمجموعات المعجمية المشتقة منه؛ فالجذر الثلاثي (ك ت ب k t b) يحمل المعنى الأساسي للكتابة، وتشارك في هذا المعنى مشتقاته المختلفة، مثل كَاتَبَ، وکِتَابَةٌ، وکُتِبَ، وکُتُبٌ، إلخ. وقد اتخذت معظم المعاجم العربية منذ بداية تأليفها جذر الكلمة أساساً لتورد تحته كافة أنواع المشتقات اعتماداً على هذين المبدأين الأساسيين:

- الجذر هو الأساس الذي تشتق منه الكلمة.
- المشتقات المختلفة للجذر الواحد تشارك – ولو جزئياً- في معنى أساسي يحمله الجذر.

ثالثاً: المعجم ودراسات الصرف الحاسوبية

1. الصرف النظامي

نالت دراسات الصرف العربي على المستوى الحاسوبي اهتماماً كبيراً من جانب مهندسي اللغة واللغويين الحاسوبيين منذ بداية الثمانينيات؛ وتفاعلت هذه الدراسات مع الجانب المعجمي تفاعلاً كبيراً؛ إذ اقترحت باكورة الأعمال الخاصة بالصرف الحاسوبي العربي أن تكون فكرة الجذور والأوزان أساس تحليل الكتابة العربية المشكلة تشكيلاً كاملاً. [4]

وقد اعتبر أصحاب هذا الاتجاه أن طبيعة اللغة العربية تجعل من الصرف النظامي Templatic Morphology المعتمد على النظرية العروضية؛ أي على فكرة الأوزان الصرفية، أكثر قدرة على التعامل مع الطبيعة غير التسلسلية للصرف العربي، وغيره من اللغات التي تتفق مع العربية في هذه الطبيعة الصرفية.

واعتمادًا على هذا الاتجاه في التحليل الصرفي صُمم عدد من النظم الحاسوبية لتحليل الكلمات العربية أو توليدها. وتمثلت الطريقة الأساسية لبناء هذه النظم في عمل قاموس للجذور العربية وقواميس للواحق، مع مراعاة التمييز بين السوابق والواحق والداخل، أو عمل معجم للجذور والأوزان تضاف إليه قوائم بالعناصر السابقة أو اللاحقة. [4]

وقد وُجّهت مجموعة من الانتقادات لهذا الاتجاه على أساس وجود بعض جوانب القصور، ومن ثم كانت الدعوة إلى تبني اتجاه بديل عند بناء قواعد البيانات المعجمية الحاسوبية. ولعل من أبرز الانتقادات التي وُجّهت لهذا الاتجاه:

• تمثيل الجذر والصيغة يصلح مع مجموعة فرعية من المعجم، في حين لا يمكن تطبيق الفكرة ذاتها على مجموعات أخرى أساسية [3]

لا يمكن تطبيق فكرة الجذر والوزن على الكلمات المعرّبة، مثل (بستان- سراط- فسفات- منجنيز- نيلون- سكرتير- بورصة- بنك- سندس- إستبرق- بلطجي)، كما لا يمكن تطبيقها على كثير من أسماء الأعلام (أشبيلية- غرناطة- دمشق- الفسطاط - إسماعيل- سيبويه).

• تمثيل الجذر والصيغة يصلح - أساسًا - مع الأفعال والمشتقات الفعلية [3]

تسري العلاقات الاشتقاقية من صيغة إلى أخرى - بشكل أساسي- على نطاق الأفعال والمشتقات الفعلية، كالمصدر، واسم الفاعل، واسم المفعول، ومن ثم يمكن تحليل الأفعال العربية والمشتقات الفعلية على أساس الجذر والوزن، لكن هذا الاتجاه يترك عددًا كبيرًا من المداخل المعجمية دون تمثيل.

• عدم انتظام الفرضية القائلة بأن المشتقات المختلفة للجذر الواحد تشترك في المعنى نفسه [3]

يدلل ديشي وفرغلي على عدم انتظام هذه الفرضية بأحد أشهر الأمثلة المستخدمة في الدراسات الصرفية، وهو الجذر (ك ت ب)؛ إذا يفترض أن المشتقات المعجمية المختلفة التي تحتوي على هذا الأصل الثلاثي تشترك في الحقل الدلالي للكتابة، على النحو الذي نراه في كُتِب، كَاتِب، ومكتب، ومكتبة، وهو ما لا يبدو واضحًا - بالنسبة لمتحدثي العربية في الوقت الحالي- عند اشتقاق كلمة نحو "كتيبة"، التي يحتاج ربطها الدلالي بالجذر إلى دراسة تاريخية معنية بتطور اللغة وتغيرها عبر التاريخ.

وقد حاولت الدراسة تتبع هذه الفرضية من خلال البحث في المشتقات المعجمية لعدد من الجذور المختارة بطريقة عشوائية، وانتهت إلى أن اشترك بعض الكلمات في حروف المادة له دلالاته على الاتصال الشكلي بين هذه الكلمات، دون أن يعني - بالضرورة- اتصالاً بينها على مستوى الدلالة، كما يظهر من الأمثلة التالية:

○ الجذر اللغوي (ر ق ب) يرتبط أساسًا بمعنى المراقبة، ومنه الفعل (راقب)، واسم الفاعل (مراقِب) بضم الميم وكسر القاف، واسم المفعول (مراقَب) بضم الميم وفتح القاف، والمصدر (مراقبة)، لكننا نحصل من الجذر ذاته على كلمة (رقبة) التي تعني مؤخر أصل العنق.

○ الجذر اللغوي (ش ك و) يرتبط - أساسًا- بمعنى الشكوى، ومنه الفعل شكأ، واسم الفاعل الشاكي، واسم المفعول مشكوى، والمصدر شكاية، والاسم شكوى، لكننا نحصل من الجذر ذاته على كلمة (مشكاة) التي تعني الكؤوة التي ليست بنافذة، وفي القرآن الكريم ﴿اللهم نور السماوات والأرض مثل نوره كمشكاة فيها مصباح﴾ نقل ابن كثير عن غير واحد أن المشكاة: موضع الفتيل من القنديل.

- الجذر اللغوي (ب ل د) نحصل منه على كلمة (البلد) بمعنى الموطن أو المستقر، ونحصل منه أيضا على كلمة (البلادة) وهي نقيض الذكاء، ولا يوجد التقاء بينهما في المعنى.
- الجذر اللغوي (ر ف ق) نحصل منه على كلمة (الرفق) بمعنى اللين، وكلمة (الرفقة) وهي الجماعة في السفر، ونحصل منه أيضا على كلمة (المِرْفَق) وهو موصل الذراع في العضد، ولا يوجد التقاء بينهما في المعنى.
- الجذر اللغوي (ب ر ج) نحصل منه على كلمة (برج) وهو الحصن والجمع (أبراج)، ومنه قوله تعالى ﴿أينما كونوا يدرككم الموت ولو كنتم في بروج مشيدة﴾ ونحصل منه على (التبرج) وهو إظهار المرأة زينتها للرجال، ونحصل منه كذلك على (البارجة)، وهي نوع من المقاتلات الحربية تعد الأضخم بعد حاملة الطائرات. ولا توجد علاقة دلالية بين هذه الكلمات.

● الشذوذ وعدم الانتظام الاشتقائي

كثيرا ما تترد في كتب الصرف جملة "وما جاء مخالفاً للقواعد شاذ، يحفظ ولا يقاس عليه"، كما في الأمثلة التالية:

- القاعدة: يشتق اسما الزمان والمكان من الثلاثي على وزن (مَفْعَل) بفتح الميم والعين إذا كان مضارعه مضموم العين، نحو: (أَكَل - يَأْكُل - مَأْكَل)، و (بَلَّغ - يَبْلُغ - مَبْلُغ)، لكننا نجد (مَسْجِد) بكسر العين من الثلاثي (سَجَد - يَسْجُد)، ونحو ذلك (مَطْلَع) من الثلاثي (طَلَع - يَطْلُع).

- القاعدة: يشترط في اشتقاق اسم التفضيل على وزن (أفعل) أن يكون له فعل، ولكننا نجد: (هذا البعير أحنك الإبل)، أي أكثرهم أكلا بحكته؛ فقد اشتق اسم التفضيل من (الحنك)، والحنك اسم وليس فعلا. ومن ذلك أيضا (سعد ألس من غيره)، أي أكثر منهم لصوصية؛ فقد اشتق اسم التفضيل من (اللس) وليس بفعل، وإنما هو اسم.

- تشيع بصورة كبيرة المصادر غير القياسية، لا سيما مصادر الثلاثي حتى شاع الرأي بين الصرفيين على أنها سماعية، أي قد نجد أوزانا للمصادر على غير قياس من ناحية، وقد نجد للوزن الواحد من الأفعال مصادر متعددة الصيغ من ناحية أخرى.

فمن أوزان المصادر غير القياسية:

- (فُعَل) بضم الفاء وفتح العين، نحو: هدى.
 - (فُعَلَة) بفتح الفاء وسكون العين، نحو: رحمة.
 - (فَعَل) بفتح الفاء والعين معا، نحو: طلب.
- ومن قبيل تعدد صيغ المصادر للأفعال ذات الوزن الواحد مصادر الأفعال الدالة على الأصوات، نحو: صرخ، وعوى، وبكى، ونهق، وقد ورد فيها أوزان ثلاثة، هي:
- (فُعَال) بضم الفاء وفتح العين، نحو: صراخ، وبكاء.
 - (فِعَال) بكسر الفاء وفتح العين، نحو: زمار (صوت النعام).
 - (فَعِيل) بفتح الفاء وكسر العين، نحو: نهيق.

- اختلاف المعجميين واللغويين حول الجذر اللغوي لبعض الكلمات: توجد كلمات عربية لا يقطع المعجميون بجذرها الحقيقي؛ فكلمة (ميناء)-مثلا- تدرجها معاجم تحت الجذر (و ن ي) وأخرى تحت الجذر (م ن أ)، وثالثة تحت الجذر (م ن أ).

2. الصرف ثلاثي المستوى

على نحو مختلف، يرى فرغلي أن التوصيف الدقيق للصرف العربي يجب أن يميز ثلاثة مستويات: أولها هو الجذر، وهو غير منطوق وغير مصنف على مستوى الأنواع المعجمية أو أنواع الكلام، والثاني الجذع، وهو منطوق ويجب أن يكون أحد الأنواع المعجمية في اللغة، والثالث الكلمة المتصرفة؛ إذ تتصل بها اللواحق لتكوين معظم الكلمات العربية الفعلية [6]، كما في المثال الموضح بالشكل التالي.

شكل رقم (1)

يوضح المستويات الثلاثة عند علي فرغلي

المستوى الأول ←	الجذر	←	كُتِبَ ب (غير منطوق – غير مصنف)
المستوى الثاني ←	الجذع	←	كَتَبَ (منطوق – مصنف)
المستوى الثالث ←	الكلمات المتصرفة	←	يكتب- يكتبون- تكتبون- كتبوا- كتبن ..

3. الصرف المعتمد على الوحدة المعجمية

يقترح سعودي وآخرون اتجاهًا في التحليل يقوم على استخدام نظرية الصرف المعتمد على الوحدة المعجمية lexeme-based morphology، كما اقترح نظامًا حاسوبيًا للتطبيق أطلق عليه MORPHE [11].

ويقصد بالوحدة المعجمية التجريد المعجمي للكلمات التي تشترك في معنى أصلي واحد، لكنها تختلف في التصريف فقط؛ فالوحدة المعجمية "بيت" التي تشير - دلاليًا - إلى المسكن أو المنزل تتضمن الكلمات "بيت" و"بيوت"، في حين أن الوحدة المعجمية "بيت" التي تشير - دلاليًا - إلى بيت الشعر تتضمن الكلمات "بيت" و"أبيات". ويرى سعودي أن الوحدة المعجمية تعد تجميعًا للصيغة الصوتية، والنحو، والدلالة، في حين يوافق الجذع أو الجذر جزءًا من هذا التجميع وهو الجزء الخاص بالصيغة الصوتية فحسب.

3. الصرف الشكلي في مقابل الصرف الوظيفي

يعتمد حبش في دراسة أكثر تفصيلاً للصرف على التمييز بين إطارين للدراسة: أحدهما شكلي والآخر وظيفي، متأثرًا في تقسيمه بأوتكار سمارتش Otakar SmrŽ، الذي يميّز هو الآخر بين صرف شكلي وآخر وظيفي [10].

يهتم الصرف الشكلي بصيغ الكلمة، وتفاعلها مع بعضها، وعلاقتها بالصيغة العامة للكلمة، في حين يُعنى الصرف الوظيفي بوظائف هذه الصيغ، وتأثيرها على السلوكين النحوي والدلالي للكلمة [7]. ففي الصرف الشكلي يعد المورفيم - وهو أصغر وحدة لغوية تحمل معنى معجميًا أو نحويًا [1] - مفهومًا مركزيًا، وتقسّم المورفيمات - عادة - إلى: مورفيمات حرة، ويقصد بها المورفيمات التي يمكن أن تظهر في صورة كلمات مستقلة، دون حاجة إلى أي تعديلات صرفية [3].

ومورفيمات مقيدة، وهي التي لا يمكنها الاستقلال بذاتها داخل الكلام، كاللواحق [3]. واعتمادًا على هذا التقسيم للمورفيمات يمكن تقسيم الكلمات إلى: كلمات أحادية المورفيم، وهي الكلمة التي تحتوي على مورفيم واحد "حر"، كما في الفعل "كَتَبَ"، وكلمات متعددة المورفيمات، وهي الكلمات التي تتكون من أكثر من مورفيم، مثل كلمة "يَكْتُبُونَ".

ويرى حبش أن اللغة العربية تتسم - شأن غيرها من اللغات السامية - بوجود المورفيمات النظامية Templatic Morphemes، والمورفيمات التسلسلية Concatenative Morphemes. وتؤدي

المورفيمات التسلسلية دوراً مهماً في تكوين الكلمات من خلال عملية تسلسل منطقية للمورفيمات، في حين تندمج المورفيمات النظامية معاً لتكون الكلمات [7].

وتنقسم المورفيمات التسلسلية بدورها إلى: جذوع، ولواحق، ولواصق Clitics. ويعد الجذع أساس المورفيمات التسلسلية، حيث تلتصق به اللواحق، وتكون إما سابقة عليه، وإما لاحقة له، أو محيطة به Cirumfixes. ويمكن التمثيل لهذه الأنواع من اللواحق على النحو التالي:

جدول رقم (1)

أنواع اللواحق في اللغة العربية

الكلمة	اللاحقة	نوع اللاحقة	المقابل الإنجليزي
يَكْتُب	يـ	سابقة Prefix	He writes
كَتَبُوا	وا	لاحقة Suffix	They wrote
تَكْتُبِينَ	تـ - ين	محيطة Cirumfix	You write

أما اللواصق Clitics فهي وحدات لغوية تشبه الكلمات، لكنها ليست مورفيمات حرة؛ لأنها لا تستقل بذاتها في الكلام، وإنما تتصل بكلمة مجاورة. وهي تختلف عن اللواحق باستقلالها- الصوتي والنحوي- عن الكلمة التي تتصل بها، بخلاف اللواحق التي تعد جزءاً من الكلمة صوتياً ونحوياً. وتنقسم اللواصق Clitics إلى قسمين: لواصق قَبَلِيَّة Proclitics، وهي التي تسبق الكلمة التي تتصل بها، ولواصق بَعْدِيَّة Enclitics وهي التي تلي الكلمة التي تتصل بها [9].

وعلى الجهة الأخرى تنقسم المورفيمات النظامية إلى: جذور، وصيغ (قوالب) Patterns، وحركات Vocalisms. فالجذور هي الحروف الأصلية الصامتة التي تُشتق منها الكلمات. أما الحركات فيقصد بها الصوائت الأساسية المستخدمة في اللغة العربية، وهي: الفتحة (a)، والكسرة (i)، والضممة (u). وتنتشر الجذور والحركات- أو الصوائت- معاً داخل مجموعة من الصيغ، أو القوالب؛ لتكوّن المشتقات المختلفة للجذر اللغوي، على النحو الذي نراه في بعض مشتقات الجذر الثلاثي (ك ت ب k t b) الموضحة بالجدول رقم (2).

جدول رقم (2)

بعض مشتقات الجذر (ك ت ب)

الجذر	الحركات	الصيغة	الكلمة	المقابل الإنجليزي
ك ت ب	a- a	فَعَلَ	كَتَبَ	He wrote

Writer	كَاتِب	فَاعِل	a a- i
Written	مَكْتُوب	مَفْعُول	a - u u
Was written	كُتِب	فُعِل	u- i
Cause to write	كَتَب	فَعَلَ	a- a
Corresponded	كَاتَب	فَاعَلَ	a a- a

أما الصرف الوظيفي فيُعنى بدراسة الكلمة في ضوء سلوكها الصرفي النحوي، والصرفي الدلالي. ويتم التمييز في هذا السياق بين ثلاث عمليات هي: الاشتقاق Derivation، والتصريف Inflection، والإصاق [9] Cliticization.

يستخدم مصطلح الاشتقاق في الصرف للإشارة إلى إحدى العمليات التي يتم بموجبها تكوين الكلمات، ويترتب عليها تغيير المعنى الأصلي للوحدة المعجمية، وتصنيفها النحوي، أي تنتج كلمة جديدة [3]، كما يبدو في المثال التالي:
اشتقاق اسم من فعل:

كَتَبَ (فعل V) ← كَاتِبَ (اسم N) kaatib

فالفعل كَتَبَ kataba يدل على حدث الكتابة مقترن بالزمن الماضي، وقد اشتق منه الاسم كَاتِبَ kaatib الدال على القائم بعملية الكتابة. وهذه العملية تشبه اشتقاق الاسم (مُتَحَدِّث speaker) في اللغة الإنجليزية من الفعل (يتحدث speak). وقد ترتب في الحالتين تغيير نوع الكلمة POS من الفعل إلى الاسم. ومن أبرز المشتقات وأشيعها في اللغة العربية اسم الفاعل، واسم المفعول، والمصدر، وأسماء الزمان والمكان، واسم الآلة، والتصغير، والنسبة.

أما التصريف فيشير إلى العملية الثانية من عمليات تكوين الكلمات، لكنها لا تؤدي إلى تغيير المعنى الأصلي للوحدة المعجمية، ولا إلى تغيير تصنيفها النحوي، وإنما يقتصر التغيير على شكلها، بحيث يلائم السياقات النحوية المختلفة [9]. وهناك ثماني حالات تُصَرَّف على أساسها الكلمة العربية هي: الزمن Tense، والشخص Person، وبناء/صيغة الفعل Voice، والحالة الإعرابية للفعل Mood، والنوع Gender، والعدد Number، والموقع الإعرابي Case، والتعيين Definiteness. وتجدر الإشارة إلى أن الاسم والصفة يصرفان وفقاً لأربع حالات هي: النوع، والعدد، والموقع الإعرابي، والتعريف، في حين يصرف الفعل تبعاً لست حالات هي: الزمن، والشخص، وبناء الفعل، والحالة الإعرابية، والنوع، والعدد، وأخيراً يصرف الضمير باعتبار الشخص، والنوع، والعدد، والحالة إلى حد ما.

وعلى هذا يعد الجذع هو الصورة التي يأخذها الجذر حين يُفَرَّغ في قالب من القوالب الصرفية، ومن ثم تتصل به السوابق واللاحق لتكوين الصيغ المختلفة للكلمة. ويفاد من ذلك أن ثمة فارقاً بين الجذر والجذع يقوم على اعتبار الجذر مادة صوتية خاماً، في حين يُعدّ الجذع تحقّقاً من تحققات الجذر، كما يفاد أن الجذر الواحد قد يكون له غير جذع واحد. وعلى الرغم من استمرار اعتماد بعض الدراسات على فكرة الجذر والوزن في معالجة الصرف العربي، فإن الاتجاه الحاسوبي الذي اقترحته معظم الدراسات الحديثة حول

التعامل الصرفي للوحدة المعجمية- بالنسبة للغة العربية- يعتبر الجذع فحسب هو المناسب؛ إذ تتصرف على أساسه قواعد التحقق. وقد اعتمد المعجم الذي طوره تيم بكوالتن ليكون جزءًا من محله الصرفي على الجذع؛ إذ يحتوي المعجم على قائمة من الجذوع الصرفية مصنفة لأغراض التحليل التصريفي، ومصحوبة بالوحدات المعجمية العربية، والمقابلات الإنجليزية.

ويعد معجم تيم بكوالتن من أشهر المعاجم الحاسوبية، ولعله الأكثر استخدامًا في التطبيقات المختلفة لمعالجة اللغة العربية آليًا؛ فقد اعتمد عليه حبش ورامبو في نظامهما المعروف باسم MAGEAD، كما اعتمد عليه اوتكار سمارتنش في "الصرف العربي الوظيفي" بوصفه مصدرًا معجميًا أساسيًا [8].

رابعًا: معجم قائم على أساس الجذع

تقترح الدراسة اعتماد الجذع أساسًا لبناء المعجم على نحو يتجاوز جوانب القصور في المعاجم المعتمدة على فكرة الجذور والأوزان- من جهة- ويفيد من مميزات بناء المعجم على أساس الجذع من جهة أخرى، وهي مميزات تجعله أكثر فاعلية وسهولة في التطوير والتوسيع.

ومن أبرز مسوغات بناء المعجم على أساس الجذع:

- التخلص من عملية توليد الجذوع من الجذور والأوزان الصرفية؛ فالمحلات الصرفية للغة العربية التي يعتمد معجمها على فكرة الجذر والوزن تحتاج إلى إدخال الجذور والأوزان الصرفية بوصفها معلومات ضرورية لتوليد الجذوع [5]. ولعملية التجذيع أهميتها في بعض التطبيقات، حين لا تكون هناك حاجة إلى تحليل الكلمة، ولكن إلى التوصل من خلال الكلمة إلى الجذع stem؛ ففي استرجاع المعلومات، والبحث على شبكة المعلومات قد نحتاج إلى التوصل من خلال كلمة "مسلمون" إلى الجذع "مسلم" دون الحاجة إلى معرفة أن كلمة "مسلمون" هي صيغة الجمع المتكونة بإضافة مورفيم جمع المذكر السالم "ون". ويطلق على هذا التطبيق "التجذيع Stemming".
 - كل المواد المعجمية تمثل وحدات معجمية فعلية، وليس وحدات افتراضية [5].
 - يعد الجذع نوعًا معجميًا يمكن ربطه بالمعلومات اللغوية على مستويات الصرف، والنحو، والدلالة مثل إطارات التصنيف الفرعي، وبنية الموضوع argument-structure، وذلك بخلاف الجذر غير المصنف على مستوى الأنواع المعجمية.
 - الاعتماد على الجذع في بناء المعجم يساعد على التخلص من الصيغ غير المستعملة، والاقتصار على ما هو مستخدم فعليًا في مدونات اللغة العربية المعاصرة. ولهذه النقطة أهمية خاصة تتعلق بمفهوم شائع في الدراسات اللغوية والمعجمية يعرف بالفجوات المعجمية⁽¹⁾ Lexical gaps؛ فكما يسمح نظام الجذور والأوزان بإنتاج الصيغ اللغوية المستعملة الصحيحة قد ينتج عنه في الوقت ذاته صيغ مهملة، أو غير مستخدمة- فعليًا- عند متحدثي اللغة.
- وتبدو فكرة الفجوات اللغوية واضحة في الجدول رقم (3) الذي يوضح المستعمل والمهمل من صيغ الجذر اللغوي (ه ب ط).⁽²⁾

جدول رقم (3)

(1) يستخدم مصطلح الفجوة في الدراسات اللغوية - بشكل عام - للإشارة إلى غياب وحدة لغوية ما في قالب من العلاقات يكون وجود هذه الوحدة اللغوية متوقعًا. وقد تظهر هذه الفجوات في مستويات الدرس اللغوي المختلفة؛ فهناك الفجوات الفونولوجية Phonological gaps، والفجوات الصرفية Morphological gaps، والفجوات النحوية Syntactic gaps، والفجوات المعجمية Lexical gaps.

(2) اعتمدت الدراسة في تحديد المستعمل من الصيغ الفعلية للجذر (ه ب ط) على ثلاثة معاجم: لسان العرب، وتاج العروس باعتبارهما الأغزر مادة، والمعجم الوسيط لاحتمال استعمال العربية الحديثة صيغًا كانت مهملة.

المستعمل والمهمل من صيغ الجذر اللغوي (ه ب ط).

الاستعمال	الكلمة	الصيغة	الجذر
مستعمل	هَبَطَ	(1) فَعَلَ	ه ب ط
مستعمل	هَبَطَ	(2) فَعَلَ	
غير مستعمل	هَابَطَ	(3) فَاعَلَ	
مستعمل	أَهْبَطَ	(4) أَفْعَلَ	
مستعمل	تَهَبَّطَ	(5) تَفَعَّلَ	
غير مستعمل	تَهَابَطَ	(6) تَفَاعَلَ	
مستعمل	انْهَبَطَ	(7) انْفَعَلَ	
غير مستعمل	اهْتَبَطَ	(8) افْتَعَلَ	
غير مستعمل	اهْبَطَ	(9) افْعَلَ	
غير مستعمل	اسْتَهَبَطَ	(10) اسْتَفْعَلَ	
غير مستعمل	اهْبُوبَطَ	(11) افْعُوعل	

وتنقسم الجذوع في اللغة العربية – تبعًا لديشي وحسون- إلى نوعين عامين:

- **النوع الأول:** يشتمل على المجموعات المعجمية الأساسية التي يمكن تمثيلها باعتبار الجذر والوزن، وتنتمي الأفعال والمشتقات الفعلية إلى هذا النوع، كما في الجذع اللغوي "تَكَبَّرَ" الذي يتألف من الجذر الثلاثي (ك ب ر)، والوزن (تَفَعَّلَ).
- **النوع الثاني:** يطلق ديشي على هذا النوع أشباه الجذوع quasi-stems، ولا يشتمل - بطبيعة الحال- على الأفعال والمشتقات الفعلية، ولكنه يشتمل على بعض الأسماء التي لا يمكن تحليلها باعتبار الجذور والأوزان، كما في الجذر اللغوي "بَرْنَامَج"؛ الذي لا يمكن اشتقاق أي فعل ولا حتى جمع مذكر سالم من الصوامت الخمسة (ب ر ن م ج).

خامسًا: المحتوى الصرفي

يشتمل المحتوى الصرفي للمدخل المعجمي على السمات التصريفية الشكلية للمدخل المعجمي، تلك السمات التي تجعله موافقًا للسياق النحوي. وقد سبقت الإشارة إلى الحالات التي تتصرف بمقتضاها المجموعات المعجمية العربية. وتختلف المجموعات المعجمية فيما بينها من حيث تصرفها أو عدمه تبعًا لكل حالة من هذه الحالات، على النحو الموضح بالجدول (4)، و(5)، و(6) على الترتيب.

جدول رقم (4)

حالات تصريف الاسم والصفة

Gender النوع		
Feminine مؤنث		Masculine مذكر
Number العدد		
Plural جمع	Dual مثني	Singular مفرد
Case الموقع الإعرابي		
Genitive الجر	Accusative النصب	Nominative الرفع
Definiteness التعيين		
Definite معرفة		Indefinite نكرة

جدول رقم (5)

حالات تصريف الفعل

Tense الزمن		
Imperative أمر	Imperfective مضارع	Perfective ماض
Number العدد		
Plural جمع	Dual مثني	Singular مفرد
Mood الحالة الإعرابية		
Jussive الجزم	subjunctive النصب	Indicative الرفع
Gender النوع		
Feminine مؤنث		Masculine مذكر
Voice بناء الفعل		
Passive Voice بناء للمجهول		Active Voice بناء للمعلوم
Person الشخص		
3rd person الغائب	2nd Person المخاطب	1st Person المتكلم

جدول رقم (6)

حالات تصريف الضمير

Person الشخص		
3rd person الغائب	2nd Person المخاطب	1st Person المتكلم
Gender النوع		
Neutral محايد	Feminine مؤنث	Masculine مذكر
Number العدد		
Plural جمع	Dual مثني	Singular مفرد
Case الموقع الإعرابي		
Genitive الجر	Accusative النصب	Nominative الرفع

وتكشف هذه الجداول عن حالات يعتمد فيها التصريف على السياق، أو الموقع الإعرابي للكلمة؛ لذا لن نتطرق إليها الدراسة، ولن تكون جزءاً من المحتوى الصرفي المعجمي، كالموقع الإعرابي للاسم، والصفة،

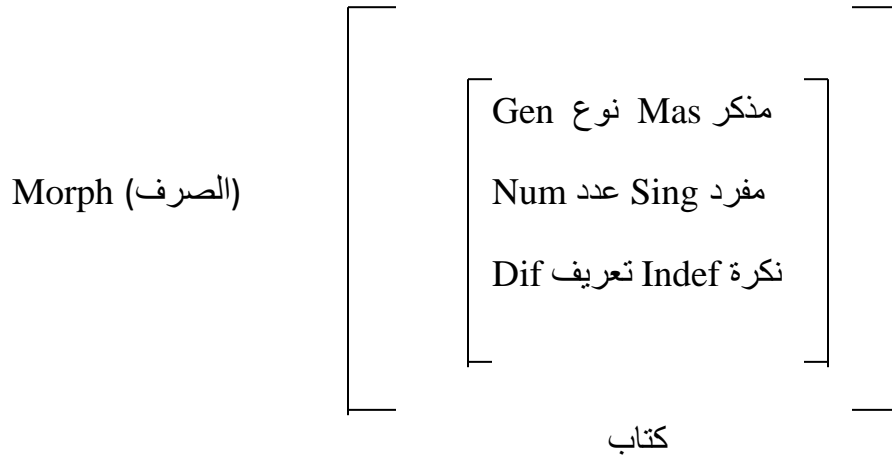
والضمير، وكذلك الحالة الإعرابية للفعل. وعلى هذا يكون المحتوى الصرفي المعجمي مقتصرًا على المعلومات الملازمة للمدخل المعجمي، كالتذكير والتأنيث، والإفراد والتثنية، والمضي والمضارعة والأمرية ... إلخ.

1. التمثيل المعجمي للمحتوى الصرفي

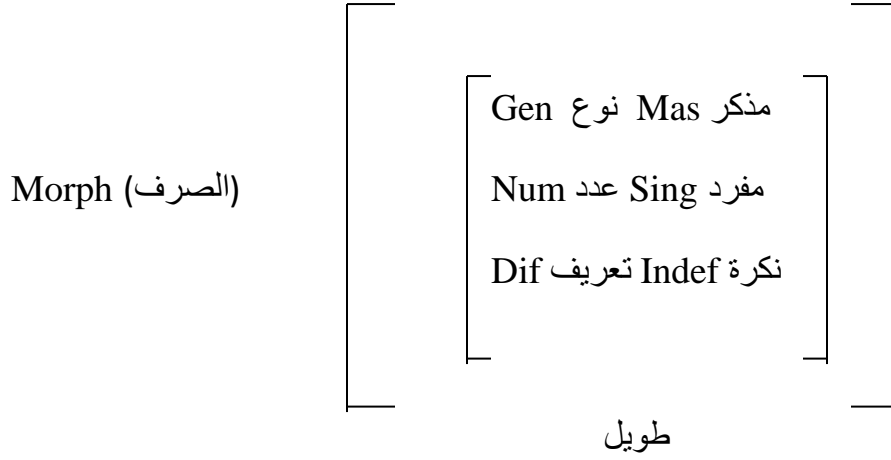
وعلى هذا يكون التمثيل المعجمي للمحتوى الصرفي للاسميات والصفات مشتملاً على المعلومات الصرفية الخاصة بالنوع والعدد والتعيين، كما في التمثيل المعجمي للمحتوى الصرفي للاسم "كتاب"، والصفة "طويل" الموضحين بالشكلين (2)، و(3) على الترتيب.

شكل رقم (2)

يوضح التمثيل المعجمي للمحتوى الصرفي للاسم "كتاب"

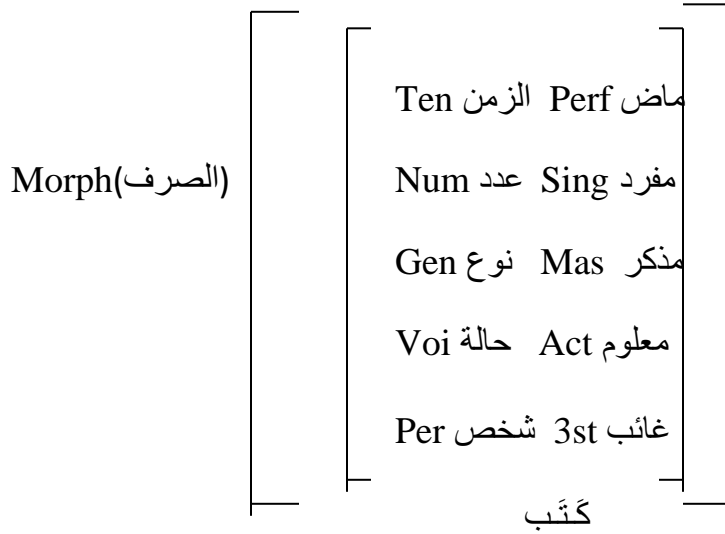


شكل رقم (3)
يوضح التمثيل المعجمي للمحتوى الصرفي للصفة "طويل"

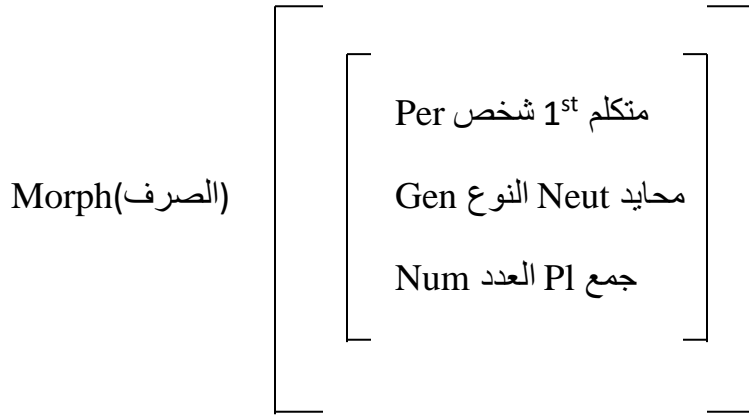


أما التمثيل المعجمي للمحتوى الصرفي للأفعال فيكون مشتملا على المعلومات الصرفية الخاصة بالزمن، والعدد، والنوع، وبناء الفعل، والشخص كما في التمثيل المعجمي للمحتوى الصرفي للفعل "كتب" الموضح بالشكل رقم (4).

شكل رقم (4)
يوضح التمثيل المعجمي للمحتوى الصرفي للفعل "كتب"



أما التمثيل المعجمي للمحتوى الصرفي للضمائر فيكون مشتملا على المعلومات الصرفية الخاصة بالشخص، والنوع، والعدد على النحو المبين في التمثيل المعجمي للمحتوى الصرفي للضمير "نحن" بالشكل رقم (5).



نحن

شكل رقم (5) يوضح التمثيل المعجمي للمحتوى الصرفي للضمير "نحن"

2. قائمة السوابق واللواحق واللواصق

السوابق واللواحق هي أنواع من المورفيمات المقيدة، تلتصق بالجدوع لتكوين الكلمات الفعلية في اللغة. وتنقسم بحسب مكانها من الجذع إلى: سوابق، ولواحق، ومحيطات. وتجدر الإشارة إلى إمكان اجتماع أكثر من سابق واحد قبل الجذع، أو أكثر من لاحق واحد بعد الجذع، كما في كلمة "سيكتبونها" التي تتكون من اللاصقة الدالة على الاستقبال "السين"، والياء الخاصة بمورفيم المضارعة للمذكر الغائب، ثم الجذع "كتب"، واللاحقتين "ون"، و"ها".

وفيما يلي جدولان يحتوي الأول منهما على قائمة ببعض السوابق واللواحق، موضحة السابقة أو اللاحقة، ونوعها، ووظيفتها، مع التمثيل لكل منها، أما الثاني فيشتمل على اللواصق العربية، وهي عبارة عن حروف وأدوات أشارت الدراسة إليها في الجزء الخاص بالمجموعات المعجمية.

جدول رقم (7)

قائمة ببعض السوابق واللواحق

اللاحقة	النوع	الوظيفة	المثال
ني	لاحقة للمفرد المتكلم بنوعيه	(ضمير مفعول)	ضربني
ي	لاحقة للمفرد المتكلم بنوعيه	(مكمل + ضمير ملكية)	لي كتابي
يَ	لاحقة للمفرد المتكلم بنوعيه	(مكمل + ضمير ملكية)	ليَ كتابيَ
نا	لاحقة لجمع المتكلم بنوعيه	(مكمل + ضمير مفعول + ضمير ملكية)	بنا ضربنا كتابنا

بِكَ ضربكَ كتابِكَ	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للمفرد المخاطب المذكر	كَ
بِكِ ضربكِ كتابِكِ	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للمفرد المخاطب المؤنث	كِ
بكما ضربكَ كتابِكَ	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للمثنى المخاطب بنوعيه	كما
بكم ضربكم كتابكم	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للجمع المخاطب المذكر	كم
بكنَّ ضربكنَّ كتابكنَّ	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للجمع المخاطب المؤنث	كنَّ
به ضربه كتابه	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للمفرد الغائب المذكر	هُ
بها ضربها كتابها	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للمفرد الغائب المؤنث	ها
بهم ضربهم كتابهم	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للجمع الغائب المذكر	هم
بهنَّ ضربهنَّ كتابهنَّ	(مكمل + ضمير مفعول + ضمير ملكية)	لاحقة للجمع الغائب المؤنث	هنَّ

جدول رقم (8)

قائمة باللواحق العربية

الرمز المقترح	الوظيفة	اللاصقة
Interrog_Part	الاستفهام	الهمزة (أ)
Equal_Part	المساواة	
Voc_Part	النداء	
Conj	العطف	الواو (و)
Sub	الربط	
Circum_Part	الحال	
Jurat_Part	القسم	
Accomp_Part	المعية	
Conj	العطف	

Sub	الربط	
RC_Part	الجزاء	
Caus_Part	السببية	
Prep	حرف جر	الباء (ب)
Prep	حرف جر	الكاف (ك)
Prep	حرف جر	هـ ح
Emph_Part	التوكيد	
RC_Part	جواب الشرط	
Fut_Part	الاستقبال	السين (س)
Def_Art	التعريف	ال

قائمة المراجع

- [1] Booij, G. (2007). The Grammar of Words: An Introduction to Linguistic Morphology. Second Edition. Oxford University Press.
- [2] Boudelaa, S & Monslen-Wilsson, W. (2001). Morphological Units in the Mental Arabic Lexicon. Cognition 81, 65 – 92.
- [3] Crystal, David. (2008). A Dictionary of Linguistics and Phonetics. Sixth Edition. Blackwell Publishing Ltd.
- [4] Dichy, J. & Farghaly, A. (2007). Grammar-Lexis Relation in The Computational Morphology of Arabic. In Souidi, A., Van Den Bosch, A. & Neumann, G. (eds.) (2007). Arabic Computational Morphology: Knowledge-based and Empirical Methods. Springer.
- [5] _____ (2003). Roots & Patterns vs. Stems plus Grammar-Lexis Specifications: on what basis should a multilingual lexical database centred on Arabic be built? MT Summit IX – Workshop: Machine Translation for Semitic Languages. New Orland, USA.
- [6] Farghaly, A. (1987). Three Level Morphology. Paper presented at the Arabic Morphology Workshop, Linguistic Summer Institute, Stanford, CA. In ACL-01 Workshop on Arabic Language Processing: Status and Prospects. Pp. 155 – 162. Toulouse, France.
- [7] Habash, Nizar. (2010). Introduction to Arabic Language Processing. Morgan & Claypool Publishers Series.
- [8] Habash, N & Rambow, O. (2006). MAGEAD: A Morphological Analyzer and Generator for Arabic Dialects. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meetings of the Association for Computational Linguistics. pp. 681 – 688. Association for Computational Linguistics, Australia: Sydne.
- [9] Lieber, R. (2009). Introducing Morphology. Cambridge: Cambridge University Press. Page.

- [10] SmrŽ, O. (2007). Functional Arabic Morphology: Formal System and Implementation. PhD Thesis, Charles University in Prague, Prague, Czech Republic.
- [11] Soudi, A., Cavalli-Sforza, V. & Jamari, A. (2001). A Computational Lexeme-Based Treatment of Arabic Morphology.

Implementing Speech recognition system on Android platform

Mostafa El-Hosiny^{*1}, Mostafa AbdEl-Raheem^{*2}, Mostafa Magdy^{*3}, Moataz Lasheen^{*4}, Motaz Mostafa^{*5}

** Electronics and Electrical Communication Engineering Dept., Cairo University. Giza, 12613, Egypt*

¹mostafa.abdallah.elhosiny@gmail.com

²mostafa.abdelreheem@gmail.com

³mostafa.m.montaser@gmail.com

⁴eng.moataz.lasheen@gmail.com

⁵motaz.halim@gmail.com

Abstract— This paper is mostly concerned about the development of an Android application for illiterate persons based on speech recognition system using Hidden Markov Model (HMM). Firstly, Hidden Markov Model tool kit (HTK) is used to implement the isolated word recognizer with phoneme based HMM models. Secondly, Pocket Sphinx system is used to handle a complete state-of-the-art HMM-based speech recognition system, designed at Carnegie Mellon University. SPHINX is one of the best and most versatile recognition systems around the world today speech recognition system. Finally, conversion to android stage and convert the engine of Speech recognition and speech verification to java and android codes, which are used to implement a user friendly mobile application. An HMM-based system, like all other speech recognition systems, functions by first learning the characteristics (or parameters) of a set of sound units, and then using what it has learned about the units to find the most probable sequence of sound units for a given speech signal. The process of learning about the sound units is called training. The process of using the knowledge acquired to deduce the most probable sequence of units in a given signal is called decoding, or simply recognition.

Keywords— HMM (Hidden Markov Model) – HTK (Hidden Markov Model Tool Kit) –PS(Pocket Sphinx) Speech Recognition - Voice Recognition - pattern recognition –API(application programming interface)Phoneme – ASR (Automatic Speech Recognition) - MFCCs (Mel Frequency Cepstral Coefficients) - Mean – Variance – Android – Java- NDK(Native Development kit).

1 INTRODUCTION

Arabic is the first language in the Arab world, i.e., Egypt, Saudi Arabia, Jordan, Oman, Yemen, Syria, Lebanon, etc. Arabic alphabets are used in several languages, such as Persian and Urdu. Standard Arabic has basically 34 phonemes, of which six are vowels, and 28 are consonants. Although Arabic is currently one of the most widely spoken languages in the world, there has been relatively little speech recognition research on Arabic compared to the other languages.

Speech is the primary means of communication between people. For reasons ranging from technological curiosity about the mechanisms for mechanical realization of human speech capabilities, to the desire to automate simple tasks inherently requiring human-machine interactions, research in automatic speech recognition (and speech synthesis) by machine has attracted a great deal of attention over the past five decades.

Automatic Speech Recognition (ASR) is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone. Although continuous speech recognition systems play a real role in speech recognition, isolated word recognition is still useful to find its applications.

The ultimate purpose of ASR technology is to allow 100% accuracy with all words that are intelligibly spoken by any person regardless of vocabulary size, background noise, or speaker variables. The last studies examine the reliability of automatic speech recognition (ASR) software used to teach pronunciation, focusing on one particular piece of software. Arabic speech recognition faces many challenges. For example, Arabic has short vowels, which are usually ignored in text. Therefore, more confusion will be added to the ASR decoder. Additionally, Arabic has many dialects where words are pronounced differently. The main problems in Arabic speech recognition are summarized, which include Arabic phonetics, discretization problem, grapheme-to-phoneme relation, and morphological complexity.

The illiteracy problem is one of the toughest problems facing Egypt. One in every four Egyptians is illiterate. Despite free education and long- running literacy programs, the number of illiterates has changed little in over two decades. Nearly 17 million adult Egyptians can neither read nor write, according to recent government data. Egypt's population of 85 million is growing at 1.76 percent a year. The strongest growth is among the rural poor.

The aim of the project is to serve illiterate people by dealing with them via user interface application. It's a mobile application in android platform, which helps the illiterate people to read the Arabic Letters, Arabic words and phrases by using an Arabic recognition system. Besides, android platform has the highest market share nowadays.

In section II, Hidden Markov Model concept is viewed.

In section III, speech recognition tools are shown. In section IV, the implemented algorithm of the Arabic speech recognition system is described. In section V, the whole experimental work is explained. In section VI, results are explained. In section VII,

conclusions are illustrated. At last, market and profit will be shown in section VIII, and then future work will be introduced in section IX.

2 HIDDEN MARKOV MODEL

In contemporary speech research community, Hidden Markov Model (HMM) is a dominant tool used to model a speech utterance. The utterance to be modelled may be a phone, a syllable or a word. In small vocabulary systems, the HMM tends to be used to model words. HMM are probabilistic models useful for modelling stochastic sequence with underlying finite state structure. Stochastic sequences in speech recognition are called observation sequences $O = o_1 o_2 \dots o_T$, where T is the length of the sequence. HMM with n states ($S_1, S_2 \dots S_n$) can be characterized by a set of parameters $\{\lambda = B, a, \pi\}$ where π is the initial distribution probability that describes the probability distribution of the observation symbol in the initial moment, and

$$\sum_{i=1}^n \pi_i = 1 \text{ and } \pi_i \geq 0$$

A is the transition probability matrix $\{a_{ij} | i, j = 1, 2, 3 \dots n\}$, a_{ij} is the probability of transition from state i to state j , and

$$\sum_{j=1}^n a_{ij} = 1 \text{ and } a_{ij} \geq 0.$$

B is the observation matrix $\{b_{ik} | i=1, 2, 3 \dots n, k=1, 2 \dots m\}$ where n is the number of the states and m is the number of observation symbols.

$$\sum_{k=1}^m b_{ik} = 1 ; b_{ik} \geq 0 ;$$

is the probability of observation symbol with index k emitted by the current state i . The main problems of HMM are: evaluation, decoding, and Learning problems.

A. Evaluation problem

Given the HMM $\{\lambda = B, a, \pi\}$ and the observation sequence $O = o_1 o_2 \dots o_T$, the probability that model λ has generated sequence O is calculated. Often this problem is solved by The Forward Backward Algorithm.

B. Decoding problem

Given the HMM $\{\lambda = B, a, \pi\}$ and the observation sequence $O = o_1 o_2 \dots o_T$, calculate the most likely sequence of hidden states that produced this observation sequence O . Usually this problem is handled by Viterbi Algorithm.

C. Learning problem

Given some training observation sequences $O = o_1 o_2 \dots o_T$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $\{\lambda = B, a, \pi\}$ that best fit training data. The most common solution for this problem is Baum-Welch algorithm which is considered the traditional method for training HMM.

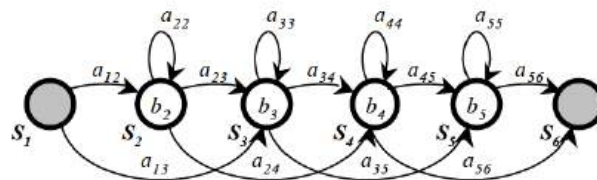


Figure 1: HMM with 4 active states

3 SPEECH RECOGNITION TOOLS

There are a number of Speech Recognition tools that are available to use and build a recognition system for any language. Each of these alternatives were compared against for factors like availability, performance, documentation, Supported platforms and languages. The comparison between the available tools is shown in Table1.

TABLE I
COMPARISON BETWEEN AVAILABLE SPEECH RECOGNITION TOOLS

	License	Development Language	Platforms	Support
Julius	BSD	C	Linux/Unix, Windows	-
HTK	Prohibits redistribution and commercial use but R&D allowed	C	Windows, Linux/Unix, MAC OS X	HTK book, Active Mailing List
Sphinx	BSD	C, JAVA	Linux/Unix, Windows, MAC OS X	Very well documented
ISIP ASR	Public domain (no restrictions)	C++	Windows	-

By considering the factors mentioned above and few others Sphinx is considered as the best tool for these reasons, it runs on any platform. It is very well documented; help is available from Source Forge group. Development language is C and Java both of which is very familiar. BSD license available, so it can be used for free Sphinx has a decoder called Pocket Sphinx that is specifically designed for cell phones and hand held devices and also runs in desktop.

4 ALGORITHM DESCRIPTION

Building the system was divided into three main categories, Firstly, Implementation of Hidden Markov Model on ten Arabic digits using HTK toolkit, Secondly, implementation of HMM using Pocket Sphinx on ten Arabic digits. Then, using existing model of Arabic phonemes, words and phrases built on Pocket Sphinx. After that, Conversion of the Pocket Sphinx on android is explained. Finally, application and user interface are illustrated. This is illustrated in Fig.2.

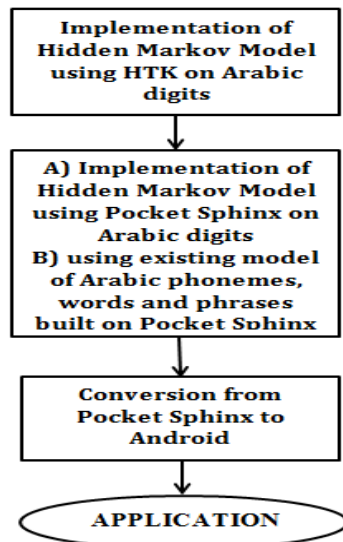


Figure 2: Flow chart of the system

A. Implementation of ASR using HTK

HTK is the most advanced and widely used system for modeling non-stationary data using HMM models. In this section, only the major blocks of the HTK are briefly presented in Fig 3. The HTK consists of three major blocks, feature extraction, classifier HMM for training and the recognition block.

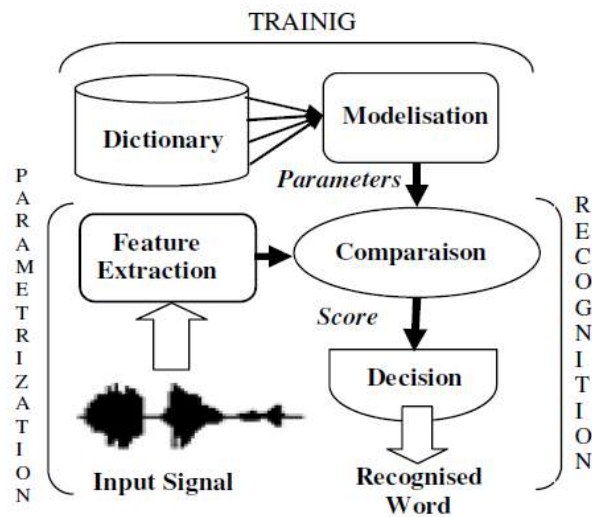


Figure 3: HTK processing phases

1) *Data preparation*: The feature extraction block supports various feature extraction techniques such as Linear Prediction Coefficients (LPC) [1], Reflection Coefficients (RC), Mel Frequency Cepstral Coefficients (MFCC) and more. This block can also estimate the dynamics of the features in time, i.e. the Derivative and Acceleration. So the first step uses the HTK command called HCopy.

2) *Training*: To initialize the parameters, HCompV was first used. The basic strategy implemented by HCompV is to make all models equal initially and move straight to embedded training. This is accomplished by equating the local mean and variance parameters of the Gaussians of each state to the global mean and variance. To complete the estimation of the HMM models, the HERest tool was used, which performs a single iteration of Baum-Welch re-estimation of the whole set of HMM models simultaneously. For each piece of training data, the corresponding models are concatenated, and the forward backward algorithm is used to collect state occupancy statistics, mean and variance statistics. When all the training data has been processed, the statistics are collected and the model parameters are re-estimated. HTK offers huge amount of tools for various auxiliary tasks like: HMM model management, grammar construction, language model construction, model adaptation, besides, a very important model editing tool HHed which performs basic and automatic editing of models like: splitting, merging, adding, etc. it also provides powerful functions for tying models (usually context dependent), where it is possible to tie any feature.

3) *Testing*: Once the models are trained, they are tested using the HTK command called HVite. User has to create a dictionary and the word network before executing HVite. HVite uses the Viterbi algorithm to test the models.

4) *Analysis*: Analyzing an HMM-based recognizer's performance is done by the tool HResults. It uses dynamic programming to align the transcriptions' output and correct reference transcriptions. HResults can also provide speakers-by-speaker breakdowns, confusion matrices and time-aligned transcriptions.

B. Implementation of ASR using Pocket Sphinx

Building model in pocket sphinx required two components: training components and decoder components.

1) *Sphinx trainer*: consists of a set of programs, each responsible for a well-defined task and a set of scripts that organizes the order in which the programs are called. The trainer learns the parameters of the models of the sound units using a set of sample speech signals. This is called a training database. A choice of training databases will also be provided. The trainer also needs to be told which sound units will be used to learn the parameters and at least the sequence in which they occur in every speech signal in the training database. This information is provided to the trainer through a file called the transcript file, in which the sequence of words and non-speech sounds are written exactly as they occurred in a speech signal, followed by a tag which can be used to associate this sequence with the corresponding speech signal. The trainer then looks into a dictionary

which maps every word to a sequence of sound units, to derive the sequence of sound units associated with each signal. Thus, in addition to the speech signals, A given set of transcripts will be for the database (in a single file) and two dictionaries, one in which legitimate words in the language are mapped sequences of sound units (or sub-word units), and another in which non-speech sounds are mapped to corresponding non-speech or speech-like sound units. These will be referred to the former as the language dictionary and the latter as the filler dictionary. In summary, the provided components for training will be: the trainer source code, the acoustic signals, the corresponding transcript file, a language dictionary and a filler dictionary.

2) *Sphinx decoder*: It consists of a set of programs, which have been compiled to give a single executable that will perform the recognition task, given the right inputs. The inputs that need to be given are: the trained acoustic models, a model index file, a language model, a language dictionary, a filler dictionary, and the set of acoustic signals that need to be recognized. The data to be recognized are commonly referred to as test data. In summary, the components provided for decoding will be: the decoder source code, the language dictionary, the filler dictionary, the language model, and the test data. In addition to these components, acoustic model is needed, which are trained for recognition. They are provided to the decoder. While training the acoustic models, the trainer will generate appropriately named model-index files. A model-index file simply contains numerical identifiers for each state of each HMM, which are used by the trainer and the decoder to access the correct sets of parameters for those HMM states. With any given set of acoustic models, the corresponding model-index file must be used for decoding. Finally, four types of models are generated and used in conversion to android stages, which are acoustic model, phonetic dictionary, language model and grammar model.

C. Conversion to Android

Acoustic models can be exported, and Pocket Sphinx can be integrated into handheld devices. Considering the availability of NDK (Native development Kit) of the Android OS for cell phones and the availability of documents on ways to integrate the jar files and the decoder into the android, decided that it will be a good option for developing the application. The Android NDK is a companion tool to the Android SDK that allows building performance critical portions of applications in native code (in this case JAVA will be used). A native development kit (NDK) is a software kit based on a native application programming interface (API) which allows computer software to be developed directly on a computing platform, rather than via a virtual machine. Creating software on a virtual machine is often easier than on a native development kit. However, the advantages of using an NDK is that it allows developers more options, and can yield higher performance. It also provides headers and libraries that allow building activities, handle user input, use hardware sensors, access application resources. Acoustic models will be generated beforehand, and the jar files copied to the cell phone. After integration with Pocket Sphinx, it can decode the commands. Accuracy rate is also not very high. The processing power of cell phones is also a concern. The developers of Sphinx have provided a sample demo project for Pocket Sphinx who is basically a rewrite of the recognition code in JAVA use with Android, which was previously written in C. The user had to build the necessary java files on their own using the swig interface of Pocket Sphinx. As shown in Fig 4.

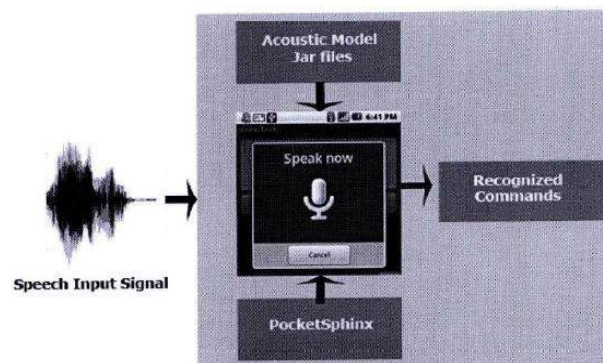


Figure4: Integrating pocket sphinx decoder using acoustic models

D. Application

After converting the engine of speech recognition and dealing easily with model files through Java's files, which are used in android platform. This is considered a user interface stage, and its output is an application which serves the illiterate people, so it characterized with help tracks that help user while using application, simple drawing, and direct choices, which enable user only one direction as not make user (illiterate person) confused with a lot of choices. Application is divided into two main

categories; training mode and test mode which is considered as a simulation of literacy class rooms' lessons, besides it contains some educational games, which add entertainment to the application's user.

5 EXPERIMENTAL WORK

A. Implementation of ASR for ten Arabic digits using HTK

The ASR system was built using Arabic digits for training and testing; the final accuracy and graphs will be shown in (Results) section. Building the system will be done by some steps. As an experiment, system was built on 200 training sound files and 100 test sound files of Arabic digits from 0 to 9, all sounds like files are sampled on 8000 kHz, 16 bits.

1) *Grammar*: HParse format grammar consists of an extended form of regular expression enclosed within parentheses. Expressions are constructed from sequences of words. It converts human-readable form of recognition network to human-unreadable HTK format.

2) *Dictionary*: It prepares pronunciation dictionary from one or more sources dictionary. The source dictionaries must be sorted in ASCII order, merge pronunciations from all source dictionaries, load the word list stored in output file, only pronunciations for words in this list will be extracted from source dictionaries, output a list of all phones encountered to file wrote to log file to the dictionary statistics and a list of a number of occurrences of each phone and construct the output dictionary which will be used in the upcoming steps in building the model.

3) *Transcription*: reading a list of editing commands from an edit script files and then make an edited copy of label files.

4) *Coding*: Copy one or more data files to a designated file, the configuration file which has a lot of important parameters like, SOURCEKIND: which is WAV file format, TARGETKIND: which is (mfcc) file, NUMCHANS, which represents, number of filter bank channels used in analysis, WINDOW SIZE: which represents window size time in units of 100ns, PREEMCOEF: which is pre-emphasis process, USEHAMMING: which represents using hamming window in windowing, SOURCERATE: which represents a sampling rate of input the wave and TARGERRATE: This represents period between each parameter vector as shown in Fig 5.

5) *Hmm initialization*: HMM training using Baum-Welch algorithm is an iterative process. Need an initial model, port an existing model from other domain/tasks (if available), Do a flat start (all models in the system have the same parameters). Determining the parameters for the initial model using the global mean as the Gaussian mean, using global variance as the Gaussian variance. This can be achieved using the HTK tool HCompV will scan a set of data files, compute the global mean and variance and set all the Gaussians in a given HMM to have the same mean and variance. Hence, assuming that a list of all the training files is stored in. The resulting initialised HMM description will be output where each vector is of length 39. This number, 39, is computed from the length of the parameterised static vector (MFCC 0 = 13) plus the delta coefficients (+13) plus the acceleration coefficients (+13).

6) *Re-estimation*: This causes the set of HMMs given in hmm0 (initial model folder) as to be loaded. The given list of training files is then used to perform one re-estimation cycle. As always, the list of training files can be stored in a script file if required. On completion, HERest outputs new updated versions of each HMM definition. If the number of training examples falls below a specified threshold for some particular HMM, then the new parameters for that HMM are ignored, and the original parameters are used instead.

7) *Decoding*: HVite is a general-purpose Viterbi word recognizer. It will match a speech file against a network of HMMs and output a transcription for each. When performing N-best recognition a word level lattice containing multiple hypotheses can also be produced. It is used to perform Viterbi decoding from a decoding network.

8) *Recognition*: HResults is the HTK performance analysis tool. It reads in a set of label files (typically output from a recognition tool such as HVite) and compares them with the corresponding reference transcription files. For the analysis of speech recognition output, the comparison is based on a Dynamic Programming-based string alignment procedure. The first line gives the sentence-level accuracy based on the total number of label files, which are the first line gives the sentence-level accuracy based on the total number of label files which are identical to the transcription files. The second line is the word accuracy based on the matching between the label files. In these transcription's second line, H is the number of correct labels, D

is the number of deletions, S is the number of substitutions, I is the number of insertions, and N is the total number of labels in the defining transcription files.

-Correct= $[(N-D-S)/N]*100\%$

-Accuracy= $[(N-D-S-I)/N]*100\%$

The accuracy is the more representative figure of recognizer performance.

```
Coding parameters

TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
NUMCEPS = 12
ENORMALISE = F
SOURCEFORMAT = WAV
```

Figure5: Configuration file for HTK

B. Implementation of HMM on 10 Arabic digits using PS

Building speech recognition system on pocket sphinx has a privilege than building it on HTK as model files can be loaded later on SD card and ability of importing the C-files of model to easeful java code and reach to application which can be installed on handled devices and PDA.

As any toolkit, Recognition system is divided into two categories, training path and decoding path (recognition).As an experiment, system was built on 200 training sound files and 100 test sound files of Arabic digits from 0 to 9, all sounds files are sampled on 8000 kHz, 16 bit.

For training data, firstly, it's converted to (.MFC) file to be in cepstra format by executing command in Fig 6 in command line, as for each training utterance, a sequence of 13-dimensional vectors (feature vectors) consisting of the Mel-frequency cepstral coefficients (MFCCs). List of paths of given wave files are in an4_test.fileids file. Since the data are all located in the same working directory, the paths are relative, not absolute. If the location of data is different, this step will take approximately 10 minutes to complete on a fast machine, but time may vary. The MFCCs will be placed automatically in a directory called (./feat). The type of features vectors which are computed from the speech signals for training and recognition is not restricted to MFCCs. Any reasonable parameterization technique can be used instead, any features of any type or dimensionality. The main reason for using MFCC is that they are currently known to result in the best recognition performance in HMM-based systems under most acoustic conditions.

Then by executing command in Fig 6, some processes are applied on (mfc) files and they take about few minutes on powerful machine.

```
perl scripts_pl/make_feats.pl -ctl etc/an4_train.fileids
```

Figure 6: Making Feature vector for training files

Process is going through the scripts in 00* through 09*, several sets of acoustic models will have been generated, each of which could be used for recognition. Some of the steps are required only for the creation of semi-continuous models, such as those used by Pocket Sphinx. If these steps are executed while creating continuous models, the scripts will benignly do nothing. Once the jobs launched from 02.ci_schmm have run to completion, the Context-Independent (CI) models for the sub-word units in dictionary will have been trained. When the jobs launched from the 04.cd_schmm_untied directory run to completion, having trained the models for Context-Dependent sub-word units (triphones) with united states. These are called CD-untied models and are necessary for building decision trees in order to tie states. The jobs in 05.buildtrees will build decision trees for each state of each sub-word unit. The jobs in 06.prunetree will prune the decision trees and tie the states. Following this, the jobs in 07.cd-schmm will train the final models for the triphones in the training corpus. These are called CD-tied models. The CD-tied models are trained in many stages. It's begin with 1 Gaussian per state HMMs, following which we train 2 Gaussian per state HMMs and so on till the desired number of Gaussians per State have been trained. The jobs in 07.cd-schmm will automatically train all these intermediate CD-tied models. Stages 08.deleted-interpolation and 09.make_s2_models are

meaningful only if models are trained for Pocket Sphinx. Deleted interpolation smooths the HMMs, which are then converted to the format used by Pocket Sphinx. Decoding stage may be proceeding even while the training is in progress, provided the certainty that the stage which generates the models wanted to decode with have been crossed. As training Arabic digits data on semi continues models, the final models will be located at ./model_parameters/an4.ci_semi models, where all files need to decode with Pocket Sphinx.

```
Perl scripts_pl/00.verify/verify_all.pl
Perl scripts_pl/01.vector_quantize/slave.VQ.pl
Perl scripts_pl/02.ci_schmm/slave_convg.pl
Perl scripts_pl/03.makeuntiedmdef/make_untied_mdef.pl
perlscripsts_pl/04.cd_schmm_untied/slave_convg.pl
perlscripsts_pl/05.builttrees/slave.treebuilder.pl
Perl scripts_pl/06.prunetree/slave.state-tie-er.pl
Perl scripts_pl/07.cd-schmm/slave_convg.pl
Perl scripts_pl/08.deleted
interpolation/deleted_interpolation.pl
perlscripsts_pl/09.make_s2_models/make_s2_models.pl
```

Figure 7: Command 2 for pocket sphinx

For test data, firstly, it's converted to (.MFC) files, by executing command in Fig 7. MFCC are computed for wave files.

```
perl scripts_pl/make_feats.pl -ctl etc/an4_test.fileids
```

Figure 8: Making Feature vector for test files

Accuracy is computed by executing command in Fig 8. This uses all of the provided components for decoding, including the generated acoustic models and model-index file from training run, to perform recognition on your test data. When the recognition job is complete, the script computes the recognition Word Error Rate (WER) or Sentence Error Rate (SER). When you run the decode script, It will also create two sets of files. One of these sets, with extension ".match", contains the hypothesis as output by the decoder. The other set, with extension ".align", contains the alignment generated by your alignment program, or by the built-in script, with the result of the comparison between the decoder hypothesis and the provided transcriptions.

```
perl scripts_pl/decode/slave.pl
```

Figure 9: Decoding the test files

There are control files that consists of dictionary file, filler file which models breathe and cough and phone file which contain list of phonemes. Also, there are configurations files for trainer and decoder which are tuned to get highest accuracy, such as number of states per HMM model, number of Gaussians per HMM, number of senones, Feature type, HMM model type and language weight.

C. Running pocket sphinx on android

Firstly, latest versions of Sphinx base and Pocket Sphinx were downloaded from the Sphinx group, then they are installed on operating system, android NDK is installed also.

After that NDK build is done, Eclipse is opened and the Pocket Sphinx Demo project is imported.. In the Builders screen SWIG and NDK build will be selected as builders for the project. SD card image has to be made manually in order to load the models which are the four types of models as mentioned before.

1) *Acoustic model*: It's used to model the sound of a phoneme. HMM is used, each phoneme has a model, It maps from HMMs to phones. For the purpose of this project the models on (an4.semi_ci) folder will be used. The folder contains the following files: (feat.params, mdef, meansmixture weights, noisedict, transition-matrices and variances. In addition to that file is required for pocketsphinx, a "sendump" file which basically consist the data from means and mixture-weights together.

2) *Phonetic dictionary*: Maps a list of words to their corresponding transcribed phonemes In CMU Sphinx, .dic files are dictionary files. For the purpose of this project the dictionary file is (arab2.dic).

3) *Language model*: Used to determine sequences of words are allowed in the language model.

4) *Grammar model*: Used to determine sequences of words are allowed. It is simpler than language model For the purpose of this project the language model file is (arabic2.gram) file.

After doing all the previous steps the recognition system, which is Voice-to-Text is running on emulator (or mobile-cell) as shown in Figure 10.

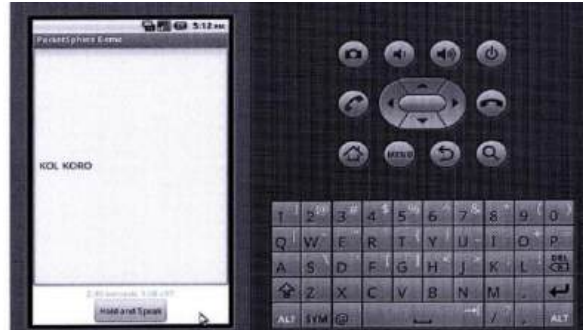


Figure 10: Emulator device in Eclipse

6 RESULTS

Results will be illustrated for each step separately.

A. Implementation of HMM on ten Arabic digits using HTK

We fused all the previously described steps and our final results achieved 95% accuracy. The accuracy depends on varying the penalty insertion as illustrated in Fig 11. The most important factor is changing the number of states representing each word as illustrated in Fig 12. This accuracy depends on the window length as shown in Fig.13 which improves the accuracy. The accuracy depends also on the training data and test data which make the model more reliable. Testing each parameter was done without changing the other two parameters. Confusion matrix is shown in Fig14. The Accuracy is illustrated in Fig15. The window size is in terms of 100ns. The confusion matrix is shown in Table 2. The accuracy is shown in Table 3, where Test data is the test sentences and all data is the same as Test data but each word is between two silences, so Test data measures the sentence accuracy and all data measures word accuracy.

1) Effect of Penalty of Accuracy

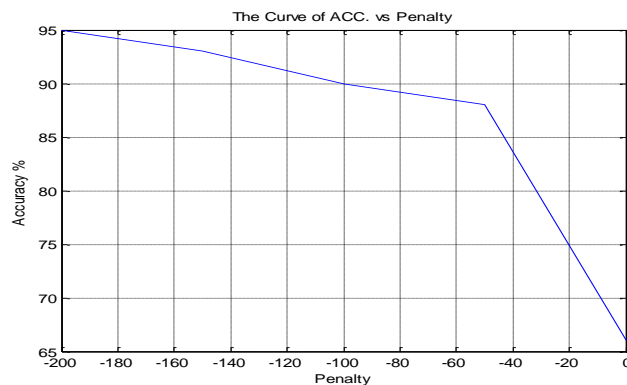


Figure11: Accuracy versus penalty insertion

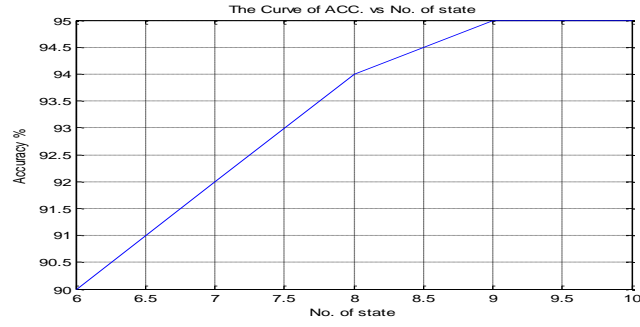


Figure 12: Accuracy versus number of states

2) Effect of window length on accuracy:

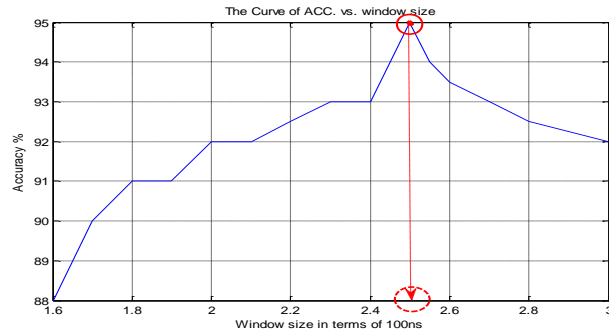


Figure 13: Effect of window length on Accuracy

3) Confusion Matrix Of The HMM Output

TABLE2
CONFUSION MATRIX

	صفر	واحد	اثنين	ثلاثة	اربعه	خمسه	سته	سبعه	ثمانيه	تسعه
صفر	9	0	0	0	0	0	0	0	0	0
واحد	0	9	0	0	0	0	0	1	0	0
اثنين	0	0	10	0	0	0	0	0	0	0
ثلاثة	0	0	0	10	0	0	0	0	0	0
اربعه	0	0	0	0	9	0	0	0	1	0
خمسه	0	0	0	0	0	10	0	0	0	0
سته	0	0	0	0	0	0	10	0	0	0
سبعه	0	0	0	0	0	0	0	10	0	0
ثمانيه	0	0	0	1	0	0	0	0	9	0
تسعه	0	0	0	0	0	0	0	0	0	10
ins	0	0	0	0	0	0	0	0	0	0

TABLE3
ACCURACY

Test sets	%correct	%Accuracy	H	D	S	I	N
Test	95%	-	95	-	5	-	100
All Data	98.66	97.99	294	1	3	2	298

B. Implementation of HMM on 10 Arabic digits using Pocket Sphinx

Firstly, wave files are recorded at mono, 16000 kHz sample rate and 16 bit resolution. Parameters tuned for training. HMM type: semi continues is used as it can be decoded on pocket sphinx decoder and then, it can be imported on handled device. Number of states: 5 states per HMM model. Number of senones: 1000. Feature type: 1s_c_d_dd which means the cepstra, the delta cepstra, and the double delta of the cepstra. Number of Gaussians starting with 8 up to 256 and it's noticed that 128 is the optimum value as shown in Fig 14. Parameter tuned for decoder is language weight: 10. The result accuracy (Sentence error rate is=96%).

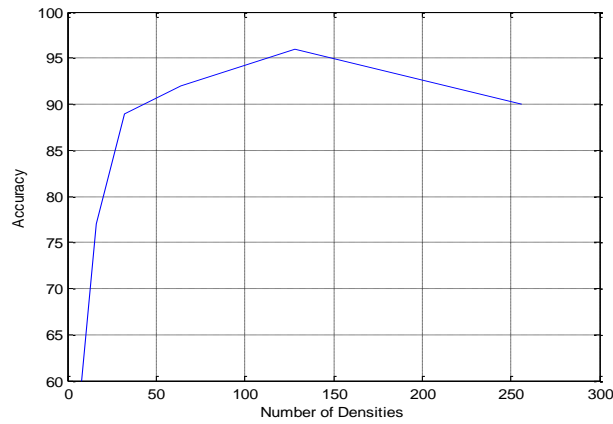


Figure14: Relation between Accuracy and Number of Densities

Then, wave files are changed to mono, 8000 bit sample rate and 16 bit resolution; the same parameters are tuned and produce accuracy (Sentence error rate= 84). The effect of changing Number of states is shown in Figure 15; also the effect of number of densities is shown as Figure 16.

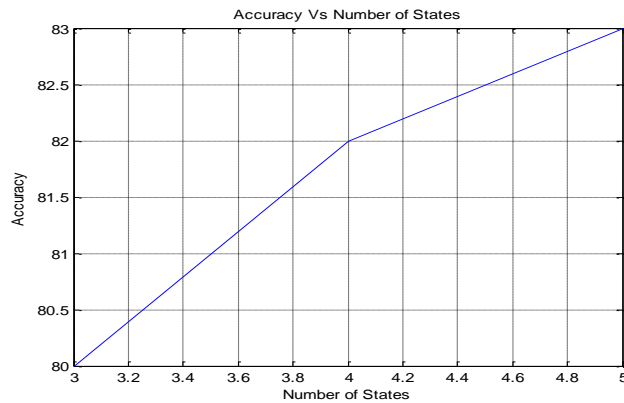


Figure 15: Relation between Accuracy and Number of States

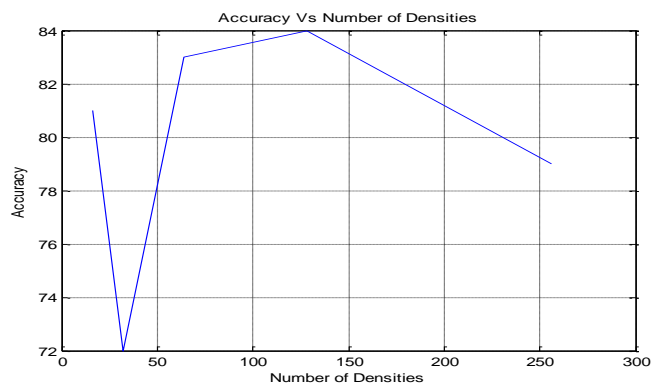


Figure 16: Relation between Accuracy and Number of Densities

C. Conversion to Android

After running pocket sphinx on terminal and getting the final accuracy for ten Arabic digits, a new model was used which is trained for 99 hours for Arabic phonemes, words and phrases. The final generated model files, phonetic dictionary, language model and grammar model are used. There were a lot of methods to increase the accuracy when running application on android like model adaptation, tuning, and formatting a strong Arabic dictionary for our application.

1) *Model adaptation*: It can be done by running `pocketsphinx_batch` and passing language model, dictionary, grammar and sample of test files for half hour.

2) *Tuning* : It was done by adjusting the values of word insertion penalty"-wip",the value of 0.0004 gives the best performance and language model weight"-lw" its effect was less than word insertion penalty and it was adjusted at 10

3) *Formatting strong dictionary*: it was very tedious task as the Arabic phonemes are larger than English phonemes and as the application was based on voice recognition for a lot of words in all test levels in the android application.

We did a lot of research on speech verification, in case of recognition, if recognizer misses the target word; it recognizes the nearest probability of the target word and produce another word from grammar as output, the target is either to recognize the target word or produce nothing, so garbage model is inserted parallel to recognition system, and parameter (wip) is used to give higher priority for the spoken word path, and if recognizer misses the target word, it's considered as a garbage (Phones loop not included in dictionary). So, it's required to generate grammar file for each word. But it's doesn't make sense as SD card is limited storage, we made java code which write new grammar file each time before verification. Unfortunately, it doesn't lead to good results.

7 CONCLUSION

Our goal was to achieve a complete system that will have a high accuracy and at the same time can be able to solve Literacy problem via simple user interface application; we have managed to develop a system and a guideline to on how an application can be developed for Android using existing tools, we collected some notes and results from a literacy's classroom. We are moving forward with it and continue working on the project to make a complete product with highest recognition accuracy.

8 MARKETING

There are three tracks for getting profits from this project. The first track is to be in the middle point between companies and organizations like (Resala, Maser El-khier ...etc.) and illiterate persons to support them by the "E-Phone" which will help them very much, briefly, to be as third party company. The second track is to sell it through TV advertisements, such as 0900 services. The third track is to participate in initiatives which are trying to solve the Illiteracy problem like Dr / Amr Khaled initiative and Vodafone initiative.

9 FUTURE WORK

Simply, the future work will be in two tracks, the first track is the speech recognition engine; we are trying to build a new model which has the most Arabic phrases, words and phonemes. And tune it to reach higher accuracy, solving the problem of verification as it's expected to get higher accuracy. The second track is to build a complete application which depends totally on speech recognition interface, add voice tag feature to the application.

REFERENCES

- [1] *Speech Recognition Theory and C++ Implementation*, Claudio Becchetti and Lucio Prina.
- [2] *Multiple Pronunciation Model for Amharic Speech Recognition System*.
- [3] *Building Arabic recognizer for Arabic digits using HTK*.
- [4] *Speech Enhancement with Application in Speech Recognition*, Xiao Xiong.
- [5] *What HMMs can do*, Jeff Bilmes.
- [6] *Investigation Arabic Speech recognition using CMU Sphinx system*, Arab academy for Banking.
- [7] *Basic HTK Tutorial*.
- [8] <http://htk.eng.cam.ac.uk/> (accessed July 2012)
- [9] <http://www.thedailyadmin.com/2008/03/powershell-tip-1-sorting-text.html> (accessed July 2012)
- [10] <http://www.myitforum.com/articles/40/view.asp?id=10541> (accessed July 2012)

- [11] <http://www.icsi.berkeley.edu/speech/docs/htkbook/node99> (accessed July 2012)
- [12] <http://www.joywang.info/?p=34> (accessed July 2012)
- [13] <http://www.voxforge.org/home/dev/acousticmodels/linux> (accessed June 2012)
- [14] <https://github.com/cesine/AndroidPocketSphinx> (accessed June 2012)
- [15] http://cmusphinx.sourceforge.net/sphinx4/#what_is_sphinx4 (accessed June 2012)
- [16] <http://swathiep.blogspot.com/2011/02/offline-speech-recognition-with.html>(accessed June 2012)
- [17] <http://www.speech.cs.cmu.edu/sphinx/tutorial.html> (accessed July 2012)
- [18] <http://cmusphinx.sourceforge.net/wiki/tutorialam>(accessed July 2012)
- [19]<http://www.android-tutorials.net/content/how-install-android- sdkubuntu-1110#3> (accessed July 2012)
- [20] [Implementation of Bangla Speech Recognition System on Cell Phones, Thesis Report ,Supervisor: Prof Dr Mumit Khan, Conducted by: K.M Tasbeer Ahsan](#) (accessed July 2012)
- [21]<http://cmusphinx.sourceforge.net/2011/05/building-pocketsphinx- on-android/> (accessed July 2012)
- [22]<http://rootzwiki.com/topic/20028-setting-up-an-android- development-environment-in-ubuntu-1110/> (accessed July 2012)
- [23] http://en.wikipedia.org/wiki/Java_Native_Interface (accessed July 2012)
- [24] <http://swathiep.blogspot.com/2011/02/offline-speech-recognition-with.html> (accessed July 2012)

An Overview of Web Intelligence

M. Adeb Ghonaimy*¹

* M. Adeb Ghonaimy, Faculty of Engineering
Ain Shams University, Egypt

¹adeebghonaimy@hotmail.com

Abstract— This paper gives an overview of Web intelligence which will enable the current Web to reach the Wisdom level by containing Distributed, Integrated, and Active knowledge. In this case it will be capable of performing tasks like problem solving and question-answering. In addition, it will be capable of processing and understanding natural languages. Web intelligence draws results from a number of disciplines like: Artificial intelligence, Information technology. Mathematics and Physics, Psychology and Linguistics. The paper covers the following topics: Web evolution and architecture- Topics related to Web intelligence-The Deep Web-Semantic computing and the Semantic Web-The Wisdom Web- Precisiated Natural Language.

1 INTRODUCTION

Web intelligence (WI) explores the impact of artificial intelligence and other advanced information technology concepts on the current Web [Zhong, 2002]. WI will allow better search procedures to return more relevant and precise information from the vast amount of knowledge distributed over the Web. The search space will constitute not only the small portion called the Surface Web, but will go beyond that to search what is called the Deep Web. Some of the main features of WI are:

The Web should be autonomic, capable of automatically delegating its functional roles to other agents. These agents should be capable of communicating with each other through an appropriate Agent Communication Language [Bradshaw, 1997]. The agent population will change dynamically as some may deactivate and others come in. The intelligent Web agents should be capable of using the Problem Solver Markup Language (PSML) or any other variant to specify their roles, settings, and relationship with other services. They should also be capable of processing and understanding natural language. It must understand and correctly judge the meaning (semantics) of concepts. Web agents must also incorporate a dynamically created source of metaknowledge that deals with the relationships between concepts and knowledge constraints. Finally, the intelligent Web can personalize user interactions. From the above, WI draws results from the following disciplines:

Artificial Intelligence, Information Technology, Mathematics and Statistics, Psychology, Linguistics, and Physics.

2 WEB EVOLUTION AND ARCHITECTURE

2-1 Web Evolution

Since its inception in 1990, the World Wide Web has been visualized as a distributed repository of knowledge. In order to search this vast knowledge base, search engines, like Google, were used for that purpose. However, it could search only a small portion of the Web, called the “Surface Web”. The “Deep Web” which contained a huge amount of knowledge remained unfathomed. Recently, some start-up companies used concepts from “quantum linguistics” to search the Deep Web. All this made use of a network architecture called “Client Server” model.

When peer-to-peer networks were introduced to allow social interaction, a new type of Web, called Web-2 was introduced and thus social networking became the next major Web application.

Returning to the original Web, sometimes called Web-1, the search made use of key words which did not in all cases result in the appropriate knowledge needed. Therefore, it was essential to resort to the Semantic Web, which necessitated the construction of Ontologies. These ontologies describe concepts in machine readable form and represent an essential component of the Semantic Web.

There are a number of Ontology Languages, like OWL (Ontology Web Language) and many other supporting tools to help in implementing the Semantic Web, sometime incorporated in what is called Web-3.

The next step in Web evolution is to make use of the distributed semantic knowledge in an “active” way to answer questions and solve problems. This requires the use of a number of “intelligent agents” each one specializing in a certain role and taking into account any constraints relevant to the specific problem. In other words, the Web should be equipped with inferencing capabilities that are performed in an autonomous manner. Those intelligent agents should be capable of using the Problem Solver Markup Language (PSML) to specify their roles, settings, and relationships with other services. Also, they should have the ability to process and understand natural languages. Web intelligence comprises many domains, among them: knowledge networks and management, Web agents, Web mining and farming, and distributed inferencing. Sometimes this conceptual, intelligent Web is called Web-4 or the Wisdom Web.

2-2 Web Architecture

We can visualize the Web as comprising a number of levels as follows [Zhong, 2002]:

- (1) Internet level : This is the infrastructure of the Web [Kleinrock, 2008] [Kleinrock, 2010]. It has evolved from the ARPANET in 1969 and is developing rapidly toward the New Generation Internet. The top 3 challenges of the Internet are: large-scale support for mobility, efficient content distribution, and security [Ortiz, 2008]. There are many research projects around the world to offer appropriate solutions for the above problems. Three of them are: Japan's New Generation Network, The U. S. GENI (Global Environment for Network Innovations), and the European Union's FIRE (Future Internet Research and Experimentation). Most of these projects have established a concept design, are expected to develop and verify the basic technology by 2015, and have the network running by 2020. Other developments are Quantum Networks for which there are a number of working prototypes around the world [Elliott, 2004] [Elliott, 2005] [Stix, 2005] [Curcic, 2005] and the Quantum Internet which is still a research activity in a number of academic institutions. [Lloyd, 2000], [Kimble, 2008].
- (2) Interface Level : Since the Web functions as an interface level for human-Internet interactions, it requires the following: Adaptive cross-language processing [Chung, 2009], personalized multimedia representation, and multimodal data processing capabilities [Oviatt, 2004] that can benefit from the recent interest in multidimensional translation [Gottlieb, 2005]. Web-enabled camera phones can enable queries in the form of an image captured by the phone camera [Stix, 2006].
- (3) Original Web Level: This is sometimes called Web-1, and it still represents a major level. An important application is searching the Web for pages satisfying a number of keywords. The means for doing that is through a number of search engines, e. g. Google. These engines return a list of the pages satisfying these keywords, but ranked in such a way that reflects the importance of each page. Here, it is assumed that pages are hyperlinked to each other. Google uses a PageRank algorithm to rank the different pages. This is, in some respects, similar to citation analysis of scientific papers [Ma, 2008]. Although billions of pages amount to this surface(or shallow) Web, there is a huge hidden Web from these search engines, that use only static HTML pages. This hidden Web (called Deep Web) contains scientific databases, library catalogs, and phone books [Wright, 2008], [He, 2007] [Goth, 2009] [Hondsuh, 2003]. A separate section will be devoted to the Deep Web.
- (4) The Social Web Level: This is sometimes called Web-2. Briefly, this is a Web in which people can contribute as much as they consume. The media coverage concentrated on blogs, video sharing and podcasting. Social networking websites let users build social connections with family, friends, and coworkers[Ko, 2010]. In the context of the social Web, user data is composed of three types of information: Identity data, social-graph data which represents who I know on the social Web, and Content data which represents what I have on the social Web [Ko,2010]. The education system can benefit tremendously from social Web services, where students, teachers, administrators, and parents will be more tightly connected.

The scientific community is also making a transition to Web -2 creating what some call Science2.0 [Waldrop, 2008]. An example of that is called OpenWetWare project at MIT where two biological engineering laboratories cooperated together as a wiki (a Web site that can be edited by anyone who has access) and uses the same software as the online encyclopedia Wikipedia [Website-1]. In 2007, the National Science foundation launched a five-year effort to transform this platform into a self-sustaining community independent of its current base at MIT. Other efforts are: the Nature Network which is an online network for scientists to discuss scientific news and events [Website-2], Science Commons which is an online project to aid open-access science on the Web [Website-3].

In the literature, there are some studies on Social Networks and Social Networking that deal with the design, development, and study of social technologies at the level of individuals, groups, and organizations [Churchill, 2005]. Also, with the development of technologies to locate individuals as they do their daily activities, a new class of location-aware information systems that link people-to-people-to geographical places (called P3 systems) is being created. Such systems can strengthen the relationship between social networks and physical places [Jones, 2005] .

The Web-2 technology has also been applied to model Patient-Centered health informatics application. This allows patients to participate in the health care system by sharing qualitative and quantitative information about their care plans, diagnosis, medications, and other relevant information [Weitzel, 2010].

- (5) The Semantic Web Level: This is sometimes called Web 3.0. In general, the Semantic Web extends the current Web so that information has a well-defined meaning. It has to provide a language that expresses both knowledge and rules for reasoning about knowledge. The Semantic Web consists mainly of three components: XML, RDF, and Ontologies.

The XML language is designed to make information self-describing. It is a metalanguage that enables exchange of information not only between different computer systems but also across national and cultural boundaries since it relies on the Unicode standard [Bosak, 1999]. In conjunction with Extensible Stylesheets Language (XSL), it is possible to reformat XML into different devices, thus achieving a “write once and publish everywhere”. XML allows users to add arbitrary structures to their documents but says nothing about their meanings.

Meaning is expressed by RDF Resource Description Framework which encodes it in a set of triples, each triple is like the subject, verb, and object of an elementary sentence, These triples can be written using XML tags, subjects and verbs are each identified by a Universal Resource Identifier (URI) [Berners-Lee, 2001].

For the Semantic Web to function, access to structured collections of knowledge, called Ontologies is essential. Ontologies are collections of statements written in a language such as RDF that define the relations between concepts and specify logical rules for reasoning about them. The Web Ontology Language (OWL) is thus the language that powers the Semantic Web.

Some vendors, like Oracle in 2005 offered RDF support in its spatial 10.2g database [Lassila, 2007].

There are also protocols for appropriate query languages as the SPARQL (The SPARQL Protocol and RDF Query Language).

It is possible to look at Web 3.0 as integrating Web 2.0 and the Semantic Web structure as shown in Figure 1 [Hendler, 2009] [Hendler, 2010].

Web 3.0	
Web 2.0	Semantic Web (RDF, OWL)
	Linked data (RDF, SPARQL)

Figure 1: Web 3.0 and its relation to web 2.0 and Semantic Web

- (6) The Wisdom Web Level: This is sometimes called Web 4. 0. At this stage we reach the level of Web Intelligence which was indicated briefly in the Introduction. In general, wisdom involves a holistic and integrative understanding of the world. Wisdom is not narrow or specialized knowledge but a broad and a deep knowledge [Lombardo, 2010]. A definition of the Wisdom Web that serves our purpose here is that it deals, with “Distributed, integrated, and active knowledge”. The different ontologies on the Web represents the first property as we have indicated in the previous section. However these Ontologies may be related to different specific domains. If we can integrate some of these ontologies that may cover our domain of interest, then the second property is satisfied. Now we come to the third and most important property, the “active knowledge”. In the context of the Wisdom Web, the distributed and integrated knowledge could be used to solve problems and answer questions. Since knowledge is always updated in a dynamic manner, the solutions and answers could always be updated to reflect the latest state of world knowledge. The ambitious goal could be achieved using techniques of Web Intelligence. If the Semantic Web and the Web 3.0 research achieve its promise and get stabilized and working by 2015 as some researchers predict, then the Wisdom Web can achieve its promise starting from 2020. It is possible to summarize what we have presented in section (2-2) in Figure 2

Before leaving this section, we describe some supporting Web services which are a framework of software technologies to support interoperable machine-to-machine interaction over a network [Leavitt, 2004]. Some of these are: SOAP (Simple Object Access Protocol) which transports a message between two points and can include extra information such as routing and the security mechanisms used.

WSDL (Web Services Description Languages) is an XML-based language that provides a description of the message, the protocols used (e. g. SOAP), and the address of the Web service.

UDDI (The Universal Description, Discovery, and Integration) specification that is used to quickly find Web services over the Internet. UDDI lists available Web services, gives their description, and provides instructions to use them.

Distributed, Integrated, and Active Knowledge
(Problem Solving, Question Answering)

XML relies on Unicode
RDF (Resource Description Framework)
Ontologies using OWL (Web Ontology Language)

People can contribute as much as they consume.
Education applications.
Science 2.0
P3 (People-to-people-to places)

Surface Web
Deep Web

Adaptive cross- language.
Personalized presentation.
Multimodal Processing.
Multidimensional translation.

New Generation Internet (GENI, FIRE, . . .)
Quantum Networks
Quantum Internet

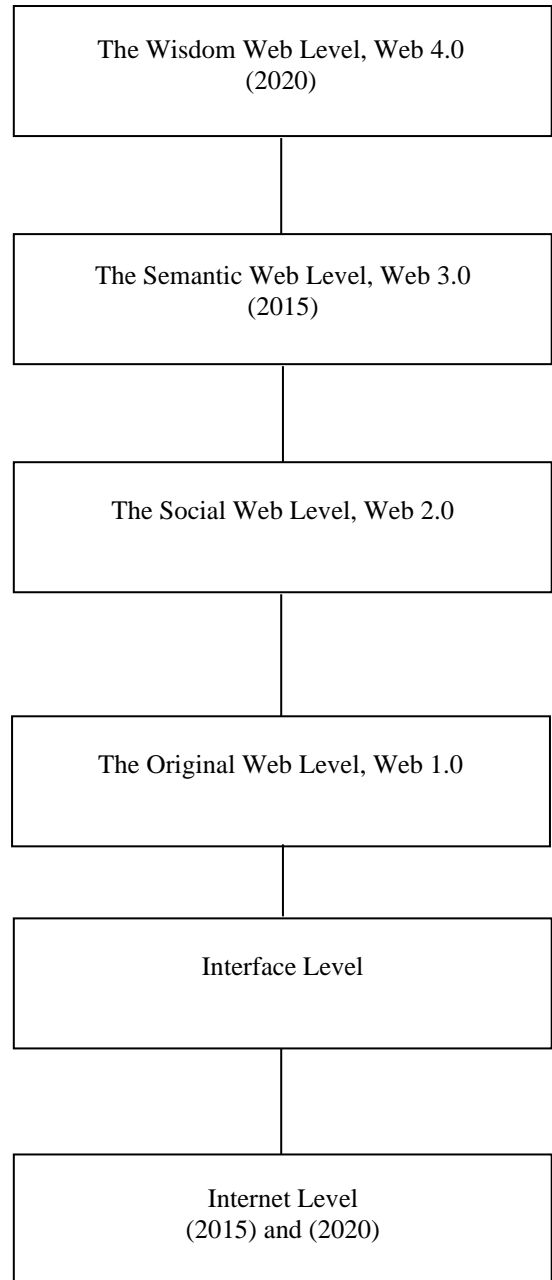


Figure 2: Web Architecture Levels

3 TOPICS RELATED TO WEB INTELLIGENCE

In this section a brief idea is given about some topics related to Web Intelligence [Yao, 2001][Zhong, 2002][Yao, 2004][Zhong, 2007].

- (1) Web Foundations include the following: Web information description and query languages – The Semantic Web – The Wisdom Web – Web protocols.
- (2) Web Human – Media Engineering includes : Multimedia information representation and processing-Visualization of Web information-Web based human computer interfacing.
- (3) Web Information Management includes: Multidimensional Web databases-Multimedia information management-Web knowledge management-Web security, integrity, privacy, and trust.
- (4) Web Information Retrieval includes: Image retrieval-Multilinguistic information retrieval-Multimedia retrieval-Ontology-based information retrieval.
- (5) Web Agents include: Dynamics of information sources-Global information collecting-Web-based cooperative problem solving.
- (6) Web Mining and Farming includes: Data mining and Knowledge discovery-Multimedia data mining-Web-based Ontology engineering-Web farming.
- (7) Web based Applications include: Business intelligence-Conversational systems-Electronic library-Web-based decision support systems-Web-based distributed information systems-Web-based learning systems.

One of the ambitious goals is to reach human-level intelligence [Zadeh, 2009]. Humans have many remarkable capabilities such as: the capability to reason, converse, and make rational decisions in an environment of imprecision, uncertainty, incompleteness of information, partiality of truth and possibility. A prerequisite to achievement of human-level intelligence is the mechanization of such capabilities and in particular natural language understanding. The latter necessitates dealing with the precisiation of meaning [Zadeh, 2004]. This will be considered in a separate section.

In order to develop the above applications in an appropriate manner, it was essential to develop the Web Engineering discipline [Ginige, 2001] which is related to Requirement and Software Engineering but adds the appropriate Web metrics. In addition, it has to take into consideration Security, Legal, Social, and Ethical issues [Deshpande, 2001].

4 THE DEEP WEB

The surface Web contains billions of static HTML pages. Two years ago, in 2008, Google had passed a milestone and added the one trillionth address to the list of Web pages it knows about. Beyond this surface Web lies an even vaster hidden Web called the Deep Web behind the forms of searchable databases. It has been estimated that this is 500 times larger than the surface Web [He, 2007]. These can include financial information, shopping catalogs, flight schedules, medical research and all kinds of other material stored in databases that remain invisible to search engines.

Deep Web searches sometimes use mediated search engines which relies on wrappers that serve as a kind of Rosetta Stone for each data source [Wright, 2008]. In order for the Deep Web to be properly searched, its semantic contents should be available for search [Goth, 2009] .

An example of a program that provides access to the hidden Web is Deep Query Manager from Bright Planet [Mostafa, 2005]. Google is also presenting a system for Surfacing the Deep Web content. It precomputes submissions for each HTML form and adding the resulting HTML pages into a search engine index [Madhavan, 2008].

A start-up Company called Dipsie has developed a number of technologies including the Dipsie. bot and algorithms based upon Quantum Linguistics to address problems related to searching the Deep Web. Its programs search a larger portion of the Web as well as analyze and interpret language in a more human-like manner to derive greater meaning. Based on Quantum Linguistics, its algorithms identify the utility of words and how they interact with, influence and are influenced by one another. Thus, they can predict the semantics of words and phrases within content and also recommend alternative content. Since semantics assumes an important role in Deep Web search, we present a brief account of word semantics and then a brief overview of quantum linguistics and its role in sentence and text semantics in general.

Word Semantics: Since meaningful sentences are composed of meaningful words, a logical starting point is to study the semantics of words and their relationships.

A tool that can help in this respect is WordNet [Miller, 1995][Fellbaum, 1998][Ghonaïmy, 2003]. It has been developed at Princeton University, and has been used together with other lexical operators to improve Web searches [Moldovan, 2000]. WordNet includes the following relations which are summarized in table (1)

- (1) *Synonymy* is WordNet's basic relation, because it uses sets of synonyms (synsets) to represent word senses. Synonymy (*syn* same, *onyma* name) is a symmetric relation between word forms.
- (2) *Antonymy* (opposing name) is also a symmetric semantic relation between word forms, especially important in organizing the meanings of adjectives and adverbs.
- (3) *Hyponymy* (sub-name) and its inverse, *hypernymy* (super-name) are transitive relations between synsets. Because there is usually only one hypernym, this semantic relation organizes the meaning of nouns in a hierarchical structure.
- (4) *Meronymy* (part-name) and its inverse, *holonymy* (whole-name), are complex semantic relations. WordNet distinguishes *component* parts, substantive parts, and member parts.
- (5) *Troponymy* (manner-name) is for verbs what hyponymy is for nouns, although the resulting hierarchies are much shallower.
- (6) *Entailment* relations between verbs are also coded in WordNet.

TABLE 1
SEMANTIC RELATIONS IN WORDNET

Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	<i>N, V, Aj, Av</i>	Pipe, tube Rise, ascend Sad, unhappy Rapidly, speedily
Antonymy (opposite)	<i>Aj, Av, (N, V)</i>	Wet, dry Powerful, powerless Friendly, unfriendly Rapidly, slowly
Hyponymy (subordinate)	<i>N</i>	Suger maple, maple Maple, tree Tree, plant
Meronymy (part)	<i>N</i>	Brim, hat Aluminum, airplane Ship, fleet
Troponymy (manner)	<i>V</i>	March, walk Whisper, speak
Entailment	<i>V</i>	Drive, ride Divorce, marry

Note : *N* = Nouns, *Aj* = Adjectives, *V* = Verbs, *Av* = Adverbs

With the development of Web 2.0, an increasing amount of user-generated content containing rich opinion and sentiment information has appeared on the Web. Texts containing opinions and emotions are referred to as direction-based texts. Sentiment classification can help determine whether a text contains positive or negative sentiments. SentiWordNet is a lexical resource for sentiment analysis [Dang, 2010], [Tufis, 2008] and is based on the original Princeton WordNet. The same concepts could be used for other languages.

A Brief Overview of Quantum Linguistics: Quantum linguistics is a recent activity that tries to apply quantum mechanics to language problems and in particular, semantics [Chen, 2002] in which he relied on the idea of "sign" presented by Ferdinand de Saussure. The sign is composed of "signifier" and "signified" illustrating the duality of symbol and concept. He suggested also a number of postulates for quantum linguistics.

Another domain in which quantum mechanics is applied is Information Retrieval. The pioneering work of Salton [Salton, 1984] suggested to represent a document by a vector of terms in the document space which can be considered as an ordinary Euclidean Vector space. Using concepts from quantum mechanics, a document can be represented as a vector in Hilbert space, and an observable such as "relevance" can be represented by a Hermitian operation. The important notions in quantum mechanics such as : state vector, observable, uncertainly, and superposition translate into analogous notions in information retrieval [Van Rijsbergen, 2004].

Semantic analysis is based on text co-occurrence matrices and data analysis techniques employing Singular Value Decomposition (SVD). Various models provide methods for determining similarity of meaning of words and passages by the analysis of large text corpora [Aerts, 2004]. In Latent Semantic Analysis one represents words by vectors spanning a finite dimensional space and text passages are represented by linear combinations of such words, with appropriate weights related to the frequency of occurrence of the words in the text. Similarity of meaning is represented by scalar products between certain word vectors.

However, LSA has some problems since it treats a text passage as bag of words in which order is irrelevant. This is a serious difficulty since the syntax is important for evaluating text meaning. As a simple example [Aerts, 2004], the sentences “Alice hits Bob” and “Bob hits Alice” cannot be distinguished by LSA.

If we resort to concepts from Quantum Information Theory (QIT) in which a basic object is not a word but a letter with the binary alphabet consisting of 0 and 1 qubits, then the ordering of qubits is obtained by means of the tensor product. Ordering of words can be obtained in the same way. The above reference gives some semantic analysis in quantum notation.

Another research direction is the development of mathematical frameworks that enable us to compute the meaning of a well-typed sentence from the meaning of its constituents. It depends on recasting the Hilbert space formalism in category-theoretic terms to admit a true logic for automating the methods used [Coecke, 2010][Clark, 2008]. It makes use of the categorical formalism for quantum mechanics introduced by Abramsky [Coecke, 2008][Coecke, 2005].

The last research direction referred to here is that which demonstrated the presence of quantum structures in language by proving the violation of Bell’s inequality [Aerts, 2004]. I am not going to give any further details here, but this research gives more evidence for the relation between quantum mechanics and natural language.

5 Semantic Computing and the Semantic Web

In Sept.. 2010, the IEEE Computer Society established a technical committee on Semantic Computing. Semantic Computing is in line with Web 3.0 which is characterized by the Semantic Web and the Internet of things, it also includes computing driven by natural language and all computational content such as software, devices, and processes. Semantic computing requires the development of new synergized technologies from natural language processing, data and knowledge engineering, software engineering, computer systems and networks, communication, signal processing, pattern recognition, and other technologies. A number of annual IEEE International Conferences on Semantic computing have already started in 2007. Also, an International Journal of Semantic Computing started to appear in the same year.

Semantic computing tries to match the semantics of computational content and the naturally expressed user intentions to help retrieve, manipulate, or even create content [Sheu, 2007][Sheu, 2010]. The connection between content and the user can be made via: semantic analysis, semantic integration, semantic services, service integration, and semantic interface. Figure 3 shows the general architecture of semantic computing adapted from [Sheu, 2010].

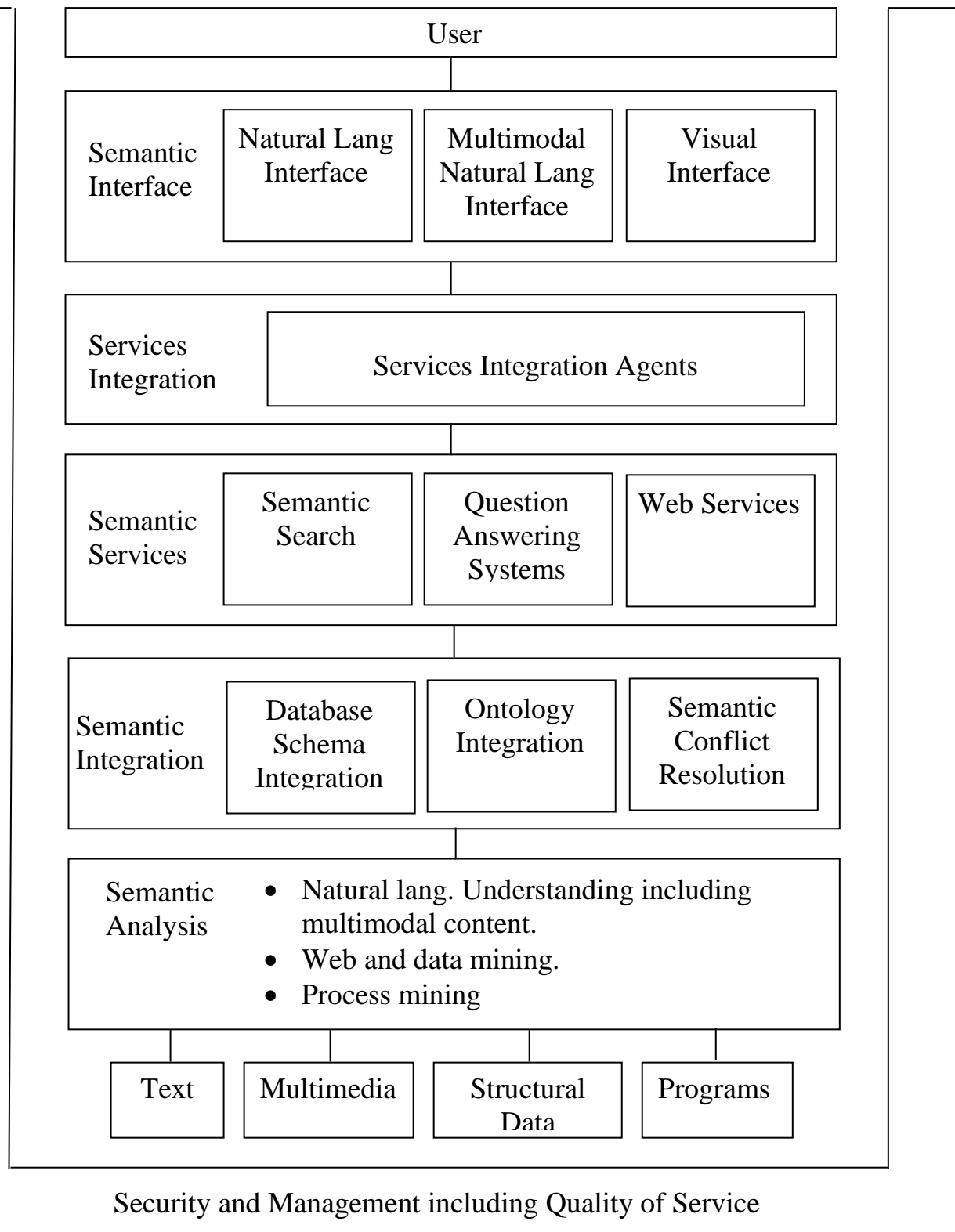


Figure 3: Architecture of Semantic Computing

Let us now turn to the Semantic Web and elaborate a little on the presentation given in section 2-2. Figure 4 gives a summary of the Semantic Web layers [Hendler, 2001].

Description Logics		
Ontology Web Language (OWL)		
Resource Description Framework (RDF)		RDF Schema
XML	Name Space	XML Schema
Unicode	Universal Resource Identifier (URI)	

Figure 4: Layers of the Semantic Web

A brief exposition is given to terms that were not considered before.

The XML-namespace syntax is used to abbreviate URIs in statements. The XML Schema standard forms a broad base on which developers can build interoperable XML applications. Therefore, XML Schema enables the cross-organizational sharing and verification of documents. The Schema specification consists of two parts: a language to describe the high-level structure of the XML document, and a list of allowable data types that can be used in those documents.

The basic building block in RDF is an object-attribute-value triple, written as $A(O, V)$ that is an object O has an attribute A with value V . Some tutorials are given in [Decker, 2000][Ghonaimy, 2003]. [Klasing, 2001]. RDF Schema lets developers define a particular vocabulary for RDF data and specify the kinds of objects to which these attributes can be applied. RDF Schema expressions are also valid RDF expressions (just as XML Schema expressions are valid XML).

In 2004, W3C announced OWL as a standard Web ontology language. It is based on Description Logics which are a family of logic-based knowledge representation formalisms that are descendents of Semantic Networks and other knowledge representation languages, but that have a formal semantics based on first-order logic. OWL was also based on earlier languages like OIL (Ontology Interchange Language) [Horrocks, 2008][Fensel, 2001]. The syntax of description logics consists of:

A set of unary predicate Symbols that are used to denote concept names.

A set of binary relations that are used to denote role names.

A recursive definition for defining concept terms from concept names and role names using constructors.

Description logics do not make the Unique Name Assumption (UNA) or the Closed World Assumption (CWA). Therefore, they employ the Open World Assumption (OWA). Under OWA, failure to derive a fact does not imply the opposite. This is closely related to the monotonic nature of first order logic which means that adding new information never falsifies a previous conclusion.

The framework for first order logic may be unsuitable for certain situations which require complete knowledge about the world. In this case inference will be non-monotonic, meaning that additional knowledge can invalidate previous conclusions [Genesereth, 1987][Schwartz, 1995]. Many knowledge modeling constructs are related to CWA and cannot be expressed in first-order logic. Default rules [Reiter, 1980] and constraints depend on CWA. Since many applications require OWA and CWA in parallel, some researchers suggest using Local Closed World (LCW) reasoning. They suggest using Autoepistemic Description Logics (ADL) to achieve that purpose [Grimm, 2006]. Autoepistemic logic is a formalism concerned with the notion of “knowledge” and “assumption”, i. e. to ask what the knowledge base knows or assumes [Levesque, 1990].

6 THE WISDOM WEB

Before I give some more information about the Wisdom Web, the following two quotations are given:

*Where is the life we have lost in living?
Where is the Wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?*
T. S. Eliot, the Rock, 1934

To know that we know what we know, and to know that we do not know what we do not know, that is true knowledge”
Nicolas Copernicus
(1473 – 1543)

The following definition for wisdom taken from [Lombardo, 2010] is given:

Wisdom is the continually evolving understanding of and fascination with the big picture of life and what is important, ethical, and meaningful. It includes the desire and ability to apply this understanding to enhance the well-being of life, both for oneself and for others.

Many thinkers and philosophers dreamt about that, like H. G. Wells (1866-1946) in his World Brain idea given in his book published in 1938 where he proposed the comprehensive organization of all knowledge [Rossman, 1993]. Also, Teilhard de Chardin (1887-1955) proposed similar ideas [Pelton, 1999].

From the above, it is essential to consider wisdom as combining knowledge with practical applications. So, knowledge is not an end in itself but reasoning about it is useful in understanding protocols in distributed systems since messages can be viewed as changing the state of knowledge of a system. This is important in cryptography theory, database and knowledge base theories [Halpern, 1992].

Therefore, a semantic model for knowledge is needed. This is essential in answering the following questions:

Do we know what facts we know?

Do we know what we don't know?

Do we know only true things, or can something we “know” actually be false?

Sometimes, a possible-worlds semantics is used to model knowledge [Halpern, 1992]. The idea behind that is that an agent's state of knowledge corresponds to the extent to which he can determine which world he is in. In a given world, we can associate with each agent the set of worlds that, according to the agent's knowledge, could possibly be the real world. Therefore, an agent knows a fact ϕ exactly if ϕ is true in all the worlds in this set, he does not know ϕ if there is at least one world that he considers possible where ϕ does not hold.

The more common axioms that characterize knowledge are summarized in the following [Halpern, 1992]:

(1) The knowledge axiom

$$Ki \phi \Rightarrow \phi, I = 1, \dots, n$$

Which states that only true facts can be known (this is the essential property, that distinguishes knowledge from belief). Ki means that agent i knows ϕ .

(2) Positive introspection axiom

$$Ki \phi \Rightarrow Ki Ki \phi, i = 1, \dots, n$$

Which states that an agent knows what facts he knows.

(3) Negative introspection axiom

$$Ki \phi \Rightarrow Ki \neg Ki \phi, I = i, \dots, n$$

Which states that an agent knows what facts he does not know.

(4) Inconsistent facts

$\neg Ki$ (*false*)

Which states that the agent does not know inconsistent facts.

In the following, a brief definition will be given for two terms: common knowledge and distributed knowledge.

Common knowledge: In Some situations it is needed to reason about the state of knowledge of a group of agents, i. e. we want to reason about facts that every one in the group knows. At other times we want to add that not only does everyone knows about them, but everyone knows that everyone knows them. These facts are said to be common knowledge.

Distributed Knowledge: It is also desirable to reason about the knowledge that is distributed in the group.

For example , if Alice knows ϕ and Bob knows $\phi \Rightarrow \psi$, then the knowledge of ψ is distributed among them, even though it might be the case that neither of them individually knows ψ . This is called distributed knowledge that corresponds to what a (fictitious) “wise person” would know. Distributed knowledge is a useful notion in describing the total knowledge available to a group of agents in a distributed environment [Halpern, 1992].

At this point it may be appropriate to give the following quotation taken from [Castells, 1996]:

*“Do you think me a learned, well-read man?”
“Certainly” replied Zi-gong. “Aren’t you?”
“Not at all” said Confucius. “I have simply grasped one thread which links up the rest”.*

In Section 2-2 (6), we defined the Wisdom Web as the layer that deals with Distributed, Integrated, and Active knowledge. So the Semantic Web with its ontologies should be augmented with extra components in order to reach the Wisdom Web level [Heflin, 2003]. The first step is to integrate the different distributed ontologies or part of them into wisdom knowledge in the sense that was just explained [Beneventano, 2003][McGuinness, 2009][Poole, 2009].

In this respect, the concept of a knowledge lens may be useful which can synthesize convergent, legitimate perspectives of the desired knowledge while suppressing the irrelevant [Edgington, 2004].

To take care of the active component of the wisdom knowledge, we indicated before that a step in that direction is to develop Web services. Some suggestions to develop coordinated agent-based services was given by [Sycara, 2004]. To give such services a form of autonomy was considered in [Paolucci, 2003]. Some work was also developed to mining actionable knowledge on the Web as in [Yang, 2004]. Distributed problem solving and other activities that need the cooperation of multiple agents were also treated in the literature [Drashansky, 1999] [Guha, 1994]. Human-level intelligence is also a research topic that is currently receiving a considerable attention [Beal, 2009].

7 PRECISIATED NATURAL LANGUAGE

A prerequisite to having the capability to reason and make rational decisions in an environment of imprecision, uncertainty, and incompleteness of information is to mechanize natural language understanding [Zadeh, 2009]. Concepts of Precisiated Natural Language (PNL) [Zadeh, 2004] and the process of language precisiation is a new approach towards semantic interpretation of natural language [Thint, 2007]. It is based on a generalized theory of uncertainty, and language percisiation is a process of adding constraints in order to clearly describe a complicated or ambiguous concept in natural language.

In PNL, the meaning of a proposition, P , in a natural language may be represented as a generalized constraint of the form:

$X \text{ is } r R$

Figure 5 illustrates the concept of a generalized constraint. In this figure, X is the constrained variable, R is the constraining relation which, in general, is not crisp (bivalent), and r is an indexing variable whose values define the modality of the constraint.

Standard constraint: $X \varepsilon C$ Generalized constraint: $X \text{ is } r R$ $X = (X_1, \dots, X_n)$
--

<p> X may be a structure : $X = \text{Location (Residence (Alice))}$ X may be a function of another variable : $X = f(Y)$ X may be conditioned : $X = (X/Y)$ r: $= / \leq \dots / < / \supset / \text{blank} / v / p / u / rs / fg / ps / \dots$ </p>

Figure 5: Generalized Constraint

The principal modalities given by r are:

Possibilistic ($r = \text{blank}$)

Veristic ($r = v$).

Probabilistic ($r = p$).

Random set ($r = rs$).

Fuzzy graph ($r = fg$).

Usually ($r = u$).

Pawlak set ($r = Ps$).

The set of all generalized constraints together with their combinations and qualifications constitutes the Generalized Constraint Language (GCL). [Zadeh, 2004]

A concept which plays a key role in PNL is that of a protoform (an abbreviation of “prototypical form”). Informally, the protoform of a lexical entity such as a proposition, command, question, or scenario is its abstracted summary. For example, the protoform (PF) of

P: Alice is young is $A(B)$ is C,

Where A is the abstraction of age, B is abstraction of Alice, and C is abstraction of young. Similarly, the protoform of

P: Most Swedes are tall is Count (B/A) is Q

Where A is abstraction of Swedes, B is abstraction of tall Swedes, count (B/A) is abstraction of the relative count of tall Swedes among Swedes, and Q is abstraction of most [Zadeh, 2005].

Importance of the protoform concept shows the deep semantic structure of the lexical entity to which it applies. So, propositions p and q are PF-equivalent written as PFE(p, q), if they have identical protoforms.

As a simple example, p: Most Swedes are tall, and

q: Few professors are rich, are PF-equivalent.

The above reference gives an application of this approach to question-answering systems.

8 CONCLUSIONS

Web intelligence explores the impact of artificial intelligence and other advanced information technology concepts on the current Web.

Topics related to Web intelligence include the following: Web foundations including the Semantic and the Wisdom Web-Web human-media engineering-Web information management including security, integrity, privacy, and trust-Web information retrieval-Web agents-Web mining and farming-Web based applications.

The Deep Web search engines are being developed to access information beyond the reach of current search engines. Semantics is a key issue in this respect making use of concepts from quantum linguistics. It extends Latent Semantic Analysis by some ideas from Quantum Information Theory.

The Semantic Web is briefly presented relating it to the new discipline of Semantic computing. Different layers of the Semantic Web are briefly described including the Web Ontology Language (OWL) and the Description Logics forming its theoretical foundation.

The Wisdom Web is also presented together with some basic material including: some knowledge axioms and the definition of the Distributed knowledge which is the foundation for Wisdom knowledge.

Finally, a very brief account for Precisiated Natural Language is presented which models natural language using a Generalized Constraint Language giving some very simple examples.

ACKNOWLEDGEMENT

This paper was presented in a seminar jointly organized by the Computers and Systems Engineering Dept. and the Information Systems Center, Faculty of Engineering, Ain Shams University, Cairo, Egypt. I appreciate very much the efforts of Professor Hosam Fahmy and Dr. Hassan Shehata in this respect.

REFERENCES

- [1] Aerts, D. and Czachor, M. "Quantum Aspects of Semantic Analysis and Symbolic Artificial Intelligence", *Journal of Physics A: Mathematical and General*, 37, No. 12 (26 March 2004), PP. 123-132.
- [2] Beal, J. and Winston, P. "The New Frontier of Human-level Artificial Intelligence", *IEEE Intelligent Systems*, July/August 2009, PP. 21-23.
- [3] Beneventano, D. et al. "Synthesizing an Integrated Ontology", *IEEE Internet Computing*, Sept./Oct. 2003 PP. 42-51.
- [4] Berners-Lee, T.; Hendler, J.; and Lassila, O. "The Semantic Web", *Scientific American*, May 2001, PP. 28-37.
- [5] Bosak, J. and Bray, T. "XML and the Second Generation Web", *Scientific American*, May 1999, PP. 79-83.
- [6] Bradshaw, J. M. (Ed). "Software Agents", MIT Press, 1997.
- [7] Castells, M. "The Rise of the Network Society", Blackwell Publishers, 1996.
- [8] Chen, J. "Quantum Computation and Natural Language Processing", Ph. D. Thesis, University of Hamburg, Germany, 2002.
- [9] Chung, W. "Web Searching in a Multilingual World". *Communications of ACM*, May 2009, PP. 32-40.
- [10] Churchill, E. F. and Halverson, C. A. "Social Networks and Social Networking", *IEEE Internet Computing*, Sept/Oct 2005, PP. 14-19.
- [11] Clark, S.; Coecke, B.; and Sadzadeh, M. "A Compositional Distributional Model of Meaning", *Proc. Conf. on Quantum Interactions*, University of Oxford, 2008, PP. 133-140.
- [12] Coecke, B. "Kindergarten Quantum Mechanics", *quantum-ph/0510032*, Oct. 2005, 18 pages.
- [13] Coecke, B. and Paquette, E. "Introducing Categories for the Practicing Physicist", *arXiv. Org/0808.1032*, 2008, 100 pages.
- [14] Coecke, B.; Sadzadeh, M.; and Clark, S. "Mathematical Foundations for a Compositional Distributional Model of Meaning", *Linguistic Analysis* 36, (Ed.) Lambeck et al. 2010, 34 pages.
- [15] Curcic, T. et al. "Quantum Networks: From Quantum Cryptography to Quantum Architecture", *ACM SIGCOMM Computer Communication Review*, Vol. 34, Number 5, October 2004, PP. 3-8.
- [16] Dang, Y.; Zhang, Y. and Chen H. "A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews", *IEEE Intelligent Systems*, July-August 2010, PP. 46-53.
- [17] Decker, S.; Mitra, P.; and Melnick, S. "Framework for the Semantic Web: An RDF tutorial", *IEEE Internet Computing*. Nov./Dec. 2000, PP. 68-73.
- [18] Deshpande, Y. and Hansen, S. "Web Engineering: Creating a Discipline Among Disciplines", *IEEE Multimedia*, April-June 2001, PP. 82-87.
- [19] Drashansky, T. et al. "Networked Agents for Scientific Computing", *Communications of the ACM*, March 1999, PP. 48-54.
- [20] Edgington et al "Adopting Ontology to Facilitate Knowledge Sharing", *Communications of the ACM*, Nov. 2004, PP.85-90.
- [21] Elliott, C. "Quantum Cryptography", *IEEE Security and Privacy*, July-August 2004, PP. 57-61.
- [22] Elliott, C. et al. "Current Status of the DARPA Quantum Network", *BBN Technologies*, March 2005, PP. 1-12.
- [23] Fellbaum, C. "WordNet", The MIT Press, 1998.
- [24] Fensel, D. et al. "OIL: An Ontology Infrastructure for the Semantic Web", *IEEE Intelligent Systems* March/April 2001, PP. 38-45.
- [25] Genesereth, M. and Nilsson, N. "Logical Foundations of Artificial Intelligence", Morgan Kaufmann, 1987.
- [26] Ghonaimy, M. Adeb "A Tutorial on WordNet", *Proc. The Fourth Conf. on Language Engineering*, Oct. 2003, Ain Shams Univ., Cairo, Egypt, PP. 1-28.
- [27] Ghonaimy, M. Adeb, "A Tutorial on the Semantic Web and Ontology Languages", *Proc. The Fourth Conf. on Language Engineering*, Oct. 003, Ain Shams Univ., Cairo, Egypt, PP. 29-68.
- [28] Ginige, A. and Muresau, S. "The Essence of Web Engineering", *IEEE Multimedia*, April-June 2001, PP. 22-25.
- [29] Goth, G. "Reaping Deep Web is a Matter of Semantics", *IEEE Internet Computing*, May/June 2009, PP. 7-10.
- [30] Gottlieb, H. "Multidimensional Translation: Semantics Turned Semiotics", *Proc. Of the Conference on Challenges of Multidimensional Translation*, 2005, PP. 1-29.
- [31] Grimm, S. and Motik, B. "Closed World Reasoning in the Semantic Web through Epistemic Operators", 2006.
- [32] Guha, R. and Lenat, D. "Enabling Agents to Work Together", *Communications of the ACM*, July 1994, PP. 127-142.

- [33] Halpern, J. and Moses, Y. "A Guide to Completeness and Complexity for Model Logics of Knowledge and Belief", *Artificial Intelligence* 54, April 1992, PP. 319-379.
- [34] Handschuh, S.; Volz, R.; and Staab, S. "Annotation for the Deep Web", *IEEE Intelligent Systems*, Sept/Oct. 2003, PP. 42-48.
- [35] He, B. et al. "Accessing the Deep Web", *Communications of the ACM*, May 2007, PP. 94-101.
- [36] Heflin, J. and Huhns, M. "The Zen of the Web", *IEEE Internet Computing*, Sept./Oct. 2003, PP. 30-33.
- [37] Hendler, J. "Agents and the Semantic Web", *IEEE Intelligent Systems*, March/April 2001, PP. 30-37.
- [38] Hendler, J. "Web 3.0: The Dawn of Semantic Search", *IEEE COMPUTER*, Jan. 2010, PP. 77-80.
- [39] Hendler, J. "Web 3.0 Emerging", *IEEE COMPUTER*, Jan. 2009, PP. 111-113.
- [40] Horrocks, I. "Ontologies and the Semantic Web", *Communications of the ACM*, Dec. 2008, PP. 58-67.
- [41] Jones, Q. and Grandhi, S. A. "P3 Systems: Putting the Place Back into Social Networks", *IEEE Internet Computing*, Sept/Oct 2005, PP. 38-46.
- [42] Kimble H. J. "The Quantum Internet", arXiv: quant-ph/0806. 4195 2008.
- [43] Klapsing, R.; Neumann, G. ; and Conen, W. "Semantics in Web Engineering: Applying the Resource Description Framework", *IEEE Multimedia*, April-June 2001, PP. 62-68.
- [44] Kleinrock, L. "An Early History of the Internet", *IEEE Communications*, August 2010, PP. 26-36.
- [45] Kleinrock, L. "History of the Internet and its Flexible Future", *IEEE Wireless Communication*, Feb. 2008, PP. 8-18.
- [46] Ko, M. N. et al "Social Networks Connect Services", *IEEE COMPUTER*, August 2010, PP. 37-43.
- [47] Lassila, O. and Hendler, J. "Embracing Web 3.0", *IEEE Internet Computing*, May-June 2007, PP. 90-93.
- [48] Leavitt, N. "Are Web Services Finally Ready to Deliver?", *IEEE COMPUTER*, Nov. 2004. PP. 14-18.
- [49] Levesque, H. "All I know: A Study in Autoepistemic Logic", *Artificial Intelligence*, Vol. 42, March 1990, PP. 263-309.
- [50] Lloyed, S. ; Shahriar, M. S.; and Hemmer, P. R. "Teleportation and the Quantum Internet", arXiv: quant-ph/0003147, 2000.
- [51] Lombardo, T. "Wisdom Facing Forward", *The Futurist*, Sept.-Oct. 2010, PP. 34-42.
- [52] Ma, N.; Guan, J.; and Zhao, Y. "Bringing PageRank to Citation Analysis", *Information Processing and Management*, 44 (2008), PP. 800-810.
- [53] Madhavan, J. et al. "Google's Deep-Web Crawl", *Very Large Data Bases VLDB08*, 24-30 August 2008, PP. 1241 – 1252.
- [54] McGuinness, D. et al. "The Emerging Field of Semantic Scientific Knowledge Integration", *IEEE Intelligent Systems*, Jan./Feb. 2009, PP. 25-26.
- [55] Miller, G. A. "WordNet" A Lexical Database for English", *Communications of ACM*, Nov. 1995, PP. 39-41.
- [56] Moldovan, D. and Milalcea "Using WordNet and Lexical Operators to Improve Internet Searches", *IEEE Internet Computing*, Jan-Feb. 2000, PP. 34-43.
- [57] Mostafa, J. "Seeking Better Web Searches", *Scientific American*, Feb. 2005, PP. 51-57.
- [58] Ortiz, S. "Internet Researchers Look to Wipe the Slate Clean" *IEEE COMPUTER*, Jan. 2008, PP.12-16.
- [59] Oviatt, S; Darrel, T.; and Flickner, M. "Multimodal Interfaces that Flex, Adapt, and Persist", *Communications of the ACM*, Jan. 2004, PP. 30-33.
- [60] Paolucci, M. and Sycara, K. "Autonomous Semantic Web Services", *IEEE Internet Computing*, Sept./Oct. 2003, PP. 34-41.
- [61] Pelton, J. "The Fast-Growing Global Brain", *The Futurist*, August/September 1999, PP. 24-27.
- [62] Poole, D.; Smyth, C.; and Sharma, R. "Ontology Design for Scientific Theories that Make Probabilistic Predictions", *IEEE Intelligent Systems*, Jan./Feb. 2009, PP. 27-36.
- [63] Rossman, P. "The Emerging WorldWide Electronic University", Praeger, 1993.
- [64] Salton, G. and McGill, M. "Introduction to Modern Information Retrieval", McGraw-Hill, 1984.
- [65] Schwarz, G. "In Search of a "true" Logic of Knowledge: The Nonmonotonic Perspective". *Artificial Intelligence*, Vol. 79, No. 1, Nov. 1995, PP. 39-63.
- [66] Sheu, P. "Editorial Preface", *Int. J. of Semantic Computing*, Vol. 1, No. 1, 2007, PP. 1-9.
- [67] Sheu, P. et al (Eds). "Semantic Computing", May 2010, Wiley-IEEE Press, Chapter 1.
- [68] Stix, G. "A Farewell to Keywords", *Scientific American*, July, 2006, PP. 73-75.
- [69] Stix, G. "Best-Kept Secrets", *Scientific American*, Jan. 2005, PP. 64-69.
- [70] Sycara, K et al. "Dynamic Discovery and Coordination of Agent-based Semantic Web Services", *IEEE Internet Computing*, May/June 2004, PP. 66-73.
- [71] Thint, M.; Beg, M.; and Qin, Z., "Precisiating Natural Language for a Question-Answering System", *11th World Multi Conference on Systems, Cybernetics, and Informatics*, 2007.
- [72] Tufis, D. "Playing with World Meanings", *From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence*, Eds: Zadeh, L. et al., 2008, PP.211-223.
- [73] Van Rijsbergen, C. J. "The Geometry of Information Retrieval", Cambridge University Press, 2004.
- [74] Waldrop, M. "Science 2.0", *Scientific American*, May 2008, PP. 46-51.
- [75] Weitzel, M. et al. "A Web 2.0 Model for Patient-Centered Health Informatics Applications", *IEEE COMPUTER*, July 2010, PP. 43-50.
- [76] Wright, A. "Searching the Deep Web", *Communications of the ACM*, Oct. 2008; PP. 14-15.
- [77] Yang, Q.; Knoblock, C.; and Wu, X. "Mining Actionable Knowledge on the Web", *IEEE Intelligent Systems*, Nov./Dec. 2004, PP. 30-31.
- [78] Yao, Y. Y. et al. "Web Intelligence (WI)". *Proc. Asia-Pacific Conference on Web Intelligence*, 2001.
- [79] Yao, Y.; Zhong, N.; and Liu, J. "Web Intelligence: Exploring Structures, Semantics, and Knowledge of the Web", *Knowledge-Based Systems*, 17(2004), PP. 175-177.
- [80] Zadeh, L. "From Search Engines to Question-Answering Systems-The Role of Fuzzy Logic.", *Progress in Informatics*, No. 1, (2005), PP. 1-3.
- [81] Zadeh, L. "Precisiating Natural Language (PNL)". *AI Magazine*, Vol. 25, Number 3, 2004, PP. 74-92.
- [82] Zadeh, L. "Toward Human Level Machine Intelligence-Is it Achievable? The Need for a Paradigm Shift", *Int. Journal of Advanced Intelligence*, Vol. 1, Number 1, Nov. 2009, PP. 1-26.
- [83] Zadeh, L. "Web Intelligence, World Knowledge and Fuzzy Logic", *BISC Program*, University of California, Berkeley. 2004.
- [84] Zhong, N.; Liu, J.; and Yao, Y. "Envisioning Intelligent Information Technologies through the Prism of Web Intelligence", *Communications of the ACM*, March 2007, PP. 89 – 94.
- [85] Zhong, N; Liu, J.; and Yao, Y. "In Search of the Wisdom Web", *IEEE COMPUTER*, Nov. 2002, PP.27-31.

10 – Web Sites (WS)

- 1- Website -1, (www.openwetware.org).
- 2- Website -2, (<http://network.nature.com>).
- 3- Website-3, (www.sciencecommons.org).

Automatic Speech Recognitions Using Wavelet Packet Increased Resolution Best Tree Encoding

Amr M. Gody^{*1}, Magdy Amer^{*2}, Maha M. Adham^{*3}, Eslam E. Elmaghraby^{*4}

**Department of Electrical Engineering,
Faculty of Engineering, Fayoum University, Egypt*

¹ amg00@fayoum.edu.eg

² moa02@fayoum.edu.eg

³ mma00@fayoum.edu.eg

⁴ eem00@fayoum.edu.eg

Abstract

The research is intended to introduce newly designed features for speech signal that can be used in Automatic Speech Recognition (ASR). The newly developed features are inherited from the wavelet packets decomposition of speech signal. The work is an enhancement of the original features developed by Amr M. Gody in [1]. Automatic Speech Recognition (ASR) of Arabic Phones without Grammar is considered as problem to be solved. Information related to speech phoneme is encoded into 15 bits instead of 7 bits in the original version of Best Tree Encoding (BTE4) in [1]. Best Tree Encoding of 5 levels of wavelet analysis (BTE5) gives 25% efficiency enhancement over the original BTE4[1] for solving the ASR problem. Whenever possible the comparison results are provided to explain the trend of enhancement in the Success Rate (SR). In Addition; BTE5 approaches 25% SR while Mel-Frequency-Cepstral-Coefficients (MFCC) approach 39% Success Rate of the ASR problem. This makes BTE5 approaches 71% of the SR of the popular and famous (MFCC) for the same ASR problem.

1 INTRODUCTION

Automatic speech recognition (ASR) has achieved some substantial successes in past few decades mostly attributing to two prevalent technologies in the field, namely hidden Markov modeling (HMM) of speech signals and efficient dynamic programming search (also known as decoding) techniques for very-large-scale networks[2].

Today, in many aspects, it has become a standard routine to build a state-of-the-art speech recognition system for any particular task if sufficient training data is provided for the target domain. However, migrating speech recognition systems from laboratory demonstrations to real world applications, even the best ASR systems available today still encounter some serious difficulties.

First of all, system performance usually dramatically degrades in the real fields because of ambient noises, speaker variations, channel distortions and many other mismatches. How to maintain and/or improve ASR performance in real-field conditions has been extensively studied in speech community under the topic of robust speech recognition.

Many good tutorial and overview papers, such as Juang (1991) [3], Gong (1995) [4] and Lee (1998) [5], can be easily found in the literature with regard to this topic. Secondly, since every speech recognizer inevitably will make some mistakes during recognition, outputs from any ASR system are always burdened with a variety of errors. Thus, in any real-world application, it is extremely important to make an appropriate and reliable judgment of speech recognition with taking into consideration the percentage of error resulted. This requires the ASR systems to automatically assess reliability or probability of correctness for every decision made by the systems.

The MFCC is one of the most famous features extraction methods in ASR fields. In the last decades, MFCC achieved good results in the field of phone recognition which is the topic of research in this paper. The average percentage of correction using this feature is 40% which improved to reach almost 70% by increasing vector size. The drawbacks of using MFCC is that its good results can't be obtained if the signal is noisy, it also needs a lot of calculations, and mainly used the assumption that the signal is stationary in a certain time [6]-[8]. All the MFCC improvements couldn't increase the results to overcome 70% for isolated phone recognition for English language [9].

The aim of this research is to introduce a new enhancement to the newly developed features by Amr M. Gody in [1]. Wavelet Packets Best Tree Encoded (BTE4) features base generation are discussed in [1]. BTE inherits some human attributes by considering the human hearing mechanism in processing the received speech. BTE depend on Wavelet packets which makes a similar processing on speech signal as the Filter banks method. It is much smarter than filter banks in that the number of filters is adapted by considering signal entropy to find the best tree. The key point in the proposed encoding system in [1] is that minimizing distance between feature vectors based on adjacency in frequency domain.

In this paper, wavelet packet Best Tree Encoding (BTE5) is introduced. It is an enhanced version of BTE4. BTE5 is designed to increase the resolution of information in the frequency domain.

In this paper, section 2 clarifies feature extraction using both BTE4 and BTE5. Section 3 describes the problem definition. Section 4 explains the experiment procedure and the results. Section 5 gives the conclusions.

2 PROBLEM DEFINITION

This research is a trial to improve the original Best Tree Encoding features by Amr M. Gody [1] BTE in phone recognition. Although the language is not the key of enhancement; the Arabic language is considered into the research. The key parameter is to increase the resolution of information that encodes the basic speech unit of speech recognition system; in our case the phone is considered as the basic speech unit.

3 BEST TREE ENCODING (BTE)

A. Best Tree Encoding BTE4



Figure 1: Block diagram of creating BTE

As showing in figure 1 the process of creating BTE4 file starts with framing the speech signal, then processing it by wavelet packet decomposition (WPD). The entropy is applied. “Shannon” entropy is chosen [10]. Wavelet family Daubechies filter with 4 points “DB4” is chosen [11]. Entropy is used to exclude leave nodes with low information. This step comes with a minimum leaves tree that maps the information into certain frequency bands encoded by indexing the tree leaves. The key point is to encode the indexing of the tree leaves in a way that ensures leaves of adjacent frequency bands have subsequent indexing values [1].

Figure 2 explains the 4 points encoding algorithm for BTE4. Wavelet analysis is a 4 level filtering process as shown in (2-a). Node 2 and node 3 are too far in frequency while they are subsequent nodes (according to Matlab wavelet packet decomposition function). Figure 2-b shows the nodes after rearranging according to BTE4. As shown in figure 2-b, recalling that the vertical axis is the frequency domain, the tree leaves are organized as the adjacent frequency leaves have almost subsequent indices. Feature vector is constructed into 4 components each is the binary encoding of the 7 supposed tree leaves of each cluster as shown in figure 2-b. Each supposed leaf node is encoded into certain bit of the feature component value number. If a leaf exists, then the corresponding bit = 1, else the bit = 0. For example; tree leaf with index 7 from figure 2-a is encoded into bit number 2 in the feature vector component as shown in figure 2-b. Each vector component covers 0.25 of the band width.

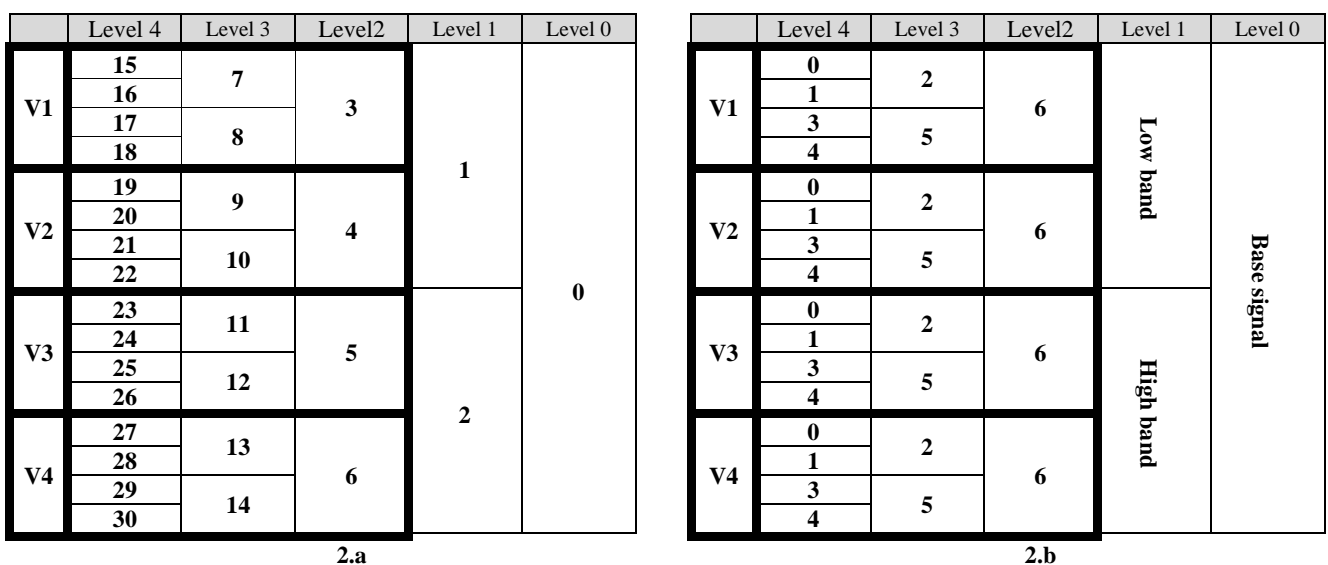


Figure 2: Clustering chart to explain the 4 points encoding algorithm, 2.a before arrangement, and 2.b after arrangement

4 INCREASED RESOLUTION BEST TREE ENCODING BTE5

Here the new generation of BTE features is introduced by adding a new level for analyzing the data in the wavelet packets decomposition. This will increase the resolution of information as shown in figure 3. The same encoding strategy as of BTE 4 is applied to encode the tree leaves. Figure 3.a explains the tree before arrangement; figure 3.b provides the encoding scheme rearranging tree leaves. Each vector's component maps 0.25 of the bandwidth. The bold area in figure explains the territory of each feature's vector component over the frequency bandwidth of the base signal. Adding the new analysis level makes that, the information will be encoded into 15 bits instead of 7 bits in BTE4. Increasing the resolution is supposed to give more information that can add more power in the discrimination process between the different underlying phonemes.

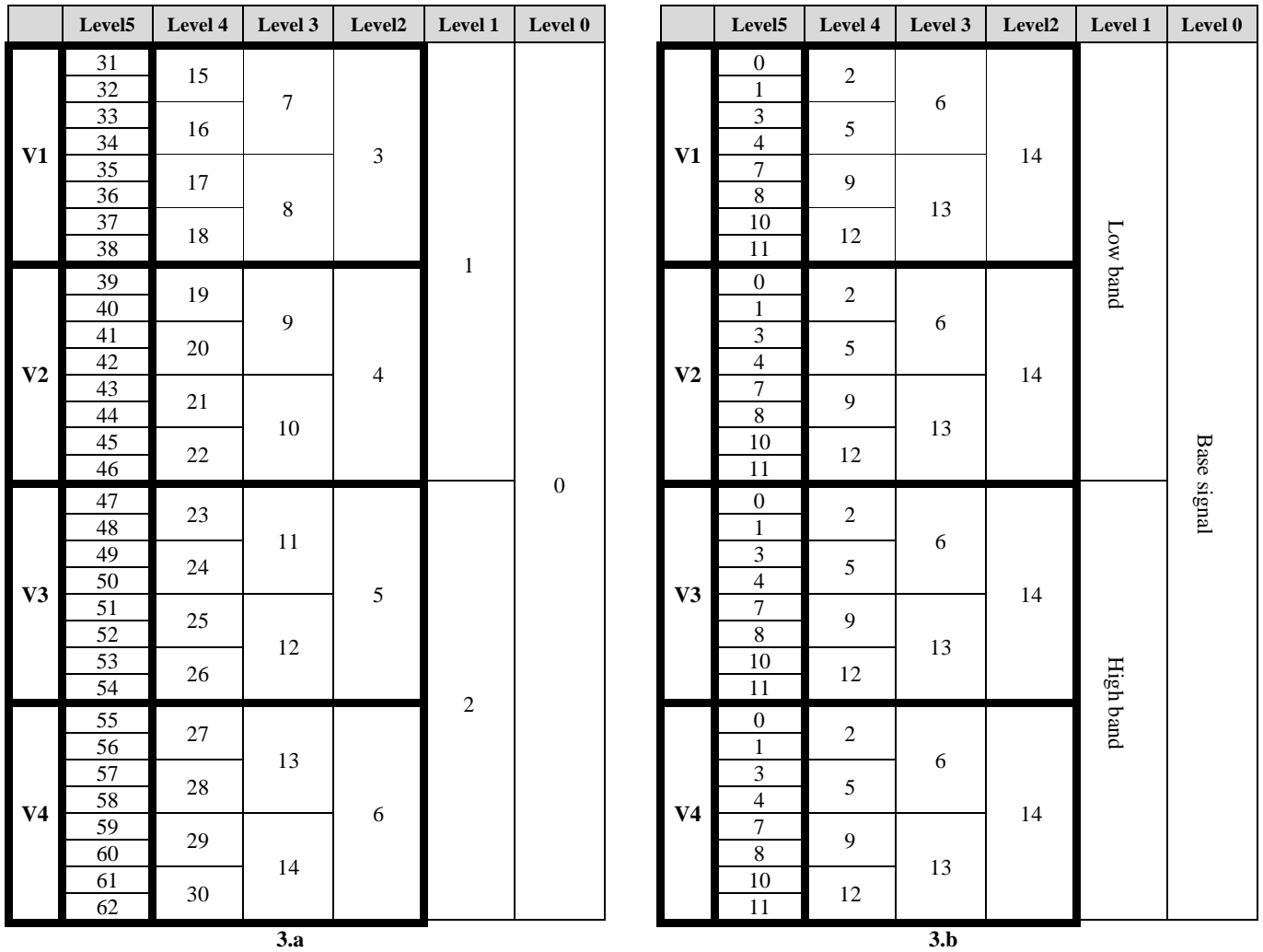


Figure 3: Clustering chart to explain the 5 levels encoding algorithm, 3.a before arrangement, and 3.b after arrangement

Figure 4 shows the MATLAB code to do the operation of the encoding of the tree leaves after adding the new level.

```

function [res]= btep1(frame)
    t = wpdec(frame,5,'db4','shannon'); u = leaves (t);  bt = besttree(t);    v = leaves (bt);
    res = box4encoder1(v);
end
function [v]= box4encoder1(a) % Encodes a best tree for wavelet packets in 4 levels.
    v = uint16([0;0;0;0]);    n = max(size(a));
    for i = 1 : n
        switch(a(i))

            case 3    v(1) = bitset(v(1),15);
            case 4    v(2) = bitset(v(2),15);
            case 5    v(3) = bitset(v(3),15);
            case 6    v(4) = bitset(v(4),15);
            case 7    v(1) = bitset(v(1), 7);
            case 8    v(1) = bitset(v(1),14);
            case 9    v(2) = bitset(v(2), 7);
            case 10   v(2) = bitset(v(2),14);
            case 11   v(3) = bitset(v(3), 7);
            case 12   v(3) = bitset(v(3),14);
            case 13   v(4) = bitset(v(4), 7);
            case 14   v(4) = bitset(v(4),14);
            case 15   v(1) = bitset(v(1), 3);
            case 16   v(1) = bitset(v(1), 6);
            case 17   v(1) = bitset(v(1),10);
            case 18   v(1) = bitset(v(1),13);
            case 19   v(2) = bitset(v(2), 3);
            case 20   v(2) = bitset(v(2), 6);
            case 21   v(2) = bitset(v(2),10);
            case 22   v(2) = bitset(v(2),13);
            case 23   v(3) = bitset(v(3), 3);
            case 24   v(3) = bitset(v(3), 6);
            case 25   v(3) = bitset(v(3),10);
            case 26   v(3) = bitset(v(3),13);
            case 27   v(4) = bitset(v(4), 3);
            case 28   v(4) = bitset(v(4), 6);
            case 29   v(4) = bitset(v(4),10);
            case 30   v(4) = bitset(v(4),13);
            case 31   v(1) = bitset(v(1), 1);
            case 32   v(1) = bitset(v(1), 2);
            case 33   v(1) = bitset(v(1), 4);
            case 34   v(1) = bitset(v(1), 5);

            case 35   v(1) = bitset(v(1), 8);
            case 36   v(1) = bitset(v(1), 9);
            case 37   v(1) = bitset(v(1),11);
            case 38   v(1) = bitset(v(1),12);
            case 39   v(2) = bitset(v(2), 1);
            case 40   v(2) = bitset(v(2), 2);
            case 41   v(2) = bitset(v(2), 4);
            case 42   v(2) = bitset(v(2), 5);
            case 43   v(2) = bitset(v(2), 8);
            case 44   v(2) = bitset(v(2), 9);
            case 45   v(2) = bitset(v(2),11);
            case 46   v(2) = bitset(v(2),12);
            case 47   v(3) = bitset(v(3), 1);
            case 48   v(3) = bitset(v(3), 2);
            case 49   v(3) = bitset(v(3), 4);
            case 50   v(3) = bitset(v(3), 5);
            case 51   v(3) = bitset(v(3), 8);
            case 52   v(3) = bitset(v(3), 9);
            case 53   v(3) = bitset(v(3),11);
            case 54   v(3) = bitset(v(3),12);
            case 55   v(4) = bitset(v(4), 1);
            case 56   v(4) = bitset(v(4), 2);
            case 57   v(4) = bitset(v(4), 4);
            case 58   v(4) = bitset(v(4), 5);
            case 59   v(4) = bitset(v(4), 8);
            case 60   v(4) = bitset(v(4), 9);
            case 61   v(4) = bitset(v(4),11);
            case 62   v(4) = bitset(v(4),12);

        end
    end
end

```

Figure 4: BTE5 MATLAB code

5 EXPERIMENTS

A. Terms and Procedures

1) *Features Extraction*: There are two different groups of features used in this experiment. SET_A will be designated for BTE4 while SET_B will be chosen for BTE5 features. Table 1 shows snapshot of sample from HMM training files.

TABLE 1
HMM TRAINING FILES

Group	Feature type	HTK feature files
SET_A	BTE4	0026_0050.htk
SET_B	BTE5	0026_0050.htk

Features can be enhanced by adding delta, acceleration and energy components to the features vector. Delta and acceleration of the feature component are used to add more dimensions to the feature vector in order to enhance recognition. The more the feature vector component the more information for the underlying phone will be added to the recognizer which in turn most likely will enhance the recognition results. Table 2 clarifies dimensions used in the experiment:

TABLE 2
DESCRIPTION OF VARIOUS FEATURE COMPONENTS

BTE4	Best Tree 4 Points Encoded (BTE).
BTE4_E	BTE4 plus the energy components ¹ .
BTE4_D_A	BTE4 plus the delta and acceleration components ² .
BTE4_E_D_A	BTE4 plus the delta, acceleration, and energy components ^{1,2} .
BTE5	Best Tree 5 level Encoded version 5 generation with five levels.
BTE5_E	Best Tree 5 level Encoded version 5 generation with five levels plus the energy component ¹ .
BTE5_D_A	Best Tree 5 level Encoded version 5 generation plus the delta and acceleration components ² .
BTE5_E_D_A	Best Tree 5 level Encoded version 5 generation plus the delta, acceleration, and energy components ^{1,2} .

¹ The energy is computed as the log of the signal energy within the analysis window.

$$E = \log \sum_{n=1}^N s_n^2$$

² The delta coefficients are computed using the following regression formula

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2}$$

where d_t is delta coefficient at time t computed in terms of the corresponding static coefficients $c_{t-\theta}$ to $c_{t+\theta}$. The value of Θ is set using the configuration parameter DELTAWINDOW. The same formula is applied to the delta coefficients to obtain acceleration coefficients except that in this case the window size is set by ACCWINDOW [12].

2) Database:

- Database is consisting of 30 speakers all of them are men, speaking different sentences (2977 files in wav format, 2652 files used for Training and 325 files used for Testing).
- Recording is mono; sampling rate is 32 kHz; sample size is 16 bits.
- The Transcription but not segmentation is available for all speech samples. The transcription is suitable for ASR of Arabic language in phone and word level recognition tasks.

3) Logic: Microsoft C# is selected to implement the recognition logic for file management and control the configuration of the Hidden Markov Model Toolkit tools (HTK tools). HTK is used as engine to execute the recognition tasks.

B. Result

HMM model with 3 emitting states and different Gaussian Mixtures in each state is used to model the recognition process. The following is the list of recognition process variables:

- 1- Feature name. This variable explains what feature is used in the recognition process. Recall table 2 form the details of all features used in this experiment.
- 2- Vector Size. This gives information about the size of feature vector which is corresponding to the selected feature.
- 3- Gaussian Mixture size. This is the number of Gaussian mixtures used in the emitting states of HMM. It is assumed that the same number in all emitting states in this research.

Table 3 gives the success rate against the experiment variables. The experiment variables are listed below

TABLE 3
EXPLAIN THE CORRECT PERCENTAGE USING BTE

Feature name	Qualifiers	Vector size	No. of mixtures	Success Rate (SR) %
BTE4	-	4	1	12.89
			2	16.75
			4	16.93
			8	20.05
	E	5	1	10.14
			2	16.55
			4	13.16
			8	12.58
	A, D	12	1	15.36
			2	19.90
			4	26.62
			8	23.80
	A, D, E	15	1	15.31
			2	17.80
			4	22.08
			8	21.78
BTE5	-	4	1	10.85
			2	14.44
			4	16.83
			8	16.08
	E	5	1	7.29
			2	12.50
			4	15.63
			8	16.67
	A, D	12	1	20.46
			2	23.42
			4	24.88
			8	23.29
	A, D, E	15	1	10.14
			2	16.60
			4	17.07
			8	18.37

Discussion of the results tabulated in table 3:

1) *Effect of Adding Mixtures*: As shown in figure.5, increasing numbers of mixtures enhances the results.

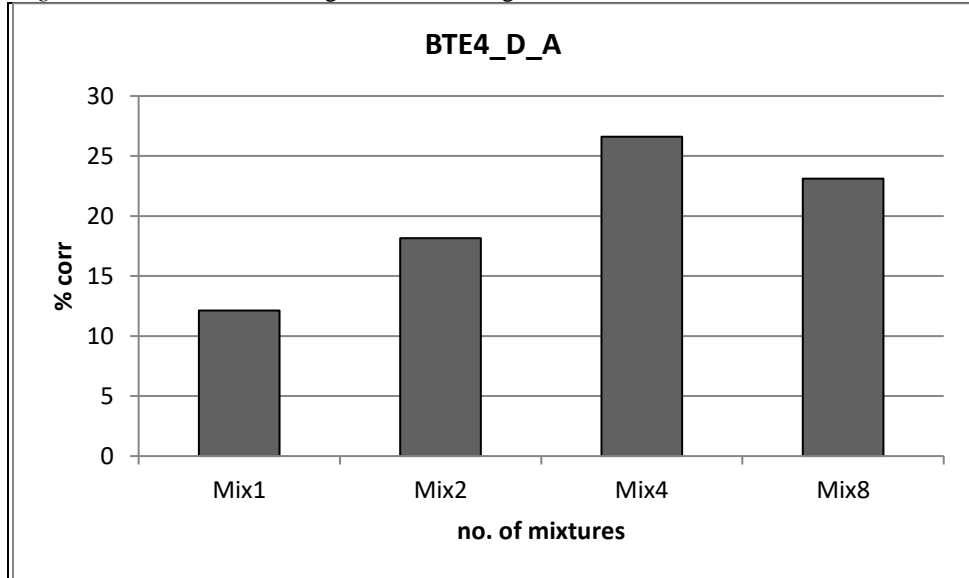


Figure 5: Effect of adding mixtures on the success rate.

The figure indicates that success rate increases as number of mixtures increases.

2) *Effect of Adding Energy, Delta, and Acceleration coefficients to Feature vector Components*: As shown in figure 6, the Success Rate (SR) increases as adding features components increases.

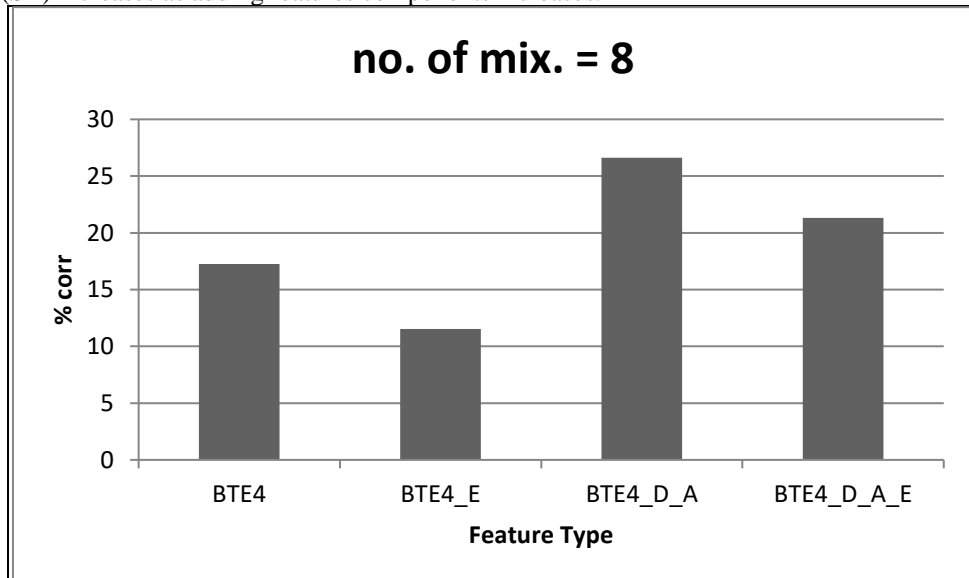


Figure 6: Effect of adding feature components on the success rate.

3) *Effect of Increasing the Number of training Iterations:* As shown in figure 7, increasing the number of training iterations enhances the results till saturation. The figure indicates that Success Rate (SR) increases as the number of iterations increases in a certain feature and certain number of mixtures.

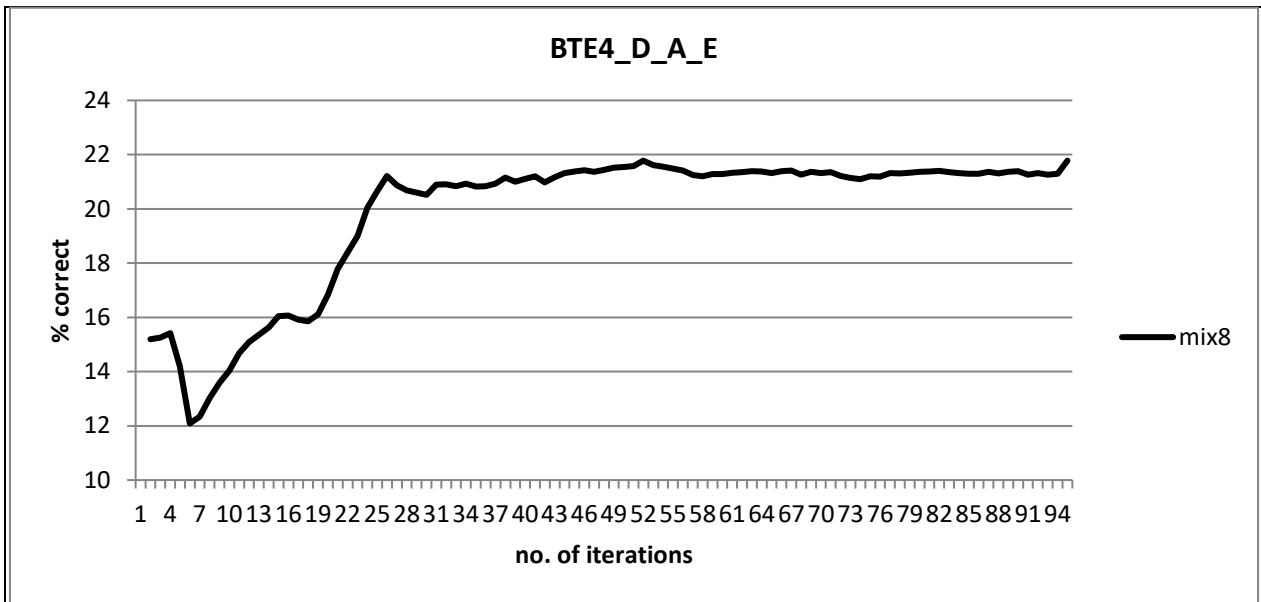


Figure 7: Effect of increase the number of iterations on the success rate.

4) *Comparison between BTE4 to BTE5:* As shown in figure8, the improvement of BTE5 over BTE4 enhances the SR. The feature components selected was E, D, and A, and the number of mixtures selected was 2. The figure indicates that the improvement in BTE5 over BTE4 makes a significant increase in SR.

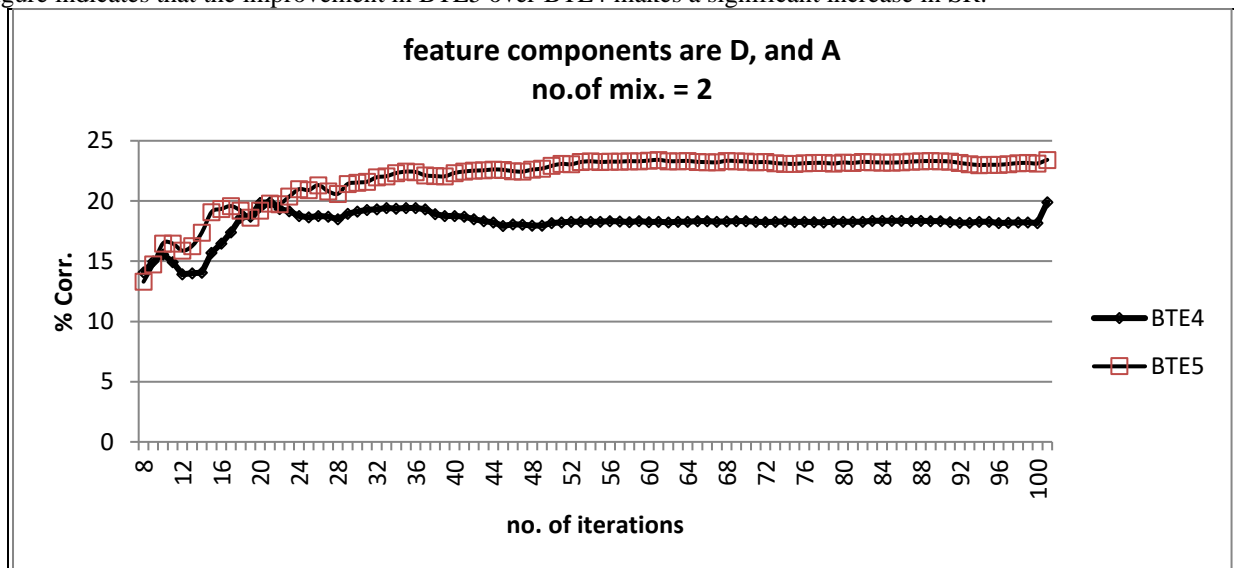


Figure 8: Improvement of BTE5 over BTE4.

6 CONCLUSION

BTE5 gives more success rate over BTE 4 for the same recognition process. Adding extra analysis level to the wavelet packet decomposition enhances information resolution. Information is stored into 15 bits instead of 7 bits of the original BTE4. The results become more optimistic to the Automatic Speech Recognition arena by approaching to 25% Success Rate compared to maximum of 20% of the original BTE4. This 5% difference in SR is $\frac{5}{20} \times 100 = 25\%$ efficiency enhancement to the original BTE4. In addition; BTE5 feature vectors with size 15, gives 25% success rate (SR). MFCC with size 13 gives 39% SR for the same problem. This is very promising that BTE5 is approaching 71% of the SR of the most popular feature used in the applied area of automatic speech recognition.

REFERENCES

- [1] Amr M. gody, "Wavelet Packets Best Tree 4 Points Encoded (BTE) Features", *The Eighth Conference on Language Engineering*, Ain-Shams University, Cairo, Egypt, pp. 17-18 December 2008.
- [2] Hui Jiang, "Confidence measures for speech recognition: A survey", *Speech Communication*, Vol. 45, no 4, pp. 455–470, 2005.
- [3] B. H. Juang, "Speech recognition in adverse environments", *Computer Speech and Language*, pp. 275–294, 1991.
- [4] Y. Gong, "Speech recognition in noisy environments: A survey", *Speech Communication* 16, pp. 261-291, 1995.
- [5] C. H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition", *Speech Communication*, Vol. 25, pp. 29–47, 1998.
- [6] Zabidi, W. Mansor and L. Y. Khuan, I. M. Yassin, R. Sahak, "Binary Particle Swarm Optimization and F-Ratio for selection of Features in the Recognition of Asphyxiated Infant Cry", *5th European IFMBE Conference*, IFMBE Proceedings, Vol. 37, pp. 61-65, 2011.
- [7] H. Jiang and M.J. Er, "Using Sub-Band Wavelet Packets Strategy for Feature Extraction", *Proceedings of 2nd WSEAS International Conference on Electronics*, Singapore, pp. 1-8, 7-9 December, 2003.
- [8] Vibha Tiwari, Dr. Jyoti Singhai, "Wavelet Based Noise Robust Features for Speaker Recognition", *Signal Processing: An International Journal (SPIJ)*, Vol. 5, Issue 2, pp. 52-64, 2011.
- [9] L.F. Lamel, J.L. Gauvain, "Experiments on Speaker-Independent Phone Recognition Using BREF", *ICASSP*, Vol. 1, pp. 557-560, 1992.
- [10] Inder Jeet Taneja, "Generalized Information Measures and Their Applications", *Universidade Federal de Santa Catarina*, Chapter1, 2001.
- [11] Michel Misiti, Yves Misiti, Georges Oppenheim, Jean-Michel Poggi, "Wavelet Toolbox for Use with MATLAB: User's Guide", *The MathWorks, Inc.*, Version 1, 1996.
- [12] University of Cambridge, "HTK Book", <http://htk.eng.cam.ac.uk/docs/docs.shtml>, Version 3.4, 2009.

Automatic Speech Recognition of Arabic Phones Using Optimal-Depth-Split-Energy Best Tree Encoding

Amr M. Gody^{*1}, Rania Ahmed Abul Seoud^{*2}, Eslam E. Elmaghraby^{*3}

**Electrical Engineering, Faculty of Engineering, Fayoum University Egypt*

1 amg00@fayoum.edu.eg

2 raa00@fayoum.edu.eg

3 eem00@fayoum.edu.eg

Abstract—Best Tree Encoding (BTE) which was first introduced in [1] gives promising results in Automatic Speech Recognition (ASR). The key factor in BTE is that solving Automatic Speech Recognition (ASR) problem in new domain for which the frequency information is mapped into 2D patterns. BTE4 and BTE5 suffer from many weaknesses that prevent them from being commercially suitable for ASR applications. The objective of this research is to enhance the efficiency of BTE in solving standard phone recognition problem, the Arabic language is used.

In this research, two factors are added to enhance BTE encoder. These factors are wavelet Analysis level (AL) and Count of features components into the features vector. Seven levels in wavelet decomposition analysis are considered as optimal analysis depth. In addition, extra 4 components are added to BTE features vector. The Energy feature component is split into 4 components. BTE7 gives 22% efficiency enhancement over BTE4 and about 45% efficiency enhancement over BTE5. In addition, BTE7 indicates more stability for recognition results over both BTE4 and BTE5. BTE7 gives more than 10% accuracy enhancements over both BTE4 and BTE5.

1 INTRODUCTION

Wavelets have come to play an important role in problems involving signal-processing [3]. Their advantages over the traditional Fourier Transform have been shown in variety of studies given their natural ability to analyze transitory components. Because there are an increasingly large number of wavelet families available, an important question is that of which one to choose for a particular application. A wavelet which works well for one application may perform poorly for another, even though the same signals are involved.

One way is to perform an exhaustive experimentation with different families of wavelets and their possible parameters. This is usually not very practical. Another way is to choose one's own favorite wavelet and stick with it, although there is no guarantee that it is the best for that particular application. Among the popular choices, for example, are those of Meyer, Daubechies and Splines [4].

Finally the introduction of Wavelet Packets [5] and algorithms try either to choose a “better base” for the chosen wavelet, or to avoid the decision of which wavelet to choose and consider a dictionary of different wavelets and let the algorithm pick the best functions available to make the required approximation.

Wavelets have come to play an important role in speech recognition researches. In [6] H. L. Rufiner and J. Goddard introduced a method for choosing between different wavelets and their corresponding parameters for phone recognition using TIMIT database for English language. In [7] M.A.Anusuya, and S.K.Katti deal with different speech processing techniques. It deals with study of the recognition accuracy against different kinds of wavelet transforms. It is shown that by using wavelet as frequency analysis in the pre-processing phase instead of conventional Fourier techniques in the conventional Speech Recognition methods, the signal recognition accuracy will be increased in both cases clean and noisy speech. Results presented in [7] show the advantage of pre-processing the signals with wavelet techniques which give good results over conventional methods. The test is done for word recognition of English language. In [8] H. Satori , M. Harti , and N. Chenfour built an Arabic Automated Speech Recognition System (ASR) by performing some experiments on different individuals (three men) each one of them was asked to utter 10 Arabic digits as the result for small size of the corpus of training and it gave results of 83%. Many of these papers either do not perform on large database or do not deal with Arabic language. Therefore, there appears to be significant room for further exploration of this area.

In this research, section 2 gives the key parameters of using wavelet as preprocessing frequency analyzer. The development history of BTE features are illustrated in section 3. The history and theory of BTE is summarized through section 3. Best tree encoding depends on the entropy to minimize the tree structure into the informative leaves. Proper encoding is used to encode the optimized tree structure into 4 16-bits numbers. Each number holds the tree structure of part of the signal bandwidth. The bandwidth is divided into 4 equal parts, each number represent a part of the 4 bandwidth quarters. Hidden Markov Model HMM is used for modeling the recognized entities. The proposed HMM model is illustrated in section 4. Gaussian Mixtures are used

to model the observations into each emitting state. Adaptive method is used to select the best number of Gaussian Mixtures to be used in emitting states. Hidden Markov Model Toolkit HTK by university of Cambridge [12] is utilized for working with HMM to ensure accuracy of the obtained results. The proposed enhancement to BTE is detailed in section 5. Discussion of chosen 7 levels of wavelet packet analysis is introduced in section 5. Energy component is split into 4 elements as of holding percentage energy distribution through one quarter of the bandwidth. Energy split gives more discrimination power for HMM to discriminate phones even they have same energy but different distribution. The experimental work is illustrated in section 6. Through this section, the database is illustrated and the training and test process are clarified. In section 7, the results are introduced through proper tables and figures. Finally the conclusions are provided in section 8.

2 KEY PARAMETERS IN WAVELET PREPROCESSING OF SPEECH SIGNAL

In this section the key parameters that affect the wavelet preprocessing of speech signal will be introduced.

A. Choice of Wavelet

Wavelet preprocessing analysis inherits some human attributes by considering the human hearing mechanism in processing the received speech. Wavelet packets consider very similar processing on speech signal as the Filter banks method. It is much smarter than filter banks in that the number of filters is adapted by considering signal entropy to find the best tree. The ideal choice of wavelet would be the one that best provides meaningful distinguishing features. In [9] Michelle Daniels illustrates some classification experiments. Three kinds of wavelet families are utilized for getting sense of which wavelet family gives better results for analyzing speech signal. The wavelet families used in Michelle Daniels experiments are Daubechies, Symlet and Coiflet wavelets. The results indicate that Daubechies family seemed to be better representative of speech signal than the other wavelet families. In this paper, Daubechies is chosen for wavelet packets as the base family.

B. Choice of Wavelet Decomposition Level:

Another parameter related to the choice of wavelet is the choice of how many levels of decomposition should be performed before extracting features. For n levels of decomposition, Wavelet Packets Decomposition (WPD) produces 2^n different sets of coefficients (or nodes) where n is number of levels. For example; 4 levels will produce a tree structure with 16 leaf nodes. Let us consider speech frame of 640 samples. At each level of wavelet decomposition, each wavelet filter will produce 640 samples. To overcome this growing in the output samples the down sampling is applied. At level 1 there will be two filters, each filter will produce $640/2=320$ (samples). Going along deeper analysis levels, shorter frame of output samples will be produced from each wavelet filter but the total number of samples will be kept at 640 samples. This limitation will lead to have maximum depth that we can assume for this signal in order to keep proper output frame size. The proper size is the minimum size that we can have information. At level 7 of wavelet decomposition there will be $2^7=128$ leaf nodes. This leads to output frame size of $640/128=5$ (samples). This is considered the minimum output frame size. Signal with less than 5 samples are hard to be analyzed to find entropy and to get any frequency information.

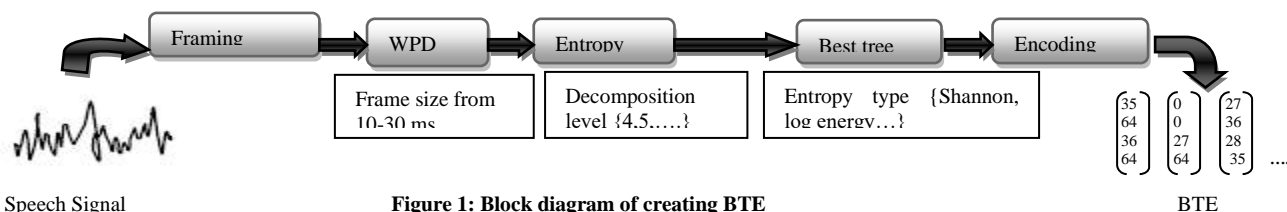
3 BEST TREE ENCODING (BTE)

Wavelet Packets Best Tree Encoding (BTE) was first introduced in [1]. The key point in BTE is moving the problem of speech recognition from the traditional frequency domain analysis to visual 2-Dimensional domain that provides representative tree-drawing figure that represents the frequency contents of the base signal. The tree structure is encoded into vector of 4 components. The encoding process is by nature normalizing the unlimited number of all possible tree structures into 4-Dimensional vector of 16-Bits values. BTE encoder is designed to produce codes that satisfy the following conditions:

- 1- Euclidian distance is used as observation measurer.
- 2- Adjacent tree Leafs have relatively Euclidian adjacent codes.
- 3- Full recover of all possible tree structures without overlapping codes (100% discriminative codes for all possible tree structures).

A. Best Tree Encoding (BTE) Block diagram and fundamentals:

As shown in figure 1 the process of creating BTE file starts with producing frames of the speech signal. The second step is the preprocessing phase. Wavelet packet decomposition (WPD) is used in the preprocessing phase as shown in figure by WPD Block. The next step is to select the proper entropy type. Then get the best tree that contains the significant signal power using the entropy from the previous step. The last step is to encode the tree structure into 4-Dimensional vector of integer values [1].



Here is a summary of each block's functionality from figure 1;

- 1) *Framing*: It is the process of segmenting the speech signal into small durations called frames in order to deal with it as stationary signal during those small durations as shown in figure 2. Frame length is most likely chosen from 10 to 30 ms in speech signals. Tri state Hidden Markov Model is chosen to model the frame of speech in the later on recognition process. This is by considering the frame contains three subsequent durations: The two outside transient durations which most likely the phone will have non-stationary properties and the middle duration is supposed to be the stable phone duration which contains stable properties. For sure there will be many possibilities of phone occupation along the time. The possible different occupations are illustrated by the colored tri parts bar in figure2. Colors are used to encode different phonemes.



Figure 2: Speech signal is segmented into sequence of features frames.

- 2) *Wavelet Packet Decomposition (WPD)*: It is a one-dimensional wavelet packet analysis function, which returns a wavelet packet tree t corresponding to the wavelet packet decomposition of the vector X at level for example 4, with a particular wavelet
 Ex: $t = \text{wpdec}(X, 4, 'db4', 'shannon');$
- 3) *Entropy*: Entropy provides a complexity measure of a time series, such as discretized speech signal. In Matlab; there are various types of entropy: Shannon, log energy, threshold, sure, norm, and user. In this paper Shannon entropy is chosen.

Shannon Entropy:

The Shannon entropy equation provides a way to estimate the average minimum number of bits needed to encode a string of symbols, based on the frequency of the symbols [10].

$$H(x) = - \sum_i s_i \log(s_i)$$

Where s_i is the probability of a given-sample.

Log energy Entropy:

The logarithm of "energy," defined as the sum over all samples:

$$H(x) = - \sum_i \log(S_i^2)$$

Where S_i is the sample value itself.

- 4) *Best tree*: Best tree M [11] function is a one- or two-dimensional wavelet packet analysis function that computes the optimal sub tree of an initial tree with respect to an entropy type criterion. The resulting tree may be much smaller than the initial one. Following the organization of the wavelet packets library, it is natural to count the decompositions issued from a given orthogonal wavelet. A signal of length $N = 2^L$ can be expanded in α different ways, where α is the number of binary sub trees of a complete binary tree of depth L where $\alpha \geq 2^{N/2}$. This number may be very large, and since explicit enumeration is generally intractable, it is interesting to find an optimal decomposition with respect to a convenient criterion, computable by an efficient algorithm. We are looking for a minimum of the criterion.
- 5) *Encoding*: The encoding process for BTE features vector is developed such as to minimize the distance based on frequency adjacency. Figure 3 explains the 4 points encoding algorithm for BTE4 that was introduced in [1] which has only 4 levels for the tree. As shown in figure 3.a that node 2 and node 3 are too far in frequency while they are subsequent nodes as wavelet packets indexing system. This problem needs to be altered such that adjacent frequency bands are listed as contiguous numbers. This way we will ensure that indexing system reflects frequency scale. Figure

3.b shows the nodes after rearranging. In figure 3 vectors are surrounded in bold boxes and as shown it is only four vectors

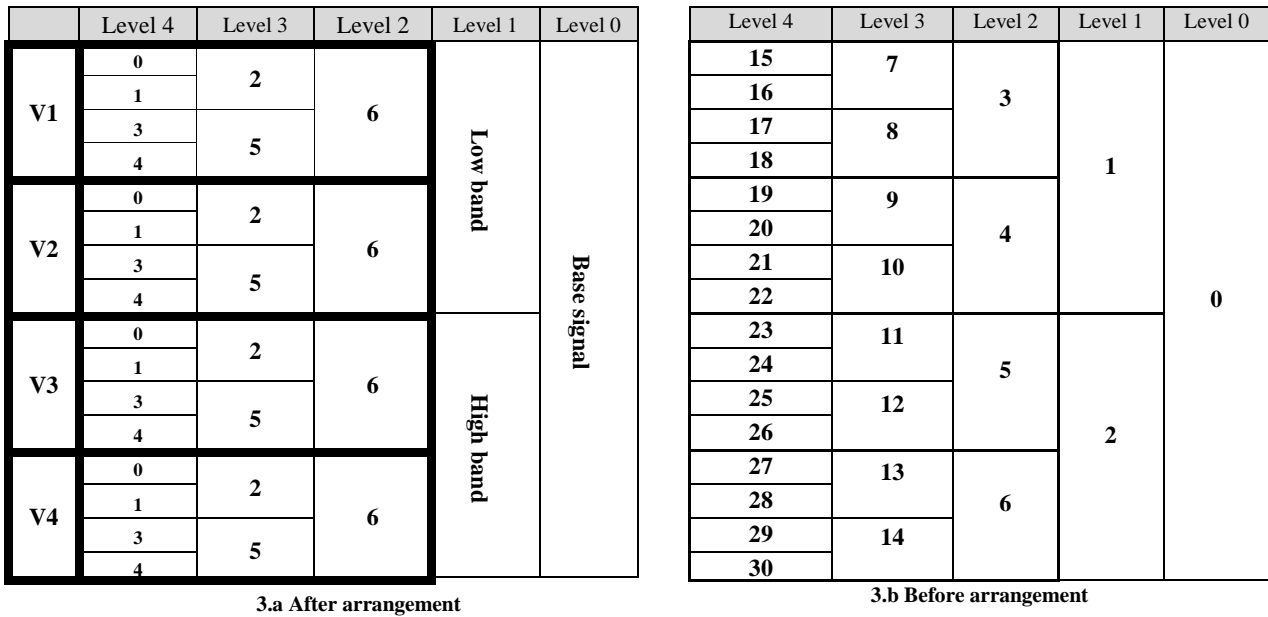


Figure 3: Clustering chart to explain the 4 points encoding algorithm, 3.a After encoding, and 3.b Before encoding

B. Best Tree Encoding (BTE), the history:

- 1) **BTE4**: BTE4 was first introduced in [1]. It has four wavelet decomposition levels. The strategy of arranging the blocks in BTE4 is done according to frequency bands as shown in figure3. BTE4 has a Frame size = window size = 20 ms, and “Shannon Entropy” is used. With 4-Dimensional vector, each vector has size of 8 bits and the maximum value is 64.
- 2) **BTE5**: BTE5 is considered the second generation of BTE4 by adding a new level for analyzing the data to increase the information resolution. The strategy of encoding the tree nodes in BTE5 is the same as of BTE4. It is also 4-Dimensional vector, Frame size =window size =20(ms), and “Shannon Entropy” is used for extracting the best tree.

4 PROPOSED HMM MODEL

Hidden Markov Model is used to solve the recognition problem. The hidden Markov Model Toolkit by university of Cambridge (HTK) [12] is used for configuring, training and testing the model. The proposed model is 3-state model. Each state is multi Gaussian statistical model to express the observed symbols. In our case the symbol here is the BTE vector. In HTK, the conversion from single Gaussian HMMs to multiple mixture component HMMs is usually one of the final steps in building the model. The mechanism provided to do this is the HHED MU command which will increase the number of components in a mixture by a process called *mixture splitting*. This approach to building a multiple mixture component system is extremely flexible since it allows the number of mixture components to be repeatedly increased until the desired level of performance is achieved.

The MU command has the form: `MU n itemList`

Where n gives the new number of mixture components required and "itemList" defines the actual mixture distributions to modify. This command works by repeatedly splitting the mixture with the largest mixture weight until the required number of components is obtained. The actual split is performed by copying the mixture, dividing the weights of both copies by 2, and finally perturbing the means by plus or minus 0.2 standard deviations e.g. `MU 3 {*.state [2-4].mix}` It is usually increasing the number of mixtures then re-estimating, then incrementing by 1 or 2 again and re-estimating, and so on until the required numbers of components are obtained. This also allows recognition performance to be monitored to find the optimum mixture. Better start with a lesser number of mixtures and work way up. As one cannot go in the reverse direction, that is, there is no way to merge mixtures in HTK. So use single Gaussian models first then increment so as to reach a mixture of 8.

5 OPTIMAL DEPTH AND SPLIT-ENERGY BEST TREE ENCODING (BTE7)

BTE7 is a new generation of BTE features where adding the new analysis levels makes that the information will be encoded into 63 bits instead of 7 bits in BTE4. It consists of two major changes that make it different of both BTE4 [1] and BTE5 [2].

- 1- Analysis of 7 levels is applied in Wavelet Packets Decomposition step on speech signal.
- 2- Split-Energy components are appended to the features vector.

C. *Optimal Depth*

In the following we explain the significance of the two listed additions/modifications above. In [2], it is shown that increasing of the levels of the wavelet decomposition is a way to improve the recognition result, so the first step that is used in the way of improvement is to increase the level. As the analysis period is 20(ms) and the sampling rate is 32kHz the frame length in sample will be $32000 \times 20 \times 10^{-3} = 640$ (samples), so the maximum number of times that signal can be down sampled during wavelet packet analysis is 9 decomposition levels at which only 1 sample will be available for each tree node at that level. Some efforts were done by Michelle Daniel [9] who mentioned that level 7 gives good results. Considering the results in [9] as well as the work done in [2] and recalling that at frame length equals to 640 (samples), only 5 samples will be available for analysis at level 7 in each tree node, then level 7 is considered as the optimal depth to be used for maximum efficiency. The strategy of encoding tree nodes in BTE7 is done according to the same manner in BTE4 and BTE5. Figure 4 explains the 7 levels encoding for BTE7. Figure 4.a shows the tree before encoding. Numbers in figure 4.a is the node index. Figure 4.b shows the nodes after BTE encoding.

	Level7 7	Level6 6	Level5 5	Level4 4	Level3 3	Level2 2	Level1 1	Level0 0		Level7 7	Level6 6	Level5 5	Level4 4	Level3 3	Level2 2	Level1 1	Level0 0											
V1	0	2	6	14	30	62	Low Band	Base Band	127	63	31	15	7	3	1	0												
	1								128																			
	3	5							13	21							29	45	61	60	129	64	32	16	34	3	1	0
	4																				130							
	7	9	28						37	44	52						60	59	131	65	33	17	36	8	3	1	0	
	8																		132									
	10	12	21	29					45	61	60	59					133	66	34	17	36	8	3	1	0			
	11																134											
	15	17	28	37					44	52	60	59					135	67	35	18	38	8	3	1	0			
	16																136											
	18	20	21	29					45	61	60	59					137	68	33	16	34	3	1	0				
	19																138											
	22	24	28	37	44				52	60	59	139	69				34	17	36	8	3	1	0					
	23											140																
	25	27	21	29	45				61	60	59	141	70				35	18	38	8	3	1	0					
	26											142																
	31	33	28	37	44				52	60	59	143	71				36	17	36	8	3	1	0					
	32											144																
	34	36	21	29	45				61	60	59	145	72				35	18	38	8	3	1	0					
	35											146																
	38	40	28	37	44				52	60	59	147	73				36	17	36	8	3	1	0					
	39											148																
	41	43	21	29	45				61	60	59	149	74				37	18	38	8	3	1	0					
	42											149																
	46	48	28	37	44	52			60	59	150	74	36	17			36	8	3	1	0							
	47										150																	
	49	51	21	29	45	61			60	59	151	75	37	18			38	8	3	1	0							
	50										151																	
	53	55	28	37	44	52			60	59	152	75	37	18			38	8	3	1	0							
	54										152																	
	56	58	21	29	45	61			60	59	153	76	38	18			38	8	3	1	0							
57	153																											
V2	62	High Band	4	2	0												
V3	62				5											
V4	62				6											

4.a After encoding

4.b Before encoding

Figure 4: BTE7 encoding. 4.a after encoding and 4.b before encoding

Numbers in figure 4.b is bit index. For example, tree node number 31 from figure 4.a is encoded as bit number 6 into the first component of BTE feature vector V1. If node 31 exists this will set bit number 6 in V1. The frame size =window size =20 (ms), and the Entropy type=log energy. In figure 4 vectors are surrounded in bold boxes, and as shown it is only four vectors too as in BTE4.

D. Split-Energy

Energy distribution percentages over the 4 portions of bandwidth, which are used to generate the 4 BTE feature components, can be used for adding more discrimination power in BTE features space. Many phones may have the same energy but not the same distribution. This will add 4 extra components each is the energy in the corresponding portion of the bandwidth. This can be done by the MATLAB function "wenergy" which give the energy for 1-D wavelet packet decomposition:

$$xx=wenergy(bt)$$

For a wavelet packet tree "bt", $xx = wenergy(bt)$ returns vector xx , which contains the percentages of energy corresponding to the leaf nodes of the tree "bt".

Figure 5 explains part of the MATLAB code for adding the energy components to the feature vector where Box4 encoder function distributes the energy values and adds the values of the energy that belong to the same vector.

6 EXPERIMENTS

A. Database:

The Database that is used in the experiments is consisting of 30 speakers all of them are men, speaking different sentences (2977 files in wav format, 2377 files used for Training, and 600 files used for Testing, 20 from each speaker).

B. Experiment framework:

- The framework we developed to train and test GMM HMM models using HTK to do feature extraction and build the baseline models which are used to align the training data.
- Microsoft C# (C sharp) is used for building the needed logic for building initial models of HTK.
- HTK tools for training and decoding is a collection of command-line options such as HERest and HVite. Each makes a special function, which is explained in detail in HTK book [12]

C. The Experiment variables

The experiment variables are listed below

- 1-Number of Gaussian Mixtures in HMM emitting states.
- 2-Number of Training Iterations.
- 3-Energy Components.

Energy components are appended to BTE main vector components for the three generations of BTE. This counts to 4 extra components in case of BTE4, BTE5 and BTE7. Figure 5 shows the MATLAB code to do the operation of the rearrangement of the tree after adding the new level and calculating the energy function for BTE7.

- 4-Delta and Acceleration coefficients of the vector components.

The performance of a speech recognition system can be greatly enhanced by adding time derivatives to the basic static parameters. In HTK, these are indicated by attaching qualifiers to the basic parameter kind. The qualifier D indicates that first order regression coefficients (referred to as delta coefficients) are appended, the qualifier A indicates that second order regression coefficients (referred to as acceleration coefficients). The delta coefficients are computed using the regression equation given in 1 as calculate in HTK book [12].

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2} \dots\dots\dots (1)$$

where d_t is delta coefficient at time t computed in terms of the corresponding static coefficients $c_{t-\theta}$ to $c_{t+\theta}$. The value of Θ is set using the configuration parameter DELTAWINDOW (Delta window size). The same formula is applied to the delta coefficients to obtain acceleration coefficients except that in this case the window size is set by ACCWINDOW (Acceleration window size) [12].


```

function [res] = btep1(frame)
    t = wpdec(frame,7,'db4','log energy');
    u = leaves (t);
    bt = besttree(t);
    v = leaves (bt);
    xx=wenergy(bt);
    res = box4encoder1(v,xx);
end
% *****Encodes a best tree for wavelet packets in 7 levels*****
function [x] = box4encoder1(a,xx)
    v = uint64([0;0;0;0]);
    x=zeros(4,1);
    n = max(size(a));
    for i = 1 : n
        switch(a(i))
            case 3
                v(1) = bitset(v(1),63);x(1)=x(1)+xx(i);
            case 4
                v(2) = bitset(v(2),63);x(2)=x(2)+xx(i);
            case 5
                v(3) = bitset(v(3),63);x(3)=x(3)+xx(i);
            case 6
                v(4) = bitset(v(4),63);x(4)=x(4)+xx(i);
            case 7
                v(1) = bitset(v(1),31);x(1)=x(1)+xx(i);
            case 8
                v(1) = bitset(v(1),62);x(1)=x(1)+xx(i);
            case 9
                v(2) = bitset(v(2),31);x(2)=x(2)+xx(i);
            case 10
                v(2) = bitset(v(2),62);x(2)=x(2)+xx(i);
            case 11
                v(3) = bitset(v(3),31);x(3)=x(3)+xx(i);
            case 12
                v(3) = bitset(v(3),62);x(3)=x(3)+xx(i);
            case 13
                v(4) = bitset(v(4),31);x(4)=x(4)+xx(i);
            case 14
                v(4) = bitset(v(4),62);x(4)=x(4)+xx(i);
            case 15
                v(1) = bitset(v(1),15);x(1)=x(1)+xx(i);
            case 16
                v(1) = bitset(v(1),30);x(1)=x(1)+xx(i);
            case 17
                v(1) = bitset(v(1),46);x(1)=x(1)+xx(i);
            case 18
                v(1) = bitset(v(1),61);x(1)=x(1)+xx(i);
            case 19
                v(2) = bitset(v(2),15);x(2)=x(2)+xx(i);
            case 20
                v(2) = bitset(v(2),30);x(2)=x(2)+xx(i);
            case 21
                v(2) = bitset(v(2),46);x(2)=x(2)+xx(i);
            case 22
                v(2) = bitset(v(2),61);x(2)=x(2)+xx(i);
            case 23
                v(3) = bitset(v(3),15);x(3)=x(3)+xx(i);
            case 24
                v(3) = bitset(v(3),30);x(3)=x(3)+xx(i);
            case 25
                v(3) = bitset(v(3),46);x(3)=x(3)+xx(i);
            case 26
                v(3) = bitset(v(3),61);x(3)=x(3)+xx(i);
            case 27
                v(4) = bitset(v(4),15);x(4)=x(4)+xx(i);
            case 28
                v(4) = bitset(v(4),30);x(4)=x(4)+xx(i);
            case 29
                v(4) = bitset(v(4),46);x(4)=x(4)+xx(i);
            case 30
                v(4) = bitset(v(4),61);x(4)=x(4)+xx(i);
            case 31
                v(1) = bitset(v(1),7);x(1)=x(1)+xx(i);
            case 32
                v(1) = bitset(v(1),14);x(1)=x(1)+xx(i);
            case 33
                v(1) = bitset(v(1),22);x(1)=x(1)+xx(i);
            .
            .
            case 253
                v(4) = bitset(v(4),57); x(4)=x(4)+xx(i);
            case 254
                v(4) = bitset(v(4),58); x(4)=x(4)+xx(i);
            end
        end
        x=[((double(v))/(10e16));x(1);x(2);x(3);x(4)];
    end
end

```

Figure 5: Adding energy to BTE7 MATLAB code

7 RESULTS:

HMM model with 3 emitting states and different Gaussian Mixtures in each state is used to model the recognition process. Table1 gives the percentage correct against the experiment variables, the qualifiers E4 means that adding the 4 energy percentage distribution to BTE4, BTE5 and BTE7, the qualifiers A, D means adding delta and acceleration components for BTE4, BTE5 and BTE7.

Table 1
EXPLAIN THE CORRECT AND ACCURACY PERCENTAGE USING BTE

Feature	Qualifiers	Vector size	No. of mixtures	Percent correct %	Accuracy %
BTE4	-	4	1	12.89	-3.6
			2	16.57	-0.71
			4	16.93	-0.95
			8	20.05	-2.48
	E4	8	1	9.21	6.86
			2	12.65	7.59
			4	12.05	7.42
			8	18.17	7.52
	A, D	12	1	15.36	-0.62
			2	19.9	-1.96
			4	26.62	1.54
			8	23.8	-1.35
	A, D, E4	24	1	11.68	7.05
			2	14.29	5.41
			4	19.8	6.24
			8	21.87	5.79
BTE5	-	4	1	10.85	-6.08
			2	14.44	-1.2
			4	16.83	-1.98
			8	16.08	-2.08
	E4	8	1	9.7	6.65
			2	15.05	7.12
			4	13.52	7.01
			8	15.38	7.29
	A, D	12	1	20.46	-3.92
			2	23.42	-6.98
			4	24.88	-3.28
			8	23.29	-4.13
	A, D, E4	24	1	13.85	8.52
			2	18.75	6.97
			4	19.19	6.6
			8	21.75	6.6
BTE7	-	4	1	9.09	7.83
			2	14.78	7.33
			4	16.13	7.76
			8	23.56	7.88
	E4	8	1	9.73	7.12
			2	14.79	8.76
			4	12.37	7.83
			8	16	8.1
	A, D	12	1	8.35	7.36
			2	15.82	11.06
			4	15.99	10.7
			8	16.19	10.08
	A, D, E4	24	1	12.01	10.87
			2	13.24	10.93
			4	15.81	12.66
			8	17.6	12.26

In table 1; the accuracy column indicates some percentage values in negative. To explain the significance of the negative value let us go through the following example. First let us recall equation 3 from HTK book for calculating the accuracy.

Assume given sample with the following transcription:



The sample contains 4 elements. Recall equation 3, N= 4;

Once the sample is successfully recognized by HTK, there will be many possible situations as listed below:

Where:

#D: number of deletions

#S: number of substitutions

#I: number of insertions

<u>Transcription</u>				<u>#D</u>	<u>#S</u>	<u>#I</u>	<u>%Accuracy</u>				
\s	\a	\f	\a	0	0	0	100%				
\s		\f	\a	1	0	0	75%				
\s	\e	\f	\a	0	1	0	75%				
\s	\a	\a	\f	\a	0	0	1	75%			
\s	\a	\a	\f	\f	\f	\f	\a	0	0	5	<u>-25%</u>

The negative value is a reflection of low stability of the recognizer. In case of negative accuracy, the insertion errors are too much as shown in the given example. The optimal situation or the most stable situation exists when the total number of recognized phones = the total number of expected phones. The expected phones are pre-evaluated before the recognition process. This process is called transcription.

A. Result of increasing wavelet decomposition level

Figure 6 provides comparison results BTE4, BTE5 and BTE7. The comparison is made for the effect of increasing the wavelet decomposition level. For BTE4 there are 4 levels, for BTE5 there are 5 levels and for BTE7 there are 7 levels. Percentage accuracy is included to measure recognizer accuracy which is a recognizer parameter that indicates how much it is the accuracy of the final results. It takes into account the insertion errors in the final results. The percentage correct which indicates how many times a phone instance was correctly labeled. The percentage correct can be calculated from equation 2 and the percentage accuracy can be calculated from equation 3, HTK book [12].

$$\text{Percent correct} = \frac{N - D - S}{N} \times 100\% \quad \dots (2)$$

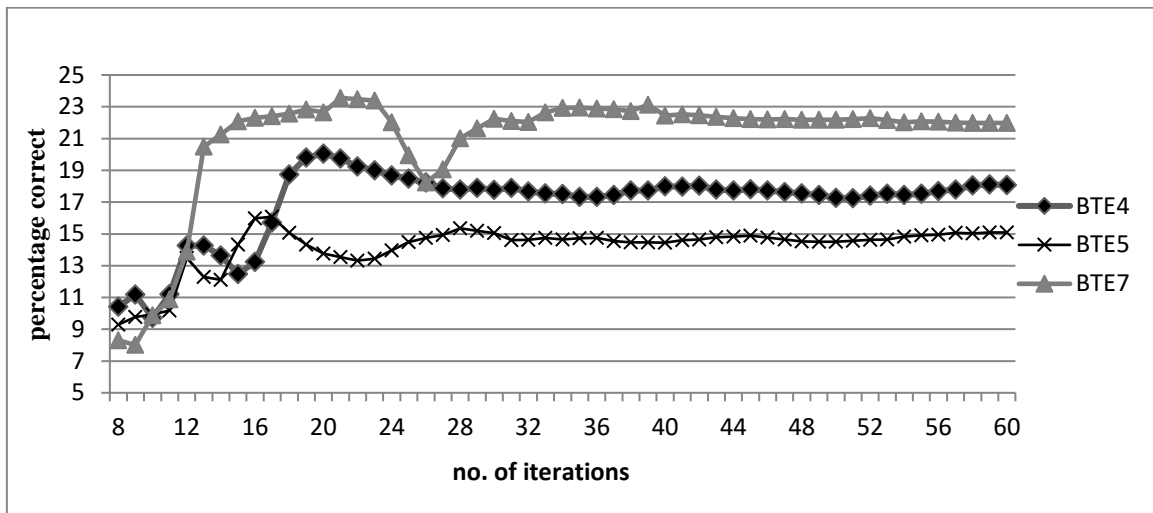
$$\text{Percent Accuracy} = \frac{N - D - S - I}{N} \times 100\% \quad \dots (3)$$

where N: the total number of labels in the reference transcriptions

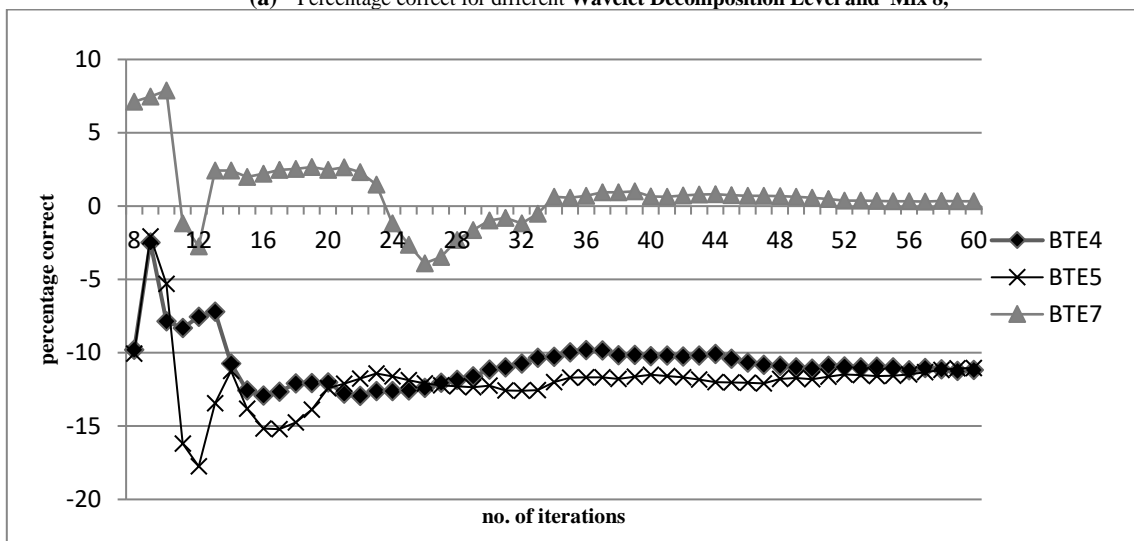
I: Number of Insertion errors in the results string.

D: Number of deletion errors in results string.

S: Number of substitution errors in results string.



(a) Percentage correct for different Wavelet Decomposition Level and Mix 8,



(b) Percentage accuracy for different Wavelet Decomposition Level and Mix 8,

Figure 6 Comparison results between BTE4, BTE5 and BTE7 at eight Gaussian mixtures without delta, acceleration and energy.

B. Result of increasing the number of Gaussian mixtures

As shown from figures 7.a, 7.b and 7.c that the increase of mixture count enhances the performance of Arabic phone recognition for the three generations of BTE without delta, acceleration and energy components. The optimal number of mixtures for BTE5 is 4 mixtures where it is 8 in BTE4 and BTE7.

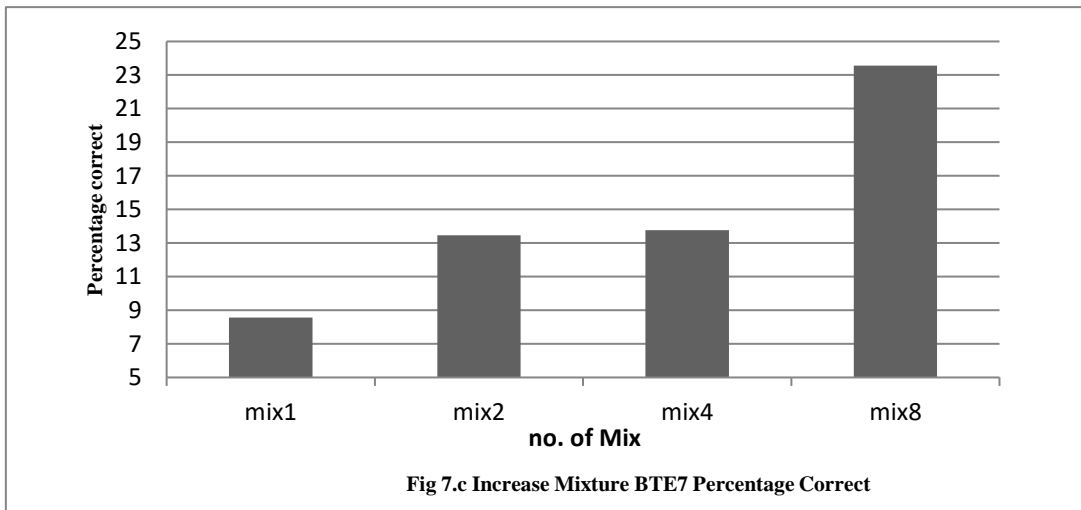
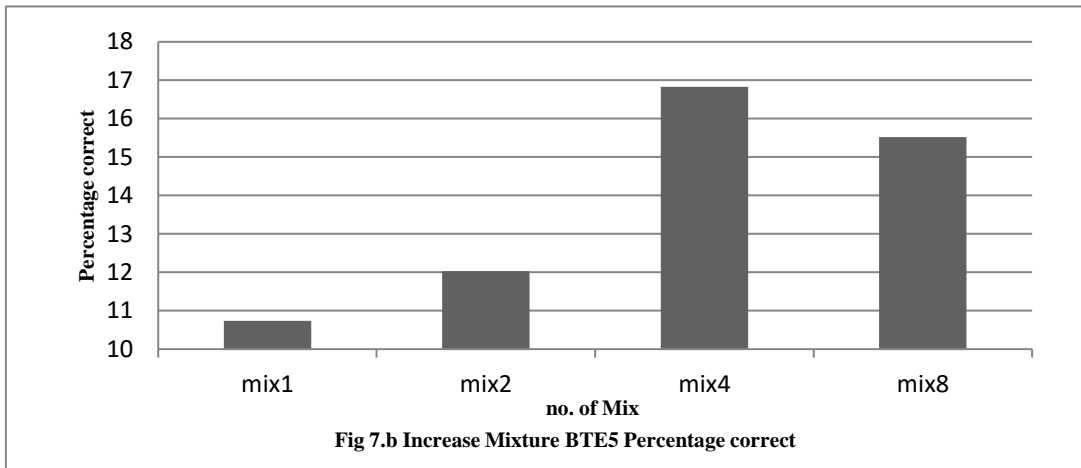
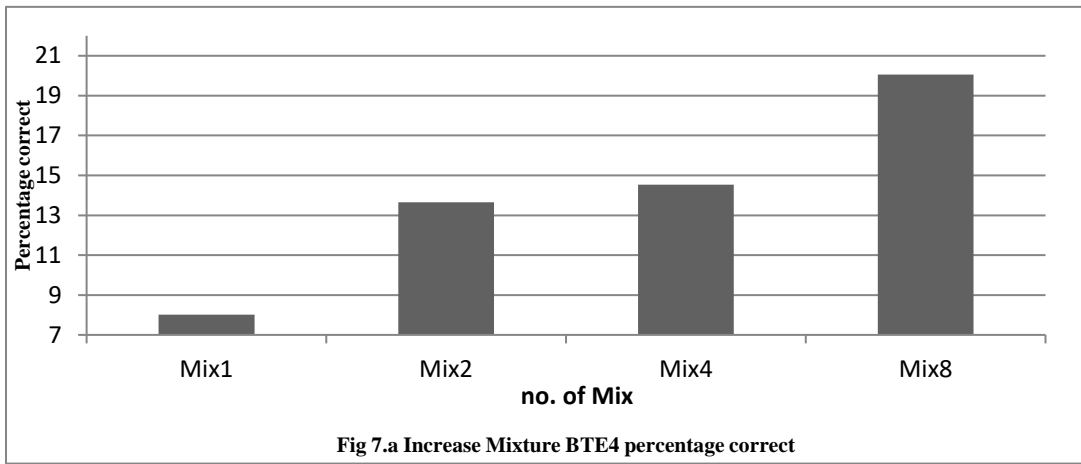


Figure 7 effect of increase number of Gaussian mixtures for different generation of BTE without delta, acceleration and energy components

B. Result of adding delta, acceleration and energy function

Figure 8 shows effect of adding energy, delta and acceleration to different types of BTE features when using four mixtures.

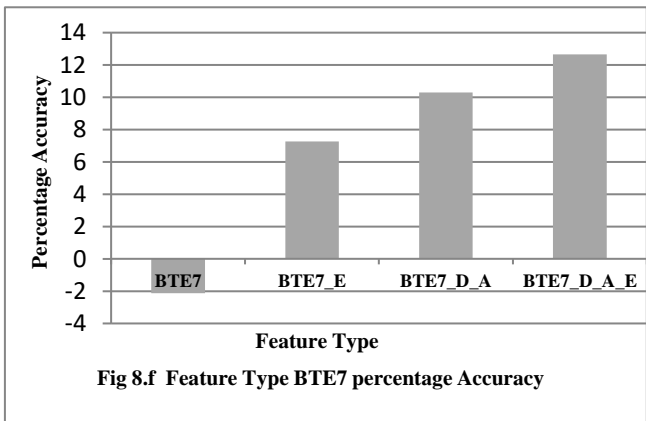
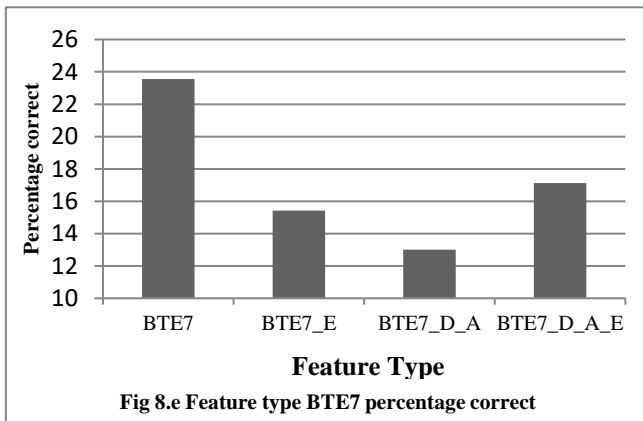
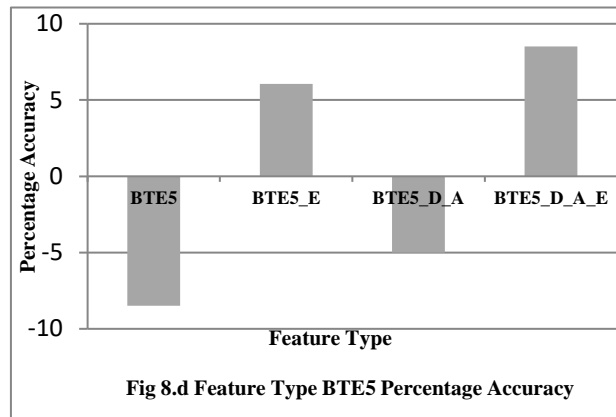
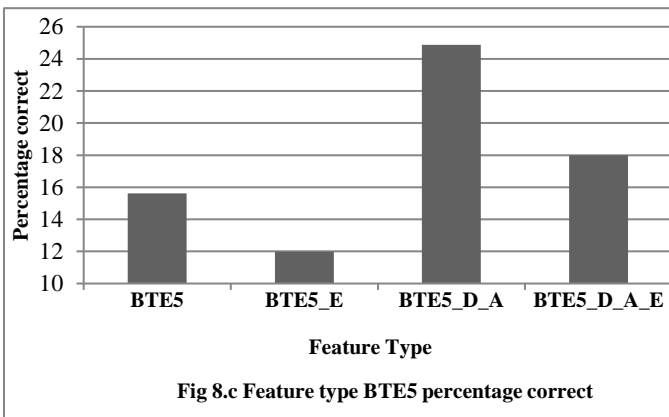
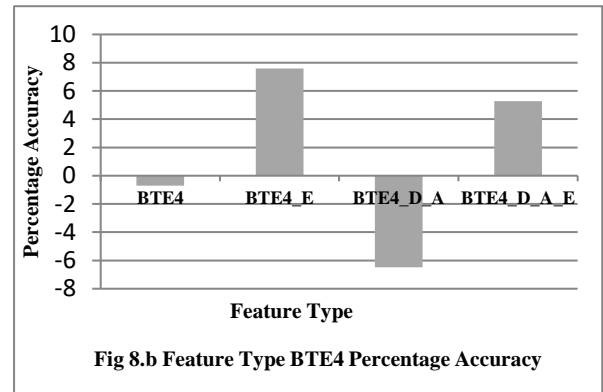
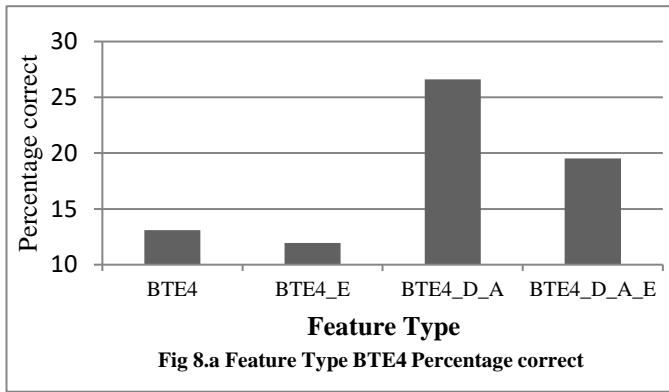


Figure 8 Effect of adding energy, delta and acceleration to different level of BTE features, for the percentage correct and percentage accuracy when using 4 Gaussian mixtures for BTE4 and BTE5 but with 1 mixture for BTE7.

As shown from figure 8 that adding the energy-split components to the features vector increases the accuracy of speech recognition process for BTE in all the three generations. The best performance appears when using delta and acceleration with the energy function which gives enhancement in both percentage correct and percentage accuracy. It is appears that adding energy component alone without delta and acceleration decreases the percentage correct by a small value for the three

generations of BTE, but it gives better enhancement for the percentage accuracy measurements. Also, adding delta and acceleration components without energy increase percentage correct for BTE4 and BTE5 but decrease it for BTE7 and vice versa for the accuracy, it decrease for BTE4 and BTE5 but increase for BTE7. Finally adding all in the same time delta, acceleration and energy give good enhancement for all generations both in parentage correct and percentage accuracy.

8 Conclusion:

Best Tree Encoding indicates a good start for new promising features. 7 levels wavelet packet analysis of speech signal is applied in the pre-processing phase of speech signal. By adding 2 more analysis levels over BTE5, it gives more resolution to store information which in return supports to make more discriminative features for Automatic Speech Recognition problem. In addition to, by including energy distribution over the 4 equal portions of the bandwidth instead of the total energy, gives more discriminative information about the phones. BTE7 gives 22% enhancements over BTE4 and BTE5. BTE7 indicates more accuracy than both BTE5 and BTE4.

REFERENCES

- [1] Amr M. Gody, "Wavelet Packets Best Tree 4 Points Encoded (BTE) Features", *The Eighth Conference on Language Engineering*, Ain-Shams University, Cairo, Egypt, 17-18 December 2008.
- [2] Amr M. Gody, Magdy Amer, Maha Adham, Eslam Elmghraby, "Automatic Speech Recognitions Using Wavelet Packet Increased Resolution Best Tree Encoded ", *The 13th Conference on Language Engineering CLE'2012*.
- [3] Olivier Rioul, Martin Vetterli IEEE, "Wavelet and Signal Processing", *IEEE Magazine on Signal Processing*, pp 14-38, Oct. 1991.
- [4] I.Daubechies, "Ten Lectures on Wavelets", *Rutgers University and AT&T Bell laboratories, Society for industrial and Applied Mathematics*, Philadelphia, PennsyLvania,1992
- [5] M.A. Cody, "The Wavelet Packet Transform: extending The Wavelet Packet Transform", *Dr. Dobb's Journal*, Apr. 1994.
- [6] H. L. Rufiner & J. Goddard, "A Method of Wavelet Selection in Phone Recognition", *Proceedings of the 40th Midwest Symposium on Circuits and Systems*. Vol. 2, pp. 889-891, Aug, 1997
- [7] M.A.Anusuya, and S.K.Katti, "Comparison of Different Speech Feature Extraction Techniques with and without Wavelet Transform to Kannada Speech Recognition", *International Journal of Computer Applications (0975 – 8887)*, Volume 26–No.4, July 2011.
- [8] H. Satori M. Harti and N. Chenfour, "Introduction to Arabic Speech Recognition Using CMU Sphinx System", submitted to *International Journal of Computer Science and Applications*, 2007.
- [9] Michelle Daniels, "Classification of Percussive Sounds Using Wavelet-Based Features", Final project, *Wavelets and Filter Banks course (ECE251C)*, 2010, <https://ccrma.stanford.edu/~danielsm/coursework.html>.
- [10] Shannon, Claude E." Prediction and entropy of printed English", *The Bell System Technical Journal*, January 1951.
- [11] R.R. Coifman, M.V. Wickerhauser, "Entropy-based Algorithms for best basis selection," *IEEE Trans. on Inf. Theory*, vol. 38, 2, pp. 713-718, 1992.
- [12] Steve Young, Mark Gales, Xunying Andrew Liu, Phil Woodland, et al." The HTK Book" ,Version 3.41 ,Cambridge University Engineering Department,2006 , <http://www.htk.eng.cam.ac.uk>.
- [13] Yahya ÖZTÜRK , "New speech processing strategies based on wavelet packet transform in cochlear implants", *MSc thesis, Dokuz Eylül university*, September, 2009, İZMİR.
- [14] Michel Misiti, Yves Misiti, Georges Oppenheim, Jean-Michel Poggi,"Wavelet Toolbox Computation Visualization Programming, MATLAB R2010a User's Guide", version 1, <http://www.mathworks.com>.
- [15] Hui Jiang, "Confidence measures for speech recognition: A survey", *ELSEVIER, Speech Communication* 45, PP 455–470, , 2005

Speech Recognition System Based on Wavelet Transform and Artificial Neural Network

Prof Ashrf H. Yahia^{*1}, El- Sayed A. El-Dahshan^{*2} and Engy R. Rady^{**3}

**Physics Department, Faculty of Science, Ain Shams University,
Cairo, Egypt*

¹ayahia@sci.asu.edu.eg.

**Physics Department, Faculty of Science, Ain Shams University,
Cairo, Egypt*

²e_eldahshan@yahoo.com

***Basic Science Department, Faculty of Computers and Information, Fayoum University,
El Fayoum, Egypt*

³engyragaay@gmail.com

Abstract—for the past several decades, designers have processed speech for a wide variety of applications ranging from mobile communications to automatic reading machines. Speech recognition reduces the overhead caused by alternate communication methods. Speech has not been used much in the field of electronics and computers due to the complexity and variety of speech signals and sounds. However, with modern processes, algorithms, and methods we can process speech signals easily and recognize the text. This paper presents an expert speech recognition system for isolated words based on a developed model of Discrete Wavelet Transform (DWT) and Artificial Neural Network (ANN) techniques to improve the recognition rate.

The data set was created by using English digits from zero to five and other nine words (spoken words) which was collected from four individuals in various time intervals. The feature vector was formed by using the parameters extracted by DWT. We have employed Daubechies 4-tap (db4) wavelet for the experiment. The feature vector was produced for all words and formed a training set for classification and recognition. Forty-four features were feed to feed forward backpropagation neural network (FFBPNN) for classification. The performance of the developed system was evaluated by using speech signals. The rate of correct classification was about 97.9 % for the sample speech signals.

Keywords: Discrete Wavelet Transform, Speech Recognition, Feature Extraction, Artificial Neural Network, Energy.

1. INTRODUCTION

The speech signal is the fastest and the most natural method of communication between humans. This fact has motivated researchers to think of speech as a fast and efficient method of interaction between human and machine [1]. The significance of speech recognition lies in its simplicity. This simplicity together with the ease of operating a device using speech, has lots of advantages. It can be used in many applications like, security devices, household appliances, cellular phones, ATM machines, and computers [2]. Automatic speech recognition methods, investigated for many years, have been principally aimed at realizing transcription and human computer interaction systems [3].

Speech features which are usually obtained via Fourier Transforms (FTs), Short Time Fourier transform (STFTs), or Linear Predictive Coding (LPC) techniques are, used for some kinds of Automatic Speech/Speaker recognition (ASR). They may not be suitable for representing speech/voice. These methods accept signal stationary within a given time frame and may therefore lack the ability to analyze localized events correctly [4]. Wavelet analysis has been proven as efficient signal processing techniques for a variety of signal processing problems [5]. It can be said that the benefits of using [6, 7, 8, 9] which are the new transforms are local; i.e. the event is connected to the time when it occurs. In studies wavelets used for speech/speaker recognition, it has been found that the original feature space can be augmented by the wavelet coefficients and will yield a smaller set of more robust features in the final classifier [7, 8, 9]. Artificial neural network is named from the network of nerve cells in the human brain [9]. ANNs have been investigated for many years in the hope of achieving human-like performance in

automatic speech recognition [10]. These architectures are composed of many non-linear computational elements operating parallel in patterns similar to the biological neural networks [11]. Artificial neural networks have been used extensively in speech recognition during the past two decades. The most important advantages of ANNs for solving speech recognition problems are their error tolerance and non-linear property [12].

In this study, an expert speech recognition system was introduced. A combination of DWT and ANN was developed to efficiently extract the features from real speech signals and then recognize them. It will aid to increase the percentage of the correct speech recognition and enable further research of speech/voice recognition to be developed.

The organization of the paper is as follows. Section 2 reviews the wavelet transform. Section 3 demonstrates the neural network classifier. The proposed algorithm is presented in section 4. Section 5 describes the design and implementation of the system. Finally section 6 presents the conclusion.

2. WAVELET TRANSFORM

The wavelet transform was borne out of a need for further developments from Fourier transforms. Wavelet analysis represents a signal as a weighted sum of shifted and scaled versions of a characteristic wave-like function. Moreover, wavelets are often irregular and asymmetric and enable better representation of signals composed of fast changes [13]. A wavelet transform involves convolving the signal against particular instances of the wavelet at various time scales and positions. Since we can model changes in frequency by changing the time scale and model time changes by shifting the position of the wavelet, we can model both frequency and location of frequency. The wavelet transform becomes an emerging signal processing technique and it is used to decompose and reconstruct non-stationary signals efficiently. The wavelet transform can be used to represent speech signals by using the translated and scaled mother wavelets, which are capable to provide multi-resolution of the speech signal [14]. Wavelet transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the speech signal. The wavelet analysis procedure is to adopt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet [15]. DWT is the most promising mathematical transformation which provides both the time and frequency information of the input signals. Wavelet transform is a technique to transform an array of N numbers from their actual numerical values to an array of N wavelet coefficients. DWT is any wavelet transform for which the wavelets are discretely sampled. It captures both frequency and location information. The digital speech signal $X[n]$ is filtered by high pass filter $H1(z)$ and low pass filter $H0(z)$. The filtered results are down sampled by 2. Since most of the speech energy concentrates on the low frequency band, the low pass filtered signals need to be split again into sub bands by applying the $H1(z)$ and $H0(z)$ filters as in Figure 1. This procedure repeats until the desired decomposition level is reached. At high frequencies, the DWT provides good time resolution and poor frequency resolution. At low frequencies, DWT gives good frequency resolution and poor time resolution and vice versa.

Daubechies wavelets are compact orthogonal filter banks which satisfy the perfect reconstruction condition. In addition, Daubechies wavelets have maximum number of vanishing moments for a given order so that they can be used to provide the good approximation of the original signal. The Daubechies 4-tap (db4) filter bank was chosen for this design work.

The DWT is defined by the following equation:

$$W(j, k) = \sum_j \sum_k x(k) 2^{-j/2} \Psi(2^{-j}n-k) \quad (1)$$

Where (Ψ_t) is a time function with finite energy and fast decay called the mother wavelet, j and k parameters refer to wavelet scale and translation factors. In wavelet analysis, we often speak of approximations and details. The approximations are the high-scale, low-frequency components of the signal (A). The details are the low-scale, high frequency components (D) [16].

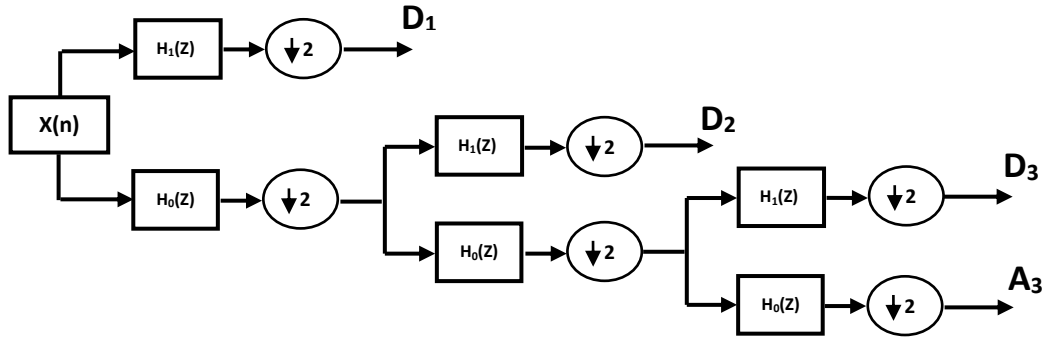


Figure 1: Discrete wavelet transform of a three stages analysis tree

3. NEURAL NETWORK

Artificial Neural Network (ANN) is non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs. The ANN may be regarded as a massive parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use [12]. The Multi Layer Perception (MLP) is a feed-forward network consisting of units arranged in layers with only forward connections to units in subsequent layers. The connections have weights associated with them. Each signal traveling along a link is multiplied by its weight. The input layer, being the first layer, has input units that distribute the inputs to units in subsequent layers. In the following (hidden) layer, each unit sums its inputs and adds a threshold to it and nonlinearly transforms the sum (called the net function) to produce the unit output (called the activation). The output layer units often have linear activations, so that output activations equal net function values [17]. MLP Neural Network is a good tool for classification purposes. It can approximate almost any regularity between its input and output [12]. The performance is measured by mean squared error (MSE):

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - y_a)^2 \quad (2)$$

Where y_t is the target value, y_a is the actual output, and n is the number of training data.

4. PROPOSED SYSTEM

A complete speech recognition system based on DWT and ANN was developed in this paper to achieve the goal of the research (increasing the accuracy of recognition). Figure 2 depicts the speech recognition system developed in this study. It consists of two stages: (a) data acquisition and preprocessing and (b) features extraction and classification.

In these studies, the developed system was successfully trained and tested in MATLAB version 7.10 using a combination of the Signal Processing Toolbox, Wavelet Toolbox, and Neural Network Toolbox for MATLAB.

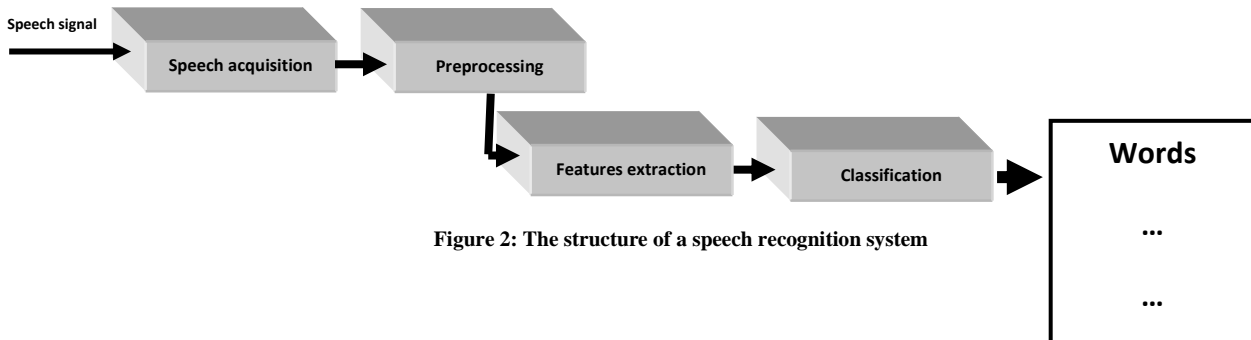


Figure 2: The structure of a speech recognition system

4.1. Preprocessing

The objective in the preprocessing stage was to modify the speech signal, so that it would be more suitable for the feature extraction analysis. A manual endpoint detection method was used to separate the word speech from the silent portions of the signal. The preprocessing stage includes removing the dc value of each signal to avoid dc offset problems and applying normalization on them to make the signals comparable. The signals were normalized by using the formula

$$x_{ni} = \frac{x_i - \bar{x}}{\sigma} \quad (3)$$

Where x_i is the i th element of the signal x , \bar{x} and σ are the mean and the standard deviation of the vector x , respectively, x_{ni} is the i th element of the signal after normalization.

Numbers from zero to five and the other nine words was uttered in English by 4 individuals, including 2 males and 2 females were transmitted to the computer by using a microphone and an audio card which had maximum 44 kHz sampling frequency and were recorded in a normal office environment by cool edit program version 2.

Each individual utterer were chosen as:

Sampling rate was 16 kHz.

Bits per sample (bit rate) were 16 bits.

Number of channels was one channel (mono).

Audio format was wave.

4.2 Feature Extraction

In order to get an expert system based on speech recognition, features extracted from the speech signals must be chosen well since the best classifier will perform poorly if the features are not chosen correctly. Consequently Discrete Wavelet Transformation (DWT) was performed on the samples in the database. Daubechies wavelet of order 4 (db4) at level 10 was found to produce the best feature representation. The 10-level DWT resulted in 11 logarithmically spaced frequency bands. The decomposition generates a set of vectors which contain signal information at different frequency bands. After the ten-level wavelet transform, the wavelet norm, energy, maximum, and minimum for each subband were computed in order to extract the feature vector of size 44 elements per utterance. The feature extraction steps are illustrated in Figure 3.

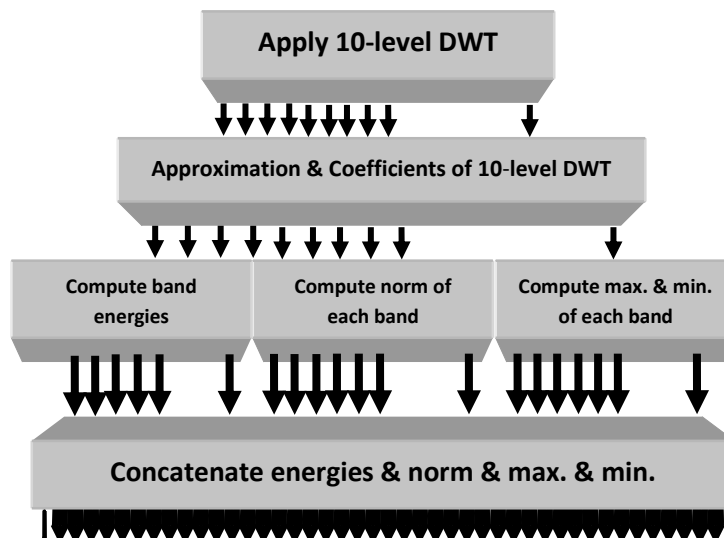


Figure 3: DWT-based feature extraction.

4.3 Classification

A database of 1800 utterances was created from the English language. 60% of these utterances were used for training, 20% were used for validation and 20% were used for testing. We used the FFBPNN for training and testing of the neural network. The training parameters used in this research are illustrated in Table 1. The architecture of the network is 5-layer architecture, which is a 44-node input layer, hidden layer with 19 nodes, hidden layer with 17 nodes, and hidden layer with 15 nodes followed by the 15-node output layer.

These were selected for the best performance after several experiments. A feature matrix of size 44×1800 which was collected for all the words were applied to the input of the neural network as in Figure 4.

Each layer of the network (5 layers) had a weight coming from the previous layer. The first layer weights came from the inputs. The last layer, which is the network output, was designed as a 15 binary digits for each feature vector.

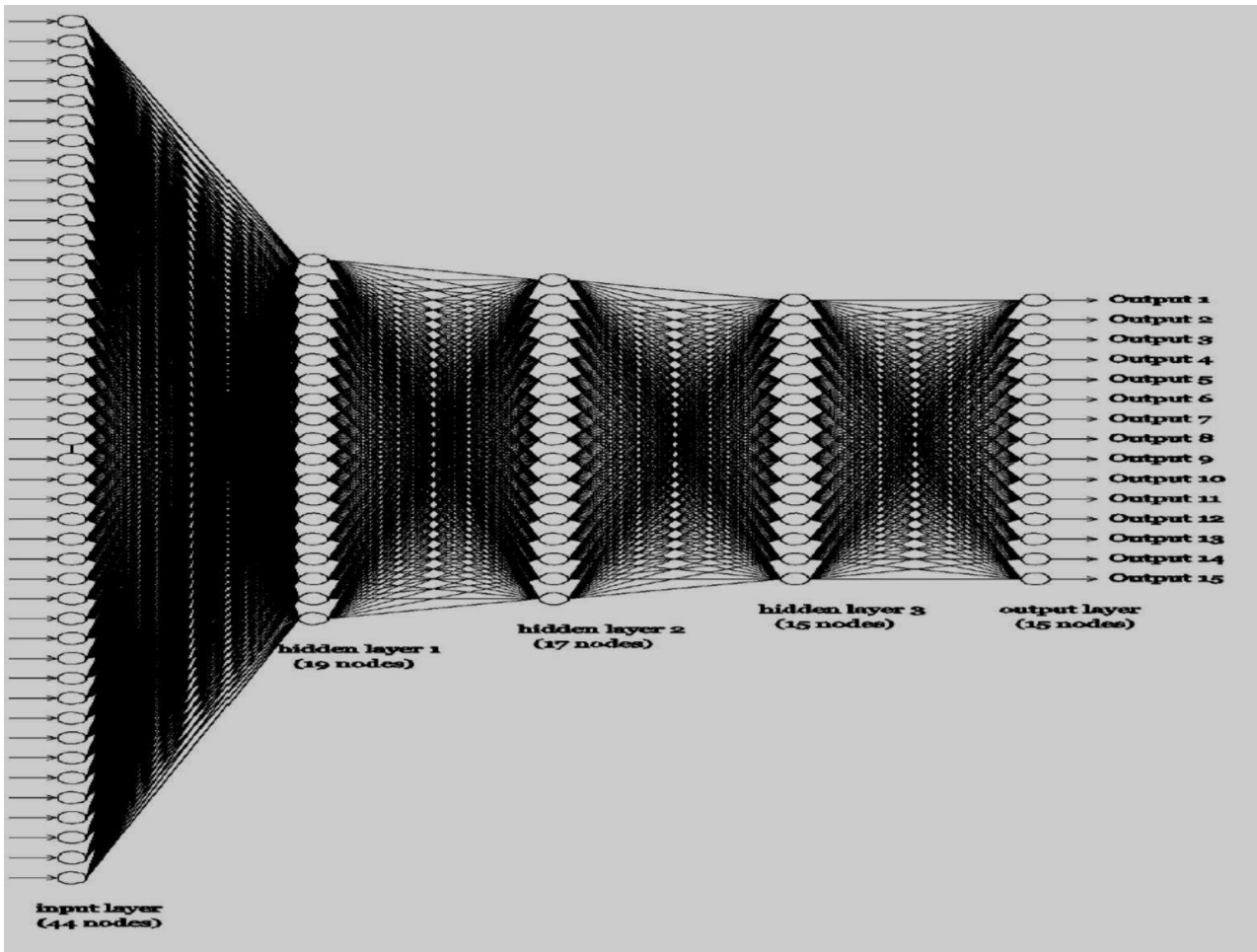


Figure 4: The feed forward backpropagation neural network

TABLE 1 PARAMETERS USED FOR THE NETWORK

Architecture	
Network type	feed-forward backpropagation
No. of layers	five layers: input, three hidden and output Input:44 Hidden:19, 17, 15 Output :15
Activation function	sigmoid
Training algorithm	Levenberg-Marquardt backpropagation
performance function(mse)	1.0000e-005
No. of epochs	1000

5. EXPERIMENT AND RESULT

The experiment was performed using a data base of 1800 English utterances for 15 words. Total of 4 individual speakers (2 males and 2 females) have spoken these 15 words. Each speaker speaks each word 30 times. Speech signals of a female and a male speaker for help word were shown in Figure 5 and 6, respectively. When the testing of the classifier was performed, an overall recognition accuracy of 97.9 % was achieved by means of FFBPNN. It indicated the effectiveness and the reliability of the proposed approach for extracting features from speech signals. Figure 7 shows a snap-shot for the GUI of the trained neural network. Testing results are tabulated in Table 2. A Receiver

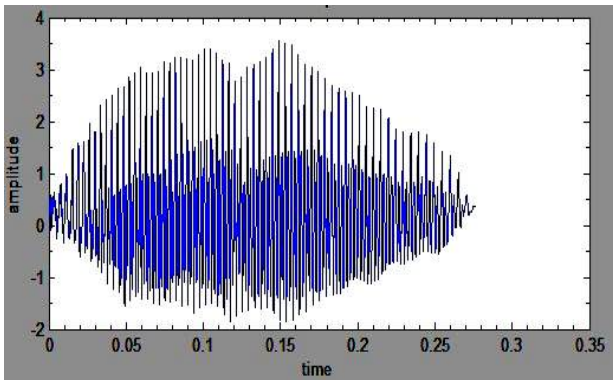


Figure 5: Speech signal of a female speaker for help word

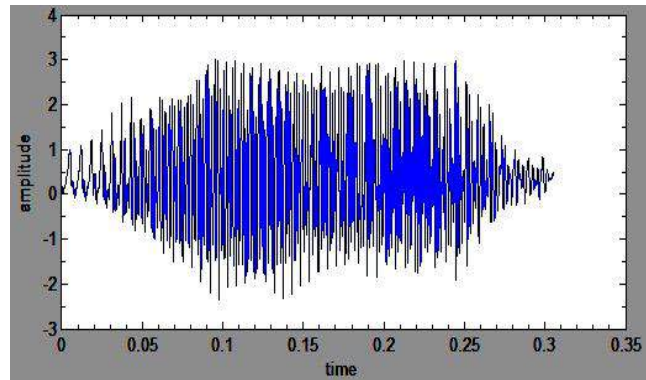


Figure 6: Speech signal of a male speaker for help word

Operating Characteristic (ROC) curve is also added in Figure 8 to indicate the performance of the recognition accuracies. The ROC curve is a plot of the true positive rate versus the false positive rate. A plot of the training errors, validation errors, and test errors appears, as shown in Figure 9. The best validation performance occurred at iteration 28.

Table 2: FFBPNN RECOGNITION RATE RESULTS

English Word	Total Number of Samples	Correct Classification	Incorrect Classification	The Average Recognition
Zero	120	114	6	95 %
One	120	117	3	97.5 %
Two	120	120	0	100 %
Three	120	114	6	95 %
Four	120	118	2	98.3 %
Five	120	120	0	100 %
Off	120	120	0	100 %
On	120	120	0	100 %
Play	120	118	2	98.3 %
Please	120	119	1	99.2 %
Sorry	120	115	5	95.8 %
Stop	120	117	3	97.5 %
Thanks	120	117	3	97.5 %
Ready	120	116	4	96.7 %
Help	120	117	3	97.5 %
Total	1800	1762	38	97.9 %

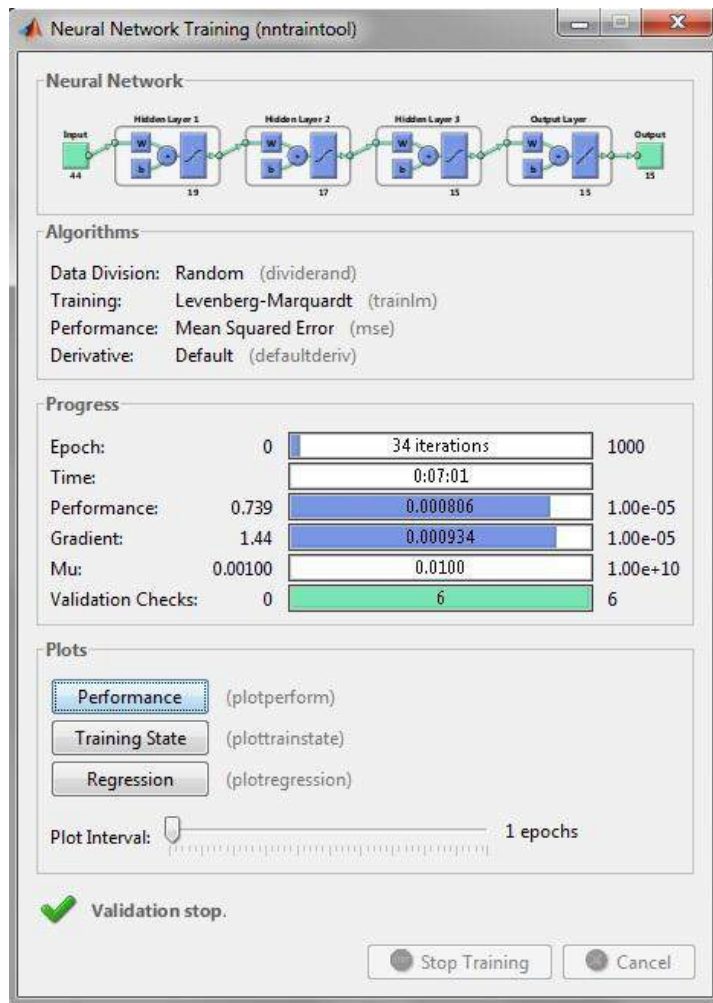


Figure 7. The GUI of the neural network training

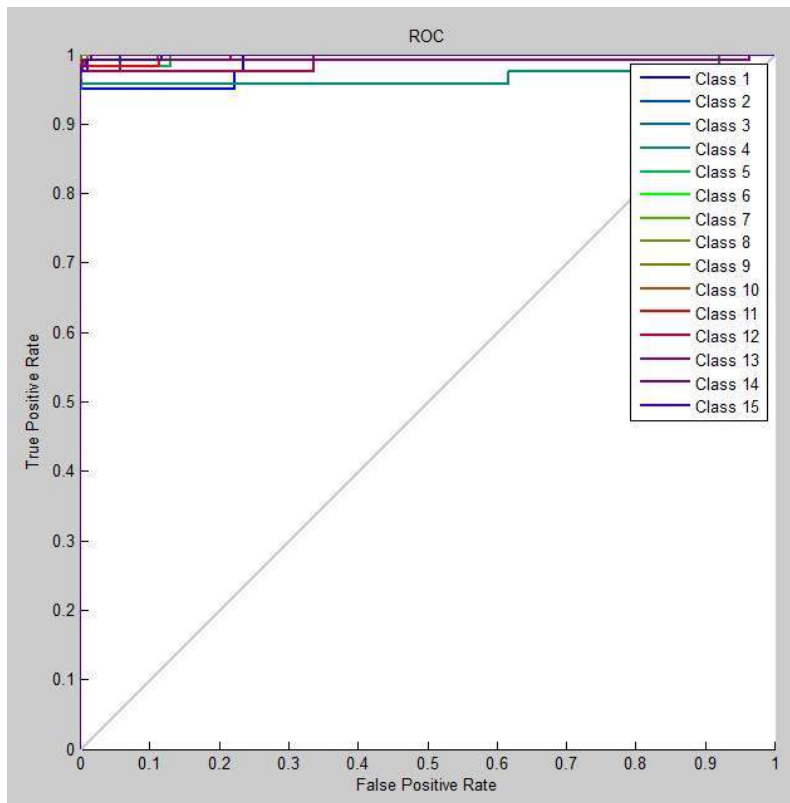


Figure 8. ROC

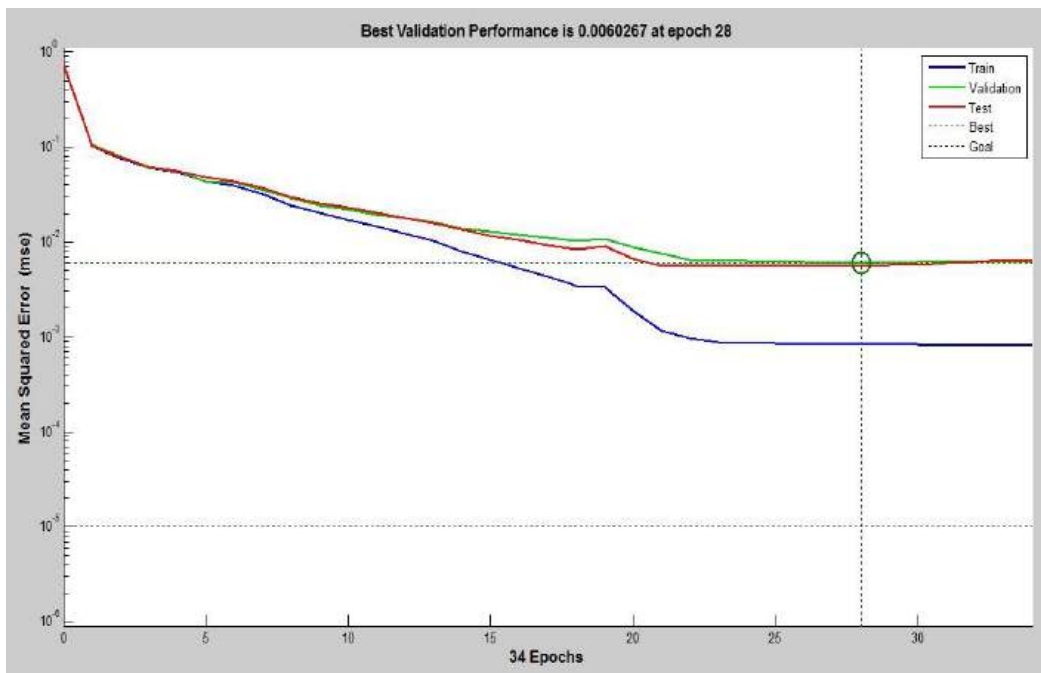


Figure 9. Performance

6. CONCLUSION

In this study, an expert speech recognition system for isolated words based on a developed model of Discrete Wavelet Transform (DWT) and Artificial Neural Network (ANN) techniques was proposed.

According to the experimental results, the proposed method can make an effectual analysis. The average identification rate of the system was 97.9 %. The stated results show that the proposed method can make an accurate and robust classifier.

REFERENCES

- [1] Ayadi, M.M.H.E., Kamel, M.S., Karray, F.: Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*(2011) 572-587
- [2] Patel, I., Dr. Srinivas Rao, Y., Speech Recognition using HMM with MFCC- An Analysis using Frequency Spectral Decomposition Technique, *Signal & Image Processing : An International Journal(SIPIJ)* , 1(2) (December 2010).
- [3] Trivedi, N., Dr. Kumar, V., Singh, S., Ahuja, S., and Chadha, R., Speech Recognition by Wavelet Analysis, *International Journal of Computer Applications* 15(8) (February 2011) 27–32.
- [4] Avci, E., and Akpolat, Z.H., Speech recognition using a wavelet packet adaptive network based fuzzy inference system, *SinceDirect*, vol.31, no. 3, 2006, pp 495- 503.
- [5] Sifarikas, M., Ganchev, T. & Fakotakis, Wavelet packets based speaker verification. In *Proceedings of the ISCA speaker and language recognition workshop – Odyssey’2004*, Toledo, Spain, May 31–June 3, (2004) 257–264.
- [6] Saito, N. “Local feature extraction and its application using a library of bases.” Phd thesis, Yale University (1994).
- [7] Buckheit, J. B. and Donoho, D. L., *WaveLab and Reproducible Research*, Dept. of Statistics, Stanford University, Tech. Rep. 474 (1995).
- [8] Wesfred, E., Wickerhauser, V., Adapted local trigonometric transforms and speech processing. *IEEE trans. on Signal Proc.* 41 N.12 (1993) 3596-3600.
- [9] Visser, E., Otsuka, M. & Lee, A spatio-temporal speech enhancement scheme for robust speech recognition in noisy environments, *Speech Communication*. 41 (2003) 393–407.
- [10] Alotaibi, Y.A., Investigation of spoken Arabic digits in speech recognition setting, *Informatics and Computer Sciences* 173 (1–3) (2005) 105–139.
- [11] Lampinen, J., Oja, E., Fast self-organization by the probing algorithm, In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume II (1989) 503-507, Piscataway, NJ. IEEE Service Center.
- [12] Haykin, S. *Neural Networks: A comprehensive Foundation*, Prentice Hall, 1999.
- [13] Canal, M.R., “Comparison of Wavelet and Short Time Fourier Transform Methods in the Analysis of EMG Signals,” *Journal of Medical Systems*, (2008)1-4.
- [14] Pang, J., Chauhan, S., FPGA Design of Speech Compression by Using Discrete Wavelet Transform, *WCECS 2008*, Francisco, USA, 22 - 24 October 2008, pp. 151 – 156.
- [15] An Introduction to Wavelets, The original version of this work appears in *IEEE Computational Science and Engineering*, Summer 1995, vol. 2, num. 2, published by the IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, USA,
- [16] Kadambe, S., Srinivasan, P., Application of Adaptive Wavelets for Speech, *Optical Engineering* 33(7) (July 1994) 2204-2211.
- [17] Vimal Krishnan, V.R., Babu Anto, P., Feature Parameter Extraction from Wavelet Subband Analysis for the Recognition of Isolated Malayalam Spoken Words, (*IJCNS*) *International Journal of Computer and Network Security*, 1(1) (October 2009).

An Approach for Mining Opinions in Arabic Religious Decrees

Ahmed M. Misbah^{*1}, Ibrahim F. Imam^{*2}

**Computer Science Department, Faculty of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport*

2033 - El Horreya, El Moshir Ismail St., behind Sheraton Building, Cairo, Egypt

¹ahmed.misbah@hotmail.com

²ifi05@yahoo.com

Abstract—Arabic Islamic Websites on the World Wide Web contain a large number of Religious Decrees issued by Islamic Scholars. Those Decrees are organized by topic or category using Text Categorization and Topic Detection techniques in Text Mining. The task of organizing Religious Decrees according to the opinion expressed by scholars into either Allowed (Halal) or Prohibited (Haraam) has not been addressed in research or implemented on Islamic Web Sites.

This paper proposes an approach that utilizes algorithms and techniques used in Opinion Mining to extract Opinion-oriented tokens within Arabic Religious Decrees and measure their Semantic Orientation. The measured Semantic Orientation is then used to identify if the opinion expressed within the decree is oriented towards Halal or Haraam. A dataset consisting of 60,000 Arabic Religious Decrees was collected and annotated from the World Wide Web to carry out the experimentation on the proposed approach using a number of Learning Algorithms. The highest accuracy rate achieved was 77% using Support Vector Machine Classifier.

1 INTRODUCTION

The World Wide Web contains a large number of Islamic Websites that introduce the Islamic Religion and its practices to both Muslims and Non-Muslims. Most of the content on Islamic Websites is in the form of text found in books, Religious articles, news, and Religious Decrees.

Arabic Religious Decrees on Islamic Websites are organized by topic or category. This can be achieved using Text Mining algorithms and techniques related to Text Categorization and Topic Detection. However, organizing Religious Decrees according to the opinions expressed by scholars using Opinion Mining tools and techniques has never been studied or implemented.

The work in this paper aims to address the problem of organizing large numbers of Arabic Religious Decrees according to the opinions expressed by Islamic scholars. An approach is proposed using Opinion Mining algorithms and technique to extract Opinion-oriented tokens within Arabic Religious Decrees and measure their Semantic Orientation. Based on the measured Semantic Orientation, the opinion expressed by the scholar on the topic being discussed within the Religious Decrees will be classified as either Allowed (Halal) or Prohibited (Haraam).

The approach proposed by this paper can be used to build tools that can automatically measure the semantic orientation of text within Arabic Religious Decrees and detect whether it expresses a Halal or Haraam opinion. Such tools can be used by Islamic Websites or Electronic Libraries to reorganize their datasets of Religious Decrees by the opinion of the scholar. This would give visitors the ability to browse and read through Religious Decrees easily by viewing the question and the final opinion of the scholar opposite the question without having to read through long and complicated text.

2 OPINION MINING AND SENTIMENT ANALYSIS IN ARABIC LANGUAGE

Text can be categorized into two main types: Facts and Opinions. Facts are objective expressions about entities, events and their properties. Opinions are subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties.

Opinion Mining aims to detect subjective expressions in text. Sentiment Analysis measures the polarity of sentiments and feelings expressed within subjective expressions. Both areas of research were promoted by the widespread of web applications that facilitate interactive information sharing and collaboration on the World Wide Web. Such web applications are described as User-generated content. Examples of such web applications include Web Blogs, Forums, Discussion Groups, Social Networks and Review sites such as Epinions.com, CNET and Amazon.

User-generated content contains large amounts of text expressing views and opinions in many different domains. Research in Opinion Mining and Sentiment Analysis introduces approaches and techniques to build tools and aggregators that gather, detect, and categorize opinions in User-generated content. Such tools would facilitate the process of finding information and assist in decision making.

Very few researches exist in Opinion Mining and Sentiment Analysis in Arabic language. Elhawary et al. [1] attribute the cause of this to the poor quality of subjective Arabic texts available on the World Wide Web. Arabic text on the World Wide Web is mostly written in Arabic Dialect, which leads to difficulties in building Text Corpora and Sentiment Lexicons that can be used for performing Feature Extraction and Sentiment Classification.

Previous works in Arabic language proposed approaches to conduct Sentiment Analysis against Arabic business reviews [2], [3]. The datasets used to benchmark those approaches were either obtained from Arabic review sites or translated text from English datasets. Other works proposed approaches to build Arabic Sentiment Lexicons similar to the SentiWordNet [4], [5]. Those lexicons can be used in corpus-based sentiment polarity calculation approaches.

3 PROPOSED APPROACH

The proposed approach in this paper consists of 4 phases as illustrated in Figure 1. The first phase is Text Data Collection and Annotation. The purpose of this phase will be to build an annotated text corpus for building a Sentiment Lexicon and evaluating the proposed approach using a number of learning algorithms. The text data will consist of Arabic Religious Decrees downloaded in HTML format from 5 well known Islamic sites. Simple Text Preprocessing will be executed against the data to prepare it for manual annotation. Manual Annotation involves labeling the text in the Arabic Religious Decree as either Halal, Haraam, Both or None.

The second phase is Feature Extraction. In this phase, a number of Feature Extraction techniques used in Opinion Mining will be used to extract Opinion-oriented tokens to build a Sentiment Lexicon in the next phase. The Feature Extraction techniques that will be applied include tokenization, stop word removal, POS tagging, filtering based in POS type, and Word Stemming.

The third phase is Sentiment Polarity Calculation. The purpose of executing this phase is to calculate the Semantic Orientation of the opinion-oriented terms extracted from the previous step. The Semantic Orientation will be calculated using an improved SO-PMI algorithm proposed in this paper. The output of this phase will be a Sentiment Lexicon consisting of all terms related to the Religious Text domain and their subjectivity status.

The fourth phase is Sentiment Classification. In this phase, overall sentiment expressed in the decree will be classified as either Halal or Haraam using a number of Learning Algorithms. The learning algorithms used for Sentiment Classification will be Average SO-PMI, Support Vector Machine Classifier, Naive Bayes Classifier, and k-Nearest Neighbor Classifier.

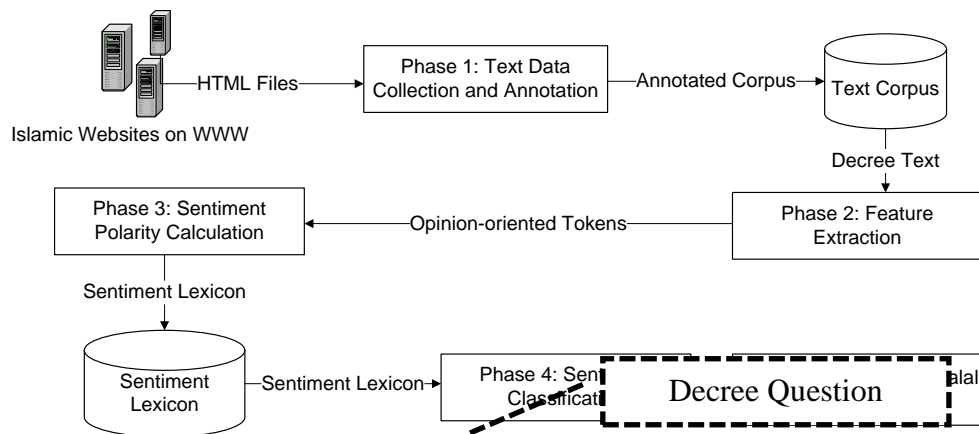


Figure 1: Proposed Approach Phases

A. Phase 1- Text Data Collection and Annotation

The purpose of this phase will be to build an annotated text corpus for building a Sentiment Lexicon and evaluating the proposed approach. The text data will consist of Arabic Religious Decrees, which will be downloaded in HTML format from well-known Islamic sites. Simple Text Preprocessing will be performed on the HTML files to prepare them for manual annotation. Manual Annotation involves labeling the text in the text data as either Halal, Haraam, Both or None.

- 1) *Text Data download using Web Crawlers:* The task of downloading text data required building Web Crawlers to download Arabic Religious Decrees from the Islamic Sites. Web Crawlers collected data from 5 well known and acknowledged Islamic sites [6] - [10]. The content on these websites is issued by famous Islamic Scholars, councils and organizations.
- 2) *Simple Text Preprocessing:* A tool was built to perform Simple Text Preprocessing on the data crawled from the web to prepare it for manual annotation. It filters downloaded HTML files from HTML Tags, Non Arabic characters, and special characters (except punctuation letters).
- 3) *Manual Data Annotation:* The dataset of Arabic Religious Decrees collected from Islamic sites lacked annotations to indicate its polarity (Halal or Haraam). It was required to manually annotate the data and insert it into a database schema. The importance of this step is to prepare the data for input into Supervised Learning Algorithms [11].

A team of post graduate students were assigned the task of manually annotating the decrees in the dataset. The annotators were instructed to read through the text of the Religious Decrees, which is composed of a question and an answer, and determine their annotation based on the opinion expressed in the answer. Keywords such as “حلال”, “مباح”, and “يجوز” indicate clearly that the decree expresses a Halal opinion. Keywords such as “حرام”, “ممنوع” and “باطل” indicate that the decree expresses a Haraam opinion. Annotators were also instructed to pay attention to negation letters as they shift the polarity of text.

The decrees were organized into the following categories:

- Halal (Decrees that clearly indicate that the topic inquired for is allowed),
- Haraam (Decrees that clearly indicate that the topic inquired for is prohibited),
- Both (Decrees that contain both opinions Halal and Haraam),
- None (Decrees that are objective and do not contain any opinion or subjective text).

B. Phase 2- Feature Extraction

In this phase, a number of Feature Extraction techniques used in Opinion Mining will be used to extract Opinion-oriented tokens for the purpose of building a Sentiment Lexicon in the next phase. The Feature Extraction techniques that will be applied include tokenization, stop word removal, POS tagging, filtering based in POS type, Word Stemming.

- 1) *Tokenization*: The first step that was executed was tokenization, which is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.
- 2) *Stop Word Removal*: The second step that was executed was the removal of Arabic stop words from the text except negation letters as they tend to shift the polarity of a given term. The list of Arabic stop words was gathered from a project on Source Forge [12]. The list of Arabic negation letters that were excluded from Arabic stop words included:

ليس، غير، لم، لَمَّا، لن، ما، لا، لات

Terms preceded with a negation letter were merged into a single token to form a bigram.

- 3) *Part of Speech Tagging*: The third step that was executed was Part of Speech Tagging. The library used to determine the POS tags of tokens was The Stanford Log-Linear Part of Speech Tagger [13].
- 4) *Filtration based on POS Tag*: This step involves extracting tokens that are relevant to detecting sentiment and measuring polarity in the document. The POS types extracted were Nouns, Verbs, Adjectives and Adverbs.
- 5) *Word Stemming*: Tokens extracted after filtering are stemmed. Word stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Table 1 shows how the words “الشاربان”، “الشاربين”، “الشارب”، and “الشاربية” are derived from the word “شارب” after applying stemming.

TABLE 1:

WORDS DERIVED FROM THE STEM “شارب”

Prefixes + Stem (Root + Pattern) + Suffixes		
Root	شرب	drink
Prefixes	ال	the
Stem	شارب	drinker
Suffixes	ين OR ان	dual
Suffixes	ون	plural
Suffixes	ة	feminine
الشاربان	the drinkers (dual)	
الشاربين	the drinkers (plural)	
الشارب	the drinker (masculine)	
الشاربيه	the drinker (feminine)	

The proposed approach uses an algorithm utilizing an implementation of Buckwalter Arabic Morphological Analyzer [14] and Stanford POS tagger [13] for obtaining each token’s stems. The algorithm simply returns the stem of one of the solutions returned by the Buckwalter Morphological Analyzer if the POS of the solution matches that of the token returned by the Stanford Morphological Analyzer. Figure 2 illustrates how the proposed algorithm returns the stem of the word “المسلم”:

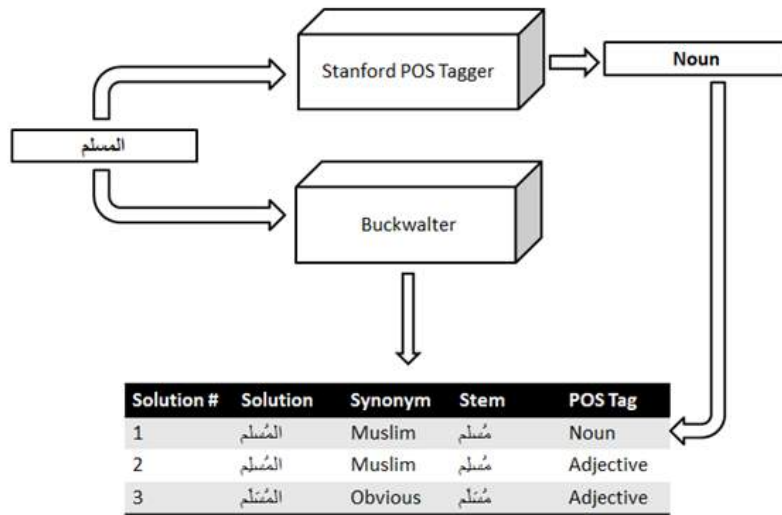


Figure 2: Proposed Word Stemming Technique

C. Phase 3- Sentiment Polarity Calculation

The third phase is Sentiment Polarity Calculation. The purpose of executing this phase is to calculate the Semantic Orientation of the opinion-oriented terms extracted from the previous step. The Semantic Orientation will be calculated using an improved SO-PMI algorithm proposed in this paper. The improvements done to the SO-PMI algorithm include:

1. Using custom built index files to execute search queries rather than corpora of web-based search engines,
2. Using a modified version of Turney's SO-PMI equation to calculate the Semantic Orientation of terms [15].

The output of this phase will be a Sentiment Lexicon consisting of all terms related to the Religious Text domain and their subjectivity status.

- 1) *Data Indexing*: Sentiment Polarity of Opinion-oriented tokens will be calculated in the proposed approach using a Corpus-based approach. A Corpus-based approach relies on co-occurrence patterns of words in large texts to determine their sentiment. Web-based Search Engines (e.g. Google, Yahoo, Bing and Alta Vista) [15] - [18] have been used in Corpus-based approaches to calculate term frequencies and co-occurrence frequencies by means of querying. However, using Web-based Search Engines had a number of disadvantages:
 - Web-based Search Engines offer their search APIs as a paid service,
 - The time required to calculate Sentiment Polarity for a large number of words (by means of querying) depends on the Internet connection's speed and reliability [27],
 - Web-based Search Engines return search results from their full corpus. This would result in inaccurate calculation of term sentiment polarity as search results will return frequency of co-occurrence of terms with each other in different domains [15].

Using a Full Text Retrieval Engine solves the disadvantages mentioned above. Apache Lucene [19], an open source Full Text Retrieval library, was used in the proposed approach to index text and run search queries on a local workstation.

- 2) *Semantic Orientation Calculation*: An improved SO-PMI equation was used to calculate the semantic orientation of tokens. The SO-PMI equation was based on an equation proposed by Turney [15]:

$$SO - PMI(\text{word}) = \log \frac{\text{hits}(\text{word NEAR } P_{\text{query}}) \text{ hits}(N_{\text{query}})}{\text{hits}(\text{word NEAR } P_{\text{query}}) \text{ hits}(N_{\text{query}})}$$

The values of P_{query} and N_{query} (also known as paradigm words) were defined in an arbitrary manner as follows:

P_{query} = حلال OR يجوز OR مباح OR مستحب OR مشروع OR يصح OR سنة OR فرض OR واجب

N_{query} = حرام OR مكروه OR ممنوع OR خطأ OR باطل OR بدعة OR فاسد OR مذموم OR فسق OR فجور OR ضلال

The NEAR operator (known as Proximity Search operator in Lucene) was given a distance value of 10 words. The distance value was defined based on the assumption that the word would occur with another paradigm word within the same sentence with at most 10 words between them.

Another set of paradigm words were defined to prevent counting results in which paradigm words are preceded with negation letters:

PN_{query} =حلال ليس OR يجوز لا يجوز OR مباح غير مباح OR مستحب غير مستحب OR مشروع غير مشروع OR يصح لا يصح OR سنة ليس سنة OR فرض ليس فرض OR واجب غير واجب

NN_{query} =حرام ليس حرام OR مكروه غير مكروه OR ممنوع غير ممنوع OR خطأ ليس خطأ OR باطل غير باطل OR بدعة ليس بدعة OR فاسد غير فاسد OR مذموم غير مذموم OR فسق ليس فسق OR فجور ليس فجور OR ضلال ليس ضلال

Results returned from every paradigm word in the new set are subtracted from those returned from the original set. This would ensure that the number of hits returned will consist only of the paradigm word, neglecting the ones preceded with negative tokens. Figure 3 illustrates an example of executing the word "زواج" with the paradigm words "مباح" and "غير مباح" from the P_{query} and PN_{query} and how the new hit results are calculated.

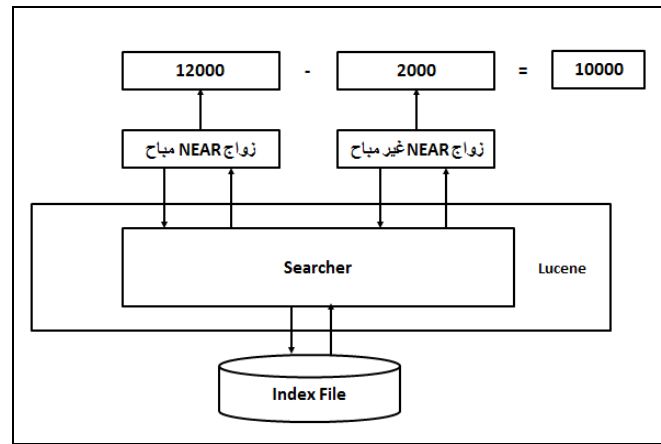


Figure 3: An example of executing the word "زواج" with "مباح" on an Index File

Therefore, the improved SO-PMI equation that will be used to calculate Sentiment Polarity of tokens will be as follows:

$$SO - PMI(\text{word}) = \log \frac{\text{hits}(\text{word NEAR } PT_{\text{query}}) \text{ hits}(NT_{\text{query}})}{\text{hits}(\text{word NEAR } NT_{\text{query}}) \text{ hits}(PN_{\text{query}})}$$

$PT_{\text{query}} = P_{\text{query}} - PN_{\text{query}}$

$NT_{\text{query}} = N_{\text{query}} - NN_{\text{query}}$

D. Phase 4- Sentiment Classification

The fourth phase is Sentiment Classification. In this phase, overall sentiment expressed in the decree will be classified as either Halal or Haraam. The learning algorithms used for Sentiment Classification will be Average SO-PMI, Support Vector Machine Classifier, Naive Bayes Classifier, and k-Nearest Neighbor Classifier.

- 1) *Average SO-PMI Algorithm:* Average SO-PMI is an algorithm proposed by Turney [15] to perform Sentiment Classification of text documents. It calculates the average of semantic orientation weights of the terms that occur in every text document represented in the Vector Space Model (Figure 4). The average value is compared with a threshold. If the value is greater than the threshold, then the decree is oriented towards Halal opinion, otherwise the decree expresses a Haraam opinion.

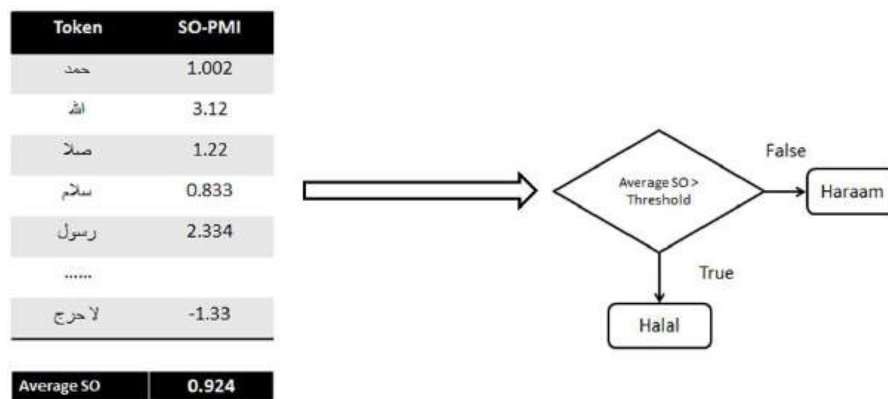


Figure 4: An example of an input to Average SO-PMI Algorithm and output

- 2) *Support Vector Machine Classifier*: SVM^{Light} library [20] was used as SVM implementation in the proposed approach. The classifier requires an annotated dataset to be split into a training set and a test set. The training set is inputted to the learning algorithm so that it would learn a trained model. The test set is inputted to the trained model for classification of unknown data. The SVM tool was configured to use a Biased Hyperplane and a Linear Kernel Function.
- 3) *Naïve Bayes and k-Nearest Neighbor Classifiers*: LingPipe Library [21] was used as an implementation of Naïve Bayes and k-Nearest Neighbor Classifiers. The value of k used was equal to 5 and Euclidean distance was used as a distance metric.

4 EXPERIMENTATION AND RESULTS

The purpose the experimentation is to evaluate the accuracy of the proposed approach using the 4 Sentiment Classification algorithms. A number of steps were implemented in all experimentations. The first step is Data Preparation. In this step, Arabic Religious Decrees from the text corpus and Opinion-oriented tokens from the Sentiment Lexicon are represented in the Vector Space Model. The term vectors are inputted to each classification algorithm. Accuracy rate from the classification results was calculated using a Confusion Matrix. Accuracy was calculated using the following equation [22]:

$$Accuracy = \frac{(a + d)}{a + b + c + d} = \frac{(TN + TP)}{TN + FP + FN + TP}$$

- TN = Decree labeled Haraam and was predicted Haraam by proposed approach**
TP = Decree labeled Halal and was predicted Halal by proposed approach
FP = Decree labeled Haraam and was predicted Halal by proposed approach
FN = Decree labeled Halal and was predicted Haraam by proposed approach

A. Experimentation using Average SO-PMI Algorithm

Data in this experimentation was represented in the Vector Space Model, which is an algebraic model for representing text documents as vectors of identifiers, such as index terms. Documents and queries are represented as vectors. Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these values, also known as (term) weights, have been developed. This experimentation uses SO-PMI as term weights.

The dataset used for this experimentation consisted of 8689 Halal decrees and 10355 Haraam decrees. Tokens of every decree were extracted along with their corresponding SO-PMI value from the Sentiment Lexicon. The decree is represented in Vector Space Model as shown in Figure 5.

Token	Decree 1
حمد	1.002
الله	1.02
صلا	1.22
سلام	0.833
رسول	2.334
.....
لا حرج	-1.33

Figure 5: Text of a Decree represented in Vector Space Model

Two Term Vectors were used to represent each decree in this experimentation. The first vector consisted of tokens of all POS types and the second vector consisted of only Adjectives and Adverbs. The purpose of using two Term Vectors was to compare the accuracy rates obtained by using all POS types [23] - [27] and only Adjectives and Adverbs which are known to express opinion [16], [23], [28], [29] more accurately than Nouns and Verbs.

Seven threshold values were used in this experimentation (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6) to determine the SO-PMI baseline. The SO-PMI baseline is the value at which the accuracy of the classification is highest. Though previous works [27], [30] have proposed a number of algorithms for calculating SO-PMI baseline, this experimentation defines 7 threshold values in an arbitrary manner to calculate the SO-PMI baseline.

Figure 6 and Figure 7 demonstrate the results obtained from this experimentation.

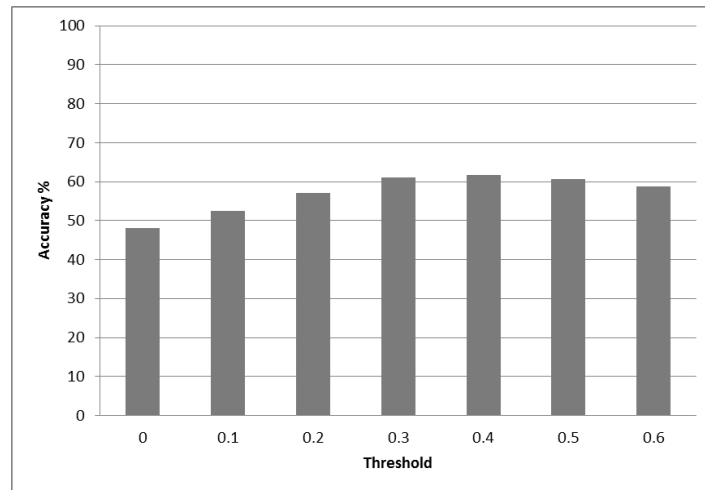


Figure 6: Results obtained using All unigrams+bigrams(valence shifters)+POS+Stems in Average SO-PMI Experimentation

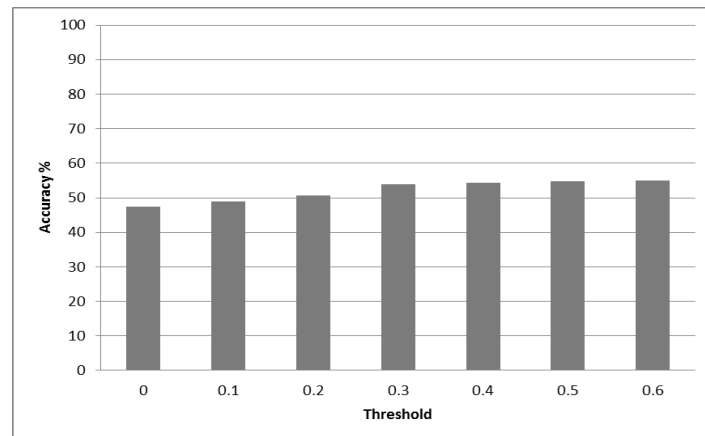


Figure 7: Results obtained using Adjective/Adverb unigrams+bigrams(valence shifters)+POS+Stems in Average SO-PMI Experimentation

B. Experimentation using Supervised Learning Algorithms

Data preparation for Supervised Learning Algorithms required splitting the dataset of 8689 Halal decrees and 10355 Haraam decrees such that 70% of the data is used as a training set and 30% is used as a test set. For SVM, every Religious Decree was represented in 4 term vectors. Two term vectors consisted of all POS types and only Adjectives and Adverbs with their corresponding SO-PMI values. The other two term vectors used a different weighting schema called Presence. In a Presence Term Vector, a 1 or 0 value is

used as the weight of the term depending on its occurrence in the decree. If the term is present in the decree a 1 is used otherwise 0. Term count in this weighting schema was ignored [31], [32].

In the Naïve Bayes and kNN Classifiers experimentation, 2 term vectors were used to represent each Religious Decree. The two term vectors consisted of all POS types and only Adjectives and Adverbs with their corresponding Presence values.

Figure 8 and Figure 9 demonstrate the results obtained in the experimentation conducted using Supervised Learning Algorithms

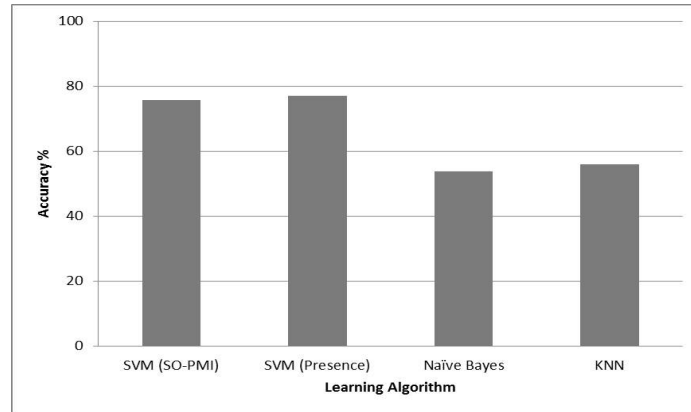


Figure 8: Results obtained using All unigrams+bigrams(valence shifters)+POS+Stems in Supervised Learning Algorithms' Experimentation

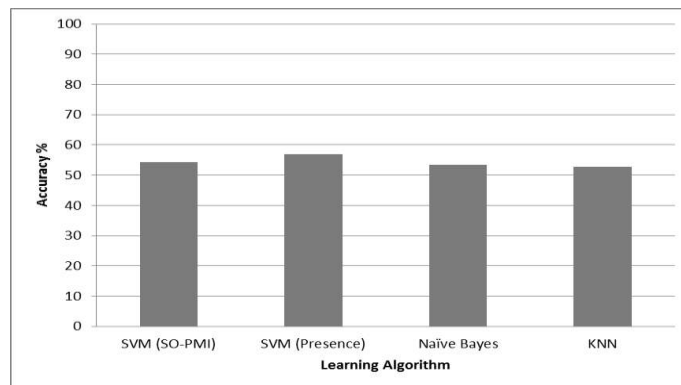


Figure 9: Results obtained using Adjective/Adverb unigrams+bigrams(valence shifters)+POS+Stems in Supervised Learning Algorithms' Experimentation

5 CONCLUSIONS

The highest accuracy rate obtained using Average SO-PMI Algorithm was 61.59% using all POS types at a threshold value of 0.4. Using all POS types produced higher accuracy rate than using Adjectives and Adverbs only. This suggests that Nouns and Verbs must be encoded as features in every religious decree's term vector as they affect the overall polarity of the opinion expressed. It also suggests that the optimum threshold value for this specific domain of texts and problem statement is 0.4.

The accuracy rates in Average SO-PMI Algorithm experimentation ranged from 47% – 61%. The reason behind these low accuracy rates is that the approach relies on selecting tokens based on their POS types and classifies Religious Decrees based on the weights obtained for those tokens. This has proven to be inefficient since Religious Decrees contain a lot of terms/phrases that are objective. Scholars tend to issue their opinion in one or two sentences then support their opinion with many sentences that tend to be informative. Including the weights calculated for tokens appearing in objective sentences affects the overall Semantic Orientation of the decrees.

The highest accuracy rate of all experimentations was obtained using Support Vector Machine Classifier, stemmed unigrams and bigrams (negation letters) and Presence as a weighing schema. Using Presence outperformed SO-PMI which suggests that probabilistic weights of terms in decree are not measure of its overall polarity. Naïve Bayes and kNN performed poorly when compared to SVM and compared to their performance in other Text Mining problems such as Text Categorization. This suggests that the use of large-margin classifiers rather than probabilistic classifiers is more suitable for this particular problem statement since data is tightly correlated.

It is recommended to use SVM as a classifier, all POS types, and Presence as a weighting schema to classify religious decrees based on their opinion. The low accuracy rates obtained by other learning algorithms suggest the hypothesis that they are not suitable for this particular domain of text and this particular problem statement.

REFERENCES

- [1] M Elhawary, M Elfeky, "Mining Arabic Business Reviews," in *IEEE International Conference on Data Mining Workshops*, 2010.
- [2] K Ahmad, D Cheng, Y Almas, "Multi-lingual Sentiment Analysis of Financial News Streams," in *Proceedings of Science, Grid Technology for Financial Modeling and Simulation*, Italy, 2006.
- [3] Y Almas, K Ahmad, "A note on extracting 'sentiments' in financial news in English, Arabic & Urdu," in *Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages Linguistic Institute*, Stanford, California, USA, pp. 1-12, 2007.
- [4] H. Ali, M. Rashwan, S. Elrahman, "Generating Lexical Resources for Opinion Mining in Arabic Language Automatically", in *Proceedings of 11th Conference on Language Engineering*, Faculty of Engineering, Ain Shams University, Cairo, Egypt, December 2011.
- [5] A. Esuli and F. Sebastiani, "SentiWordNet: A publicly available lexical resource for opinion mining," in *Proceedings of Language Resources and Evaluation (LREC)*, 2006.
- [6] Islam Way Web Site: <http://www.islamway.com>, (accessed on July 2010).
- [7] Islam Online Web Site: <http://www.islamonline.net>, (accessed on June 2009).
- [8] Islam Question and Answer Web Site: <http://islamqa.com/ar>, (accessed on June 2009).
- [9] "Islam Web" Web Site: <http://www.islamweb.net>, (accessed on June 2009).
- [10] Al Eman Web Site: <http://www.al-eman.com/>, (accessed on June 2009).
- [11] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 115–124, 2005.
- [12] Arabic Stop Words, Available from: <http://sourceforge.net/projects/arabicstopwords>, (accessed on June 2009).
- [13] Stanford Log-linear Part-Of-Speech Tagger, Available from: <http://nlp.stanford.edu/software/tagger.shtml>, (accessed on June 2010).
- [14] AraMorph, Available from: <http://www.nongnu.org/aramorph/>, (accessed on June 2009).
- [15] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 417–424, 2002.
- [16] Turney, P. D., & Littman, M. L., "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," in Technical Report ERB-1094, National Research Council Canada, Institute for Information Technology, 2002.
- [17] A. M. Misbah, I. F. Imam, "Mining Opinion in Arabic Data: A Comparison between Supervised and Unsupervised Classification Approaches", in *Proceedings of 11th Conference on Language Engineering*, Faculty of Engineering, Ain Shams University, Cairo, Egypt, December 2011.
- [18] A. M. Misbah, I. F. Imam, "Mining Opinions in Arabic Text using an Improved Semantic Orientation using Pointwise Mutual Information Algorithm", in *Proceedings of 8th International Conference on Informatics and Systems*, Faculty of Computers and Information, Cairo University, Cairo, Egypt, May 2012.
- [19] Apache Lucene, Available from: <http://lucene.apache.org/java/docs/index.html>, (accessed on February 2012).
- [20] SVMLight Support Vector Machine Classifier, Available from: <http://svmlight.joachims.org>, (accessed on June 2009).
- [21] LinePipe, Available from: <http://alias-i.com/lingpipe/index.html>, (accessed on June 2010).
- [22] R. Kohavi and F. Provost, "Glossary of terms," in *J. Mach. Learn.* 30, 2/3, 271–274, Editorial for the special issue on Applications of Machine Learning and the Knowledge Discovery Process, 1998.
- [23] A. Andreevskaia and S. Bergler, "Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses," in *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.
- [24] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss analysis," in *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, 2005.
- [25] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: Mining customer opinions from free text," in *Proceedings of the International Symposium on Intelligent Data Analysis (IDA)*, number 3646 in Lecture Notes in Computer Science, pp. 121–132, 2005.
- [26] Takamura, H., Inui, T., & Okumura, M., "Extracting semantic orientations of words using spin model," in *Proceedings of the 43rd annual meeting of the ACL*, Ann Arbor, pp. 133–140, 2005.
- [27] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," in *ACM Transactions on Information Systems (TOIS)*, vol. 21, pp. 315–346, 2003.
- [28] J. Wiebe, "Learning subjective adjectives from corpora," in *Proceedings of AAAI*, 2000.
- [29] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V. S. Subrahmanian, "Sentiment analysis: Adjectives and adverbs are better than adjectives alone," in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.
- [30] P. Chaovalit and L. Zhou, "Movie review mining: A comparison between supervised and unsupervised classification approaches," in *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, 2005.
- [31] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86, 2002.
- [32] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," in *Computational Intelligence*, vol. 22, pp. 110–125, 2006.

Educating Illiterate People on Mobile Sets

Doha Yousef^{*1}, Manar Ahmed^{*2}, Manal Ezzat^{*3}, Marwa Mamdouh^{*4}, Marwa Mohsen^{*5}

** Electronics and Electrical Communication Engineering Dept., Cairo University. Giza, 12613, Egypt*

¹doha.yousef@yahoo.com

²manar_ahmedsh@yahoo.com

³manaalezzat@yahoo.com

⁴marwa_mamdouh90@yahoo.com

⁵eng_marwa.m_2012@yahoo.com

Abstract—this project can be classified into educational category. The main topic is Arabic Handwritten on mobile sets. The project aims to provide easy tool for all users to learn the Arabic writing, therefore depending on the wide spreading mobile sets in the past few years. We decide to make this application on mobile sets of Android which have reasonable price for all users to be able to buy and use at any time. The application depends mainly on concepts of image processing where the operations are done on users online writing to be able to grade it and give him a feedback about it to regulate his writing. We try to make the application in number of gradual levels to provide a suitable tool for the user to learn gradually. The program is mainly divided into two separate categories, dividing the learning levels to lines, chars and words. Within each level the degree of difficulty increases gradually, e.g. lines level start with simple straight lines then curved lines and finally more complex lines and so on in the remaining levels as will be shown later. Targeted people are People who attend literacy classes. This application may be to follow up those people within the class or even after finishing the class this may be done if the application result is sent to server or may be used for those people to remain familiar with writing and Children who drop out of education. This can be done by using a game layout in designing the application to be attractive for children.

1 INTRODUCTION

A. Illiteracy in Egypt

According to a June report issued by the Council of Ministers' Information and Decision Support Centre, nearly 27% of Egypt's 85 million citizens are illiterate. In addition, the female illiteracy rate is even worse -- some 20% higher than among males, particularly in the 15 to 35 age group.

B. Literacy Program in Egypt:

1) Using teachers (traditional methods):

A program consists of three stages of a period of 9 months 3 months each stage. According Search Literacy Program in most areas teaches 20 people in 9 months and this needs a lot of resources such as classes, teachers and money to get the tools like books, blackboards and chalk.

2) Use of computer in literacy

The project aims to improve the capabilities of the targeted literacy programs by teaching the literacy curriculum using technology CDs.

3) Use of mobile in literacy:

Since the objective of this project is to find a cheap tool which consistently available with the learner. We have found that mobile is an effective way to do this role as it helps to continue the educational process even after the end of the basic program of literacy. We also found that the application of testing the literacy program on Mobile take less time than traditional methods

With the help of a lot of companies we can provide promotional offers to learners, whether to complete the basic program for literacy, or maintain constantly what they have learned or to improve their level in the house.

Using mobile in literacy is a kind of mobile learning or "M-Learning". Mobile learning has attracted the interest of educators, researchers, and companies developing learning systems and instructional materials.

2 PATTERN RECOGNITION SYSTEM

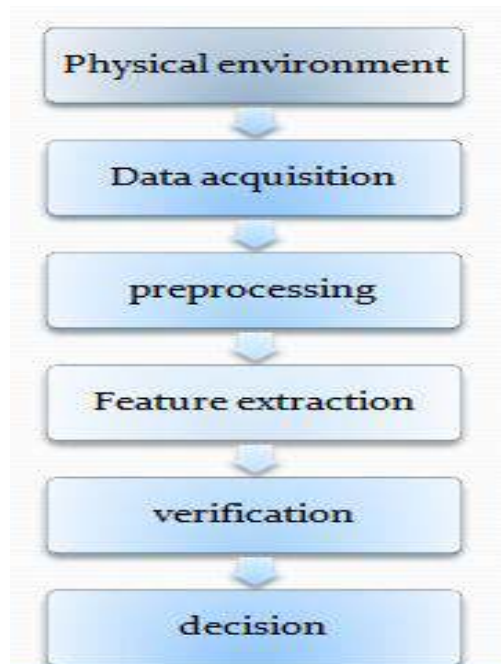


Figure 1: W Object/process diagram of a pattern recognition system.

A. *Data acquisition and sensing:*

- Measurements of physical variables
- Important issues: bandwidth, resolution, sensitivity, distortion, SNR, latency, etc.

B. *Pre-processing:*

- Removal of noise in data
- Isolation of patterns of interest from the background.

C. *Feature extraction:*

- Finding a new representation in terms of features

D. *Model learning and estimation:*

- Learning a mapping between features and pattern groups and categories.

E. *Classification:*

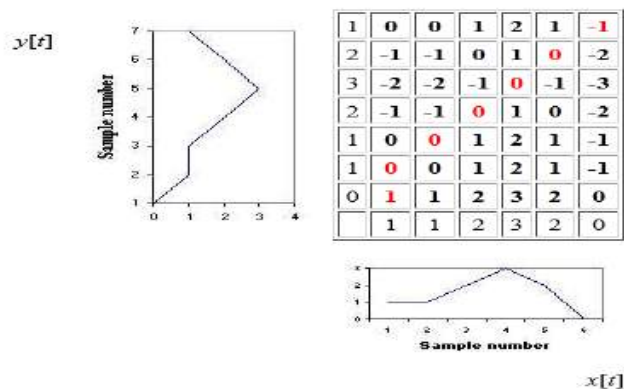
- Using features and learned models to assign a pattern to a category.

F. *Post-processing:*

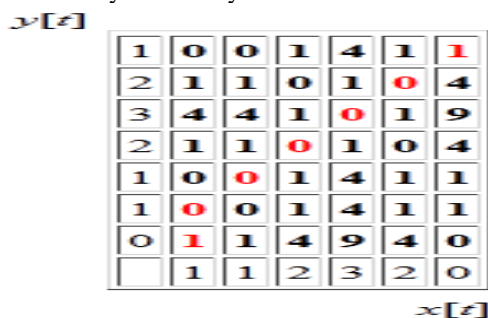
- Evaluation of confidence in decisions.
- Exploitation of context to improve performance and Combination of experts.

3 DYNAMIC TIME WARPING

The basic problem that DTW attempts to solve is how to align two sequences in order to generate the most representative distance measure of their overall difference. If you have two signals encoded as a sequence of evenly spaced values (representing, for example, the peak frequency of the signals), then an obvious way to compare the signals is to sum the differences in frequency at each point along the signals. However, a problem arises if there is any discrepancy in the alignment of the signals if for example one of the signals is stretched or compressed compared to the other the DTW algorithm uses a dynamic programming technique to solve this problem.



First calculate the difference between both signals (input $x[t]$ and reference signal $y[t]$), then there is a sequence of low numbers, close to the diagonal, indicating which samples of $x[t]$ are closest in value to those of $y[t]$. These are marked in red. Instead of a simple subtraction, it is customary to use a symmetrical distance measure, such as $(x[t] - y[t])^2$



4 APPLICATION STAGES

A) Offline:

- The application has two levels (lines and characters)
- Comparison is done offline, since the user writing is saved as an image then compare this image with the reference one stored in the memory.

1) Horizontal line:

We sum the pixels at all rows to get the line thickness and all Columns to get its length, and then evaluate the written line depend on the calculated:

- Length.
- Width (thickness).

Then, we give a tolerance to the user up and down around the original line of about $\frac{1}{4}$ the thickness of the original line. The written line far away the original line. If the user writes a correct line but it far away the original line his grade will decrease.



Figure 2: Horizontal line

2) *Vertical line:*

As made in horizontal line, we sum rows pixels and columns pixels to get the projection of the line in 2_D, but here we are interested in the sum of columns as it determines the line thickness to make sure that the written is straight line, then Evaluate the written line depend on the calculated:

- Length.
- Width.

The same as in horizontal line. The written line far away the original line.

3) *Diagonal line:*

In the diagonal line there are two regions to evaluate the written line, the first region is confined with the shown one with small tolerance , if the user 's written line within this region he will get high grade ,if he writes outside the grade will decrease gradually .

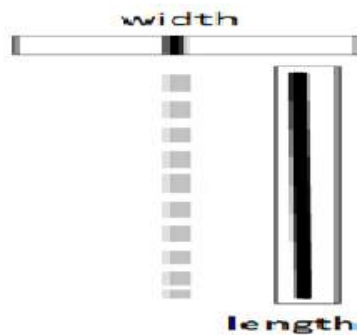


Figure 3: Vertical line

4) *Characters:*

We have image for each character, the users must follow the character as it is written on the screen and there are some restrictions so that we can evaluate their writing and give them feedback to improve their ability to write correctly:

- The users must pass on a number of important points which we choose according to each character and its properties like (shape and curvatures).
- The users must not write outside the boundaries, we put the boundaries to give the users area to write the character properly without restricting them, but still learning them the correct way for writing.

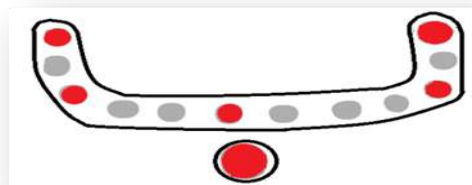


Figure 4: Example of boundaries



Figure 5: Example of hot points

B) *Online:*

At the application beginning a video will play to show the user how to hold a pen, different images will be shown to illustrate right and wrong methods while holding the pen, a skip button is available, if the user doesn't need to watch it, then the writing application starts through three main stages.

1) *Lines stage:*

In this stage the user will be trained to draw line to make sure that he is familiar with pen (stylus) this is done gradually through straight lines in first trail then curved lines and then more complex lines. This stage mainly is for assuring the user ability of holding pen and writing.

2) *Characters stage*

After lines stage the user can enter the characters level. In this stage a single character is displayed at a time for the user to follow (write above), this template character is stored as a reference and the user writing is compared with this reference. Dynamic Time Warping algorithm is used to perform the comparison task and then a grade is given to the user, this is repeated for different characters, and then the total stage grade is calculated and if accepted the user can pass to the next stage.

3) *Words stage*

In this stage different techniques are used:

- **Progressive animation**

In this stage the words are shown as strokes (stroke by stroke).the first stroke is animated and the user should follow it with the up motion the written stroke is saved and compared with the reference one, applying evaluation technique, if the written stroke is accepted, the next stroke is animated, if not the same stroke is animated again and so on until the word is completed, this stage is considered as a very important one as here the user is more constrained and will not pass a given stroke until he writes it relatively correct, so his ability in writing will be improved

- **Complete animation**

Here the whole word is animated and then the user follow it with the same manner as he sees in animation and then press submit to get the grade. Different words are used to train the user and if the whole stage score is accepted, he will pass to the free writing stage.

- **Free writing stage**

In this stage most of the constraints are removed, and the user has the ability to write freely, here we need to follow the written word but just write it in your way as you have trained in previous stages.

5 APPLICATION BUILDING FLOW

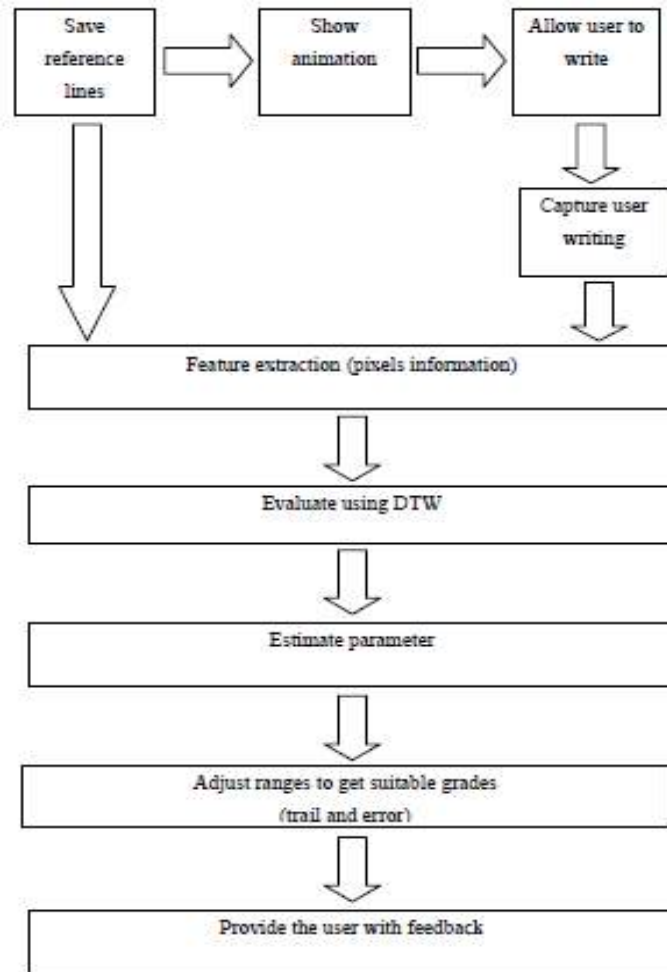


Figure 6: Application building flow

6 CHALLENGES

A) *Different speed between ref. and user*

- Removal of duplicated points.

B) No clear threshold for DTW result.

- Estimation of max accepted threshold.

C) In free writing: DTW is not efficient enough.

- Using more than one reference in verification.

7 HARDWARE FEATURES



Figure 7: Samsung Galaxy Pocket

- Android Operating System, version 2.3 (Gingerbread)
- Java MIDP emulator

8 SOFTWARE FEATURES

A) *Android*

Android is a software stack for mobile devices which means a reference to a set of system programs or a set of application programs that form a complete system. This software platform provides a foundation for applications just like a real working platform. The software stack is divided in four different layers, which include 5 different groups:

- 1) The application layer
- 2) The application framework
- 3) The libraries
- 4) The runtime
- 5) The kernel

As shown in Figure 8

B) *Java*

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets.

Android applications are developed using the Java language. As of now, that's really your only option for native applications. Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks, programming languages are only as powerful as their libraries. These libraries exist to help developers build applications

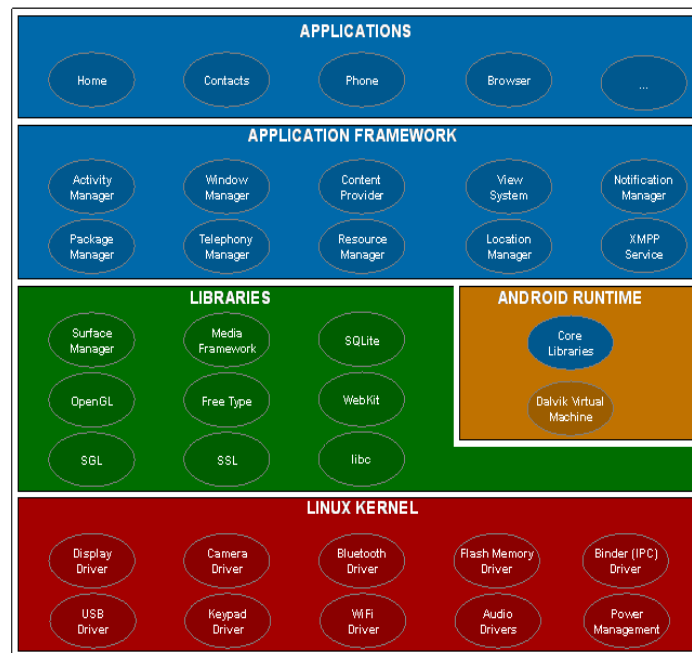


Figure 8: Major components of the Android operating system

9 CONCLUSION

- Using mobile sets in learning is a new trend called M-Learning, which will be very helpful and more attractive.
- Android mobile sets are very useful to be used for such applications as android is an open source operating system & cheap hand sets are available to satisfy our needs.
- Dynamic Time Warping is a very good algorithm which gives reasonable accuracy in our verification problem.
- Free writing stage needs more advanced algorithms & classifiers like HMM and SVM.

10 FUTURE WORK

- Improving evaluation technique to provide more convenient feedback to the users this can be done by using more advanced online features and more advanced classifiers like Hidden Markov Model (HMM).

HMM Background. ^[14]

It is sometimes useful to use HMMs in specific structures in order to facilitate learning and generalization. For example, even though a fully connected HMM could always be used if enough training data is available it is often useful to constrain the model by not allowing arbitrary state transitions. In the same way it can be beneficial to embed the HMM into a greater structure; which, theoretically, may not be able to solve any other problems than the basic HMM but can solve some problems more efficiently when it comes to the amount of training data required.

- Adding more letters and words to generalize the application. As the current application is a prototype to represent the idea and we use samples of letters and words.
- Adding more exercises preferable to be in a game form to be more attractive.

REFERENCES

- [1] *Selim Aksoy .” introduction to pattern recognition”, Department of Computer Engineering Bilkent University ,CS 551, Spring 2012*
- [2]http://alshorfa.com/en_GB/articles/meii/features/main/2011/08/05/feature-01(accessed 8 July 2012)
- [3]http://www.iiz-dvv.de/index.php?article_id=208&clang=1(accessed July 2012)
- [4]http://www.gsmarena.com/huawei_u8180_ideos_x14204.phphttp://en.wikipedia.org/wiki/Mobile_operating_system(accessed 8 July 2012)
- [5] *Joshua Dobbs “ Android application development for dummies”.*
- [6] *Samsung Electronics “S Pen SDK2.0.TutorialwithSample Code”.*
- [7] *Pavel Senin “Dynamic Time Warping Algorithm Review” , Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, senin@hawaii.edu ,December 2008.*
- [8][http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)) (accessed 8 July 2012)
- [9]http://www.gsmarena.com/samsung_galaxy_pocket_s5300-4612.php(accessed 8 July 2012)
- [10]<http://developer.android.com/about/versions/android-2.3-highlights.html>(accessed 8 July 2012)
- [11]http://en.wikipedia.org/wiki/Dynamic_time_warping(accessed 8 July 2012)
- [12]http://www.phon.ox.ac.uk/jcoleman/old_SLP/Lecture_5/DTW_explanation.html(accessed 8 July 2012)
- [13]http://awww.net/edu_subjects.html(accessed 8 July 2012)
- [14] http://en.wikipedia.org/wiki/Hierarchical_hidden_Markov_model (accessed 8 July 2012)
- [15]<http://mobile.tutsplus.com/tutorials/android/java-tutorial/>(accessed 8 July 2012)
- [16]<http://sip.clarku.edu/tutorials/java/java.html>(accessed 8 July 2012)
- [17]<http://developer.android.com/about/versions/android-2.2-highlights.html>(accessed 8 July 2012)

Association Rule-based Web Document Clustering

Noha Negm ^{*1}, Passent Elkafrawy ^{*2}, Abd-Elbadeeh M. Salem ^{**3}

** Faculty of Science, Menoufia University*

Shebin El-Kom, EGYPT

¹Noha_negm@yahoo.com

²passentmk@gmail.com

Faculty of Computers and Information, Ain Shams University

Cairo, EGYPT

³Abmsalem@yahoo.com

Abstract— Document Clustering is one of the main themes in text mining. It refers to the process of grouping documents with similar contents or topics into clusters to improve both availability and reliability of text mining applications. Some of the recent algorithms address the problem of high dimensionality of the text by using frequent term sets for clustering. In this paper, a novel approach for document clustering based on Association Rule Mining has been introduced; it provides a natural way of reducing a large dimensionality of the document vector space. A new algorithm for effective mining has been presented. Our approach consists of three phases: the text preprocessing phase, the association rule mining phase, and the document clustering phase. To overcome the drawbacks of Apriori algorithm, an efficient Hash-based Association Rule Mining in Text (HARMT) algorithm is presented. The generated association rules are used for obtaining the partition, and grouping the partition that have the same documents. Furthermore, the resultant clusters are effectively obtained by grouping the partition by means of derived keywords. Our approach can reduce the dimension of the text efficiently for very large text documents, thus it can improve the accuracy and speed of the clustering algorithm.

1 INTRODUCTION

Document cluster is a set of similar documents and automatic grouping of text documents is called Document Clustering. The documents within a cluster have high similarity in comparison to one another but are dissimilar to documents in other clusters. Document clustering has been studied intensively because of its wide applicability in areas such as Web mining, Search Engines, Information Retrieval, and Topological Analysis. Unlike in document classification, in document clustering no labeled documents are provided.

The problem of document clustering is generally defined as follows [1]: Given a set of documents, would like to partition them into a predetermined or an automatically derived number of clusters, such that the documents assigned to each cluster are more similar to each other than the documents assigned to different clusters. Documents are represented using the vector space model, which treats a document as a bag of words [2]. A major characteristic of document clustering algorithms is the high dimensionality of the feature space, which imposes a big challenge to the performance of clustering algorithms. They could not work efficiently in high dimensional feature spaces due to the inherent sparseness of the data. Next challenge is that not all features are important for document clustering, some of the features may be redundant or irrelevant and some may even misguide the clustering result [3].

Clustering algorithms are mainly categorized into hierarchical and partitioning methods [4-9]. K-means and its variants are the most well-known partitioning methods that create a flat, non-hierarchical clustering consisting of k clusters. The bisecting k-means algorithm first selects a cluster to split, and then employs basic k-means to create two sub-clusters, repeating these two steps until the desired number k of clusters is reached [10]. Steinbach in [5] showed that the bisecting k-means algorithm outperforms basic k-means as well as agglomerative hierarchical clustering in terms of accuracy and efficiency.

A hierarchical clustering method works by grouping data objects into a tree of clusters. These methods can further be classified into agglomerative and divisive hierarchical clustering depending on whether the hierarchical decomposition is formed in a bottom-up or top down fashion [11]. Steinbach in [12] showed that Unweighted Pair Group Method with Arithmetic Mean (UPGMA) is the most accurate one in agglomerative category.

Both hierarchical and partitioning methods do not really address the problem of high dimensionality in document clustering. Frequent itemset-based clustering method is shown to be a promising approach for high dimensionality clustering in recent literature [12-27]. It reduces the dimension of a vector space by using only frequent itemsets for clustering.

The concept of frequent itemsets originates from association rule mining [13] which uses frequent itemsets to find association rules of items in large transactional databases. A frequent itemsets is a set of frequent items, which co-occur in transactions

more than a given threshold value called minimum support. Recent studies on frequent itemsets in text mining fall into two categories. One is to use association rules to conduct text categorization [14, 15] and the other one is to use frequent itemsets for text clustering [12-27].

In this paper, we introduce a novel approach for document clustering based on association rules mining instead of frequent term sets. Such association rules are efficiently generated by using our new mining algorithm. The time is a critical factor in the mining and clustering process. Consequently a novel HARMT algorithm in the mining process is presented to overcome the drawbacks of the Apriori algorithm. The generated association rules are used for obtaining the partition, and grouping the partition that have the same documents. Furthermore, the resultant clusters are effectively obtained by grouping the partition by means of derived keywords.

The rest of this paper is organized as follows: Section 2 represents the review of literature in the field; Section 3 presents our proposed approach for document clustering. The experimental results are represented in Section 4, and Section 5 outlines the conclusions and future direction to the document clustering.

2 BACKGROUND

Depending on the use of frequent term set-based clustering method, various researchers put their efforts in order to solve the problem of high dimensionality, scalability, and accuracy. A brief review of some recent researches related to frequent terms-based text clustering is presented here:

SuffixTree Clustering in [16] considered the earlier work in this field. Its idea is to form clusters of documents sharing common terms or phrases (multi-word terms). Basic clusters are sets of documents containing a single given term. A cluster graph is built with nodes representing basic clusters and edges representing an overlap of at least 50% between the two associated basic clusters. A cluster is defined as a connected component in this cluster graph. The drawback of SuffixTree Clustering is that, while two directly neighboring basic clusters in the graph must be similar, two distant nodes (basic clusters) within a connected component do not have to be similar at all. Unfortunately, SuffixTree Clustering has not been evaluated on standard test data sets so that its performance can hardly be compared with other methods.

A new criterion for clustering transactions using frequent itemsets is presented in [12]. In principle, this method can also be applied to document clustering by treating a document as a transaction; however, the method does not create a hierarchy for browsing. The FTC and HFTC are proposed in [17].

The basic motivation of FTC is to produce document clusters with overlaps as few as possible. FTC works in a bottom-up fashion. Starting with an empty set, it continues selecting one more element from the set of remaining frequent itemsets until the entire document collection is contained in the cover of the set of all chosen frequent itemsets. In each step, FTC selects one of the remaining frequent itemsets which has a cover with minimum overlap with the other cluster candidates.

The documents covered by the selected frequent itemsets are removed from the collection, and in the next iteration, the overlap for all remaining cluster candidates is recomputed with respect to the reduced collection. In FTC, a cluster candidate is represented by a frequent itemsets and the documents in which the frequent itemsets occur. It calculates each candidate's EO which is decided by occurrence distribution of the candidates' documents. Thus, FTC tends to select cluster candidate, of which its number of documents is small while occurrence frequencies of these documents are large, as document cluster. However, it will cause large amount clusters with only one document, i.e. isolated documents.

As HFTC greedily picks up the next frequent itemset to minimize the overlapping of the documents that contain both the itemset and some remaining itemsets. The clustering result depends on the order of picking up itemsets, which in turn depends on the greedy heuristic used. The drawback of the HFTC algorithm is that it is not scalable for large document collections.

The FIHC algorithm is proposed in [18]; the FIHC measures the cohesiveness of a cluster directly using frequent itemsets. Two kinds of frequent item are defined in FIHC: global frequent item and cluster frequent item. FIHC develops four phases to produce document cluster: finding global frequent itemsets, initial clustering, tree construction, and pruning.

FIHC is based on cluster profile, not pair wise similarity used in classic clustering method. FIHC provides a tree for document clusters which is easy to browse with meaningful cluster description. Its characteristics of scalability and non-sensitivity to parameters are desirable properties for clustering analysis. However, FIHC has three disadvantages in practical application: first, it cannot solve cluster conflict when assigning documents to clusters. That is, a document may be partitioned into different clusters and this partition has great influence on the final clusters produced by FIHC. Second, after a document has been assigned to a cluster, the cluster frequent items were changed and FIHC does not consider this change in afterward overlapping measure. Third, in FIHC, frequent itemsets is used merely in constructing initial clusters. FIHC other processes in FIHC, such as making clusters disjoint and pruning, are based on single items of documents and decided by initial clusters.

In [19] the text-clustering algorithm known as Frequent Term Set-based Clustering (FTSC) is introduced. FTSC algorithm used the frequent feature terms as candidate set and does not cluster document vectors with high dimensions directly. Initially, it extracts significant information from documents and put it into databases. Later, it employed the Apriori to mine the frequent itemsets. At last, it clusters the documents as per the frequent words in subsets of the frequent term sets. At the same time, the subset of a frequent feature term set corresponds to a document category, which can provide more accurate description for clustering class. The results of the clustering texts by FTSC algorithm cannot reflect the overlap of text classes. But FTSC and FTSHC algorithms are comparatively more efficient than K-Means algorithm in the clustering performance.

Clustering based on Frequent Word Sequence (CFWS) is proposed in [20]. CFWS uses frequent word sequence and K-mismatch for document clustering. The difference between word sequence and word itemset is that word sequence considers words' order while word itemsets ignores words' order. The word's order is very important in word sequence than word itemset. By using the CFWS there are overlaps in the final clusters. With K-mismatch, frequent sequences of candidate clusters are used to produce final clusters.

Document Clustering Based on Maximal Frequent Sequences (CMS) is proposed in [21]. The basic idea of CMS is to use Maximal Frequent Sequences (MFS) of words as features in Vector Space Model (VSM) for document representation and then K-means is employed to group documents into clusters. CMS is rather a method concerning feature selection in document clustering than a specific clustering method. Its performance completely depends on the effectiveness of using MFS for document representation in clustering, and the effectiveness of K-means.

Frequent Itemset-based Clustering with Window (FICW) method is presented in [22], which employed the semantic information for text clustering with a window constraint. The experimental results obtained from three (hypertext) text sets revealed that FICW performed better in terms of both clustering accuracy and efficiency.

A simple hybrid algorithm (SHDC) is presented in [23] on the basis of top-k frequent term sets and k-means so as to overcome the main challenges of current web document clustering. Top-k frequent term sets were employed to provide k initial means, which were regarded as initial clusters and later refined by k-means. The final optimal clustering was returned by k-means whereas the clear description of clustering was given by k frequent term sets. Experimental results on two public datasets showed that SHDC performed better other two representative clustering algorithms both on efficiency and effectiveness.

A web-text clustering method for personalized e-learning based on maximal frequent itemsets is introduced in [24]. In the beginning, the web documents were represented by vector space model. Later, maximal frequent word sets were determined. In the end, on the basis of a new similarity measure of itemsets, maximal itemsets were employed for clustering documents. Experimental results proved that the presented method was efficient.

A frequent term based parallel clustering algorithm which could be employed to cluster short documents in very large text database is presented in [25]. A semantic classification method is also employed to enhance the accuracy of clustering. The experimental analysis proved that the algorithm was more precise and efficient than other clustering algorithms when clustering large scale short documents. In addition, the algorithm has good scalability and also could be employed to process huge data.

The document clustering algorithm on the basis of frequent term sets is proposed in [26]. Initially, documents were denoted as per the Vector Space Model and every term is sorted in accordance with their relative frequency. Then frequent term sets can be mined using frequent-pattern growth (FP growth). Lastly, documents were clustered on the basis of these frequent term sets. The approach was efficient for very large databases, and gave a clear explanation of the determined clusters by their frequent term sets. The efficiency and suitability of the proposed algorithm has been demonstrated with the aid of experimental results.

A clustering algorithm for discovering and unfolding the topics included in a text collection is proposed in [27]. The algorithm depended on the most probable term pairs generated from the collection and also on the estimation of the topic homogeneity related to these pairs. Topics and their descriptions were produced from those term pairs whose support sets were homogeneous for denoting collection topics. The obtained experimental results over three benchmark text collections showed the efficacy and usefulness of the approach.

In [28] a hierarchical clustering algorithm using closed frequent itemsets that use Wikipedia as an external knowledge to enhance the document representation is presented. Firstly, construct the initial clusters from the generalized closed frequent itemsets. Then used the two methods TF-IDF and Wikipedia as external knowledge, to remove the document duplication and construct the final clusters. The drawback in this approach is that it might not be of great use for datasets which do not have sufficient coverage in Wikipedia.

A Frequent Concept based Document Clustering (FCDC) algorithm is proposed in [29]. It utilizes the semantic relationship between words to create concepts. It exploits the WordNet ontology in turn to create low dimensional feature vector which allows us to develop an efficient clustering algorithm. It used a hierarchical approach to cluster text documents having common concepts. FCDC found more accurate, scalable and effective when compared with existing clustering algorithms like Bisecting K-means, UPGMA and FIHC.

A Wordset based Document Clustering algorithm for large datasets is proposed in [30]. WDC uses a wordsets based approach to build clusters. It first searches frequent closed wordsets by association rule mining and then form initial cluster of documents which each cluster representing single closed wordsets. Then the algorithm refines the initial clusters and makes final results as a clustering tree like representations. The idea is to do clustering of documents by using the wordsets that occur in sufficient number of documents. Each document in this approach corresponds to transaction and each word corresponds to an item as in association rule mining. WDC performs well in terms of quality of cluster form. Page Layout.

3 PROPOSED DOCUMENT CLUSTERING APPROACH

The proposed document clustering approach based on association rule mining is shown in Fig.1. The main characteristic of the approach is that it introduces a novel methodology for document clustering based on association rules between frequent termsets. Moreover it introduces a new mining algorithm for generating the association rules.

Our approach consists of three phases: 1) Text Preprocessing Phase includes: filtration, stemming, and indexing of documents, 2) Association Rule Mining Phase introduces a novel HARMT algorithm for generating association rules, and 3) Document Clustering Phase includes two main steps: the partitioning based on association rules and the clustering from the partitions.

A. Text Processing Phase

The text preprocessing phase begins after collecting the text documents that need to be clustered. The documents are filtered to eliminate unimportant words by using a list of stop words. After the filtration process, the system does word stemming that removes prefixes and suffixes of each word. Finally, the documents are automatically indexed based on the important words by using the weighting scheme.

1) *Filtering*: In this process, the documents are filtered by removing the unimportant words from documents content. Therefore, the unimportant words (noise words) get discarded or ignored (e.g. articles, pronouns, determiners, prepositions and conjunctions, common adverbs and non-informative verbs (e.g., be)) and more important or highly relevant words are single out. Furthermore, the special characters, parentheses, commas, and etc., are replaced with distance between words in the documents. This process is more critical in the case of formatted documents, such as web pages, where formatting tags can either be discarded or identified and their constituent terms attributed different weights.

2) *Stemming*: *Stemming is a technique that is commonly used to control the list of indexed words by removing the prefixes and suffixes of each word.* For example, the words “connected”, “connection”, “connections” all reduced to the stem “connect”. Stemmer reduces similar words to the same root and this has two positive effects: 1) the number of indexed terms is reduced because similar terms are mapped into one entry. 2) The relevancy of the results is often improved by a significant margin. Porter’s algorithm is the de facto standard stemming algorithm.

3) *Indexing*: Our approach automatically indexes documents by labeling each document by a set of the most important words with their frequencies. The techniques for automated production of indexes associated with documents usually rely on frequency-based weighting schema. The weighting schema is used to index documents and to select the most important words in all document collections [31].

The purpose of weighting schema is to reduce the relative importance of high frequency terms while giving a higher weight value for words that distinguish the documents in a collection. The weighting scheme TF-IDF (Term Frequency, Inverse Document Frequency) is used to assign higher weights to distinguished terms in a document. The weighting scheme includes the intuitive presumption that is: the more often a term occurs in a document, the more representative of the content of the document (term frequency). Moreover the more documents the term occurs in, the less discriminating it is (inverse document frequency).

$$w(i, j) = tfidf(d_i, t_j) = \begin{cases} Nd_{i,t_j} * \log_2 \frac{|C|}{Nt_j} & \text{if } Nd_{i,t_j} \geq 1 \\ 0 & \text{if } Nd_{i,t_j} = 0 \end{cases} \quad (1)$$

where $w(i,j) \geq 0$, N_{d_i,t_j} denotes the number the term t_j occurs in the document d_i (term frequency factor), N_{t_j} denotes the number of documents in collection C in which t_j occurs at least once (document frequency of the term t_j) and $|C|$ denotes the number of the documents in collection C. The first clause applies for words occurring in the document, whereas for words that do not appear ($N_{d_i,t_j} = 0$), we set $w(i,j) = 0$.

Once a weighting scheme has been selected, automated indexing can be performed by simply selecting the words that satisfy the given weight constraints for each document. The major advantage of an automated indexing procedure is that it reduces the cost of the indexing step.

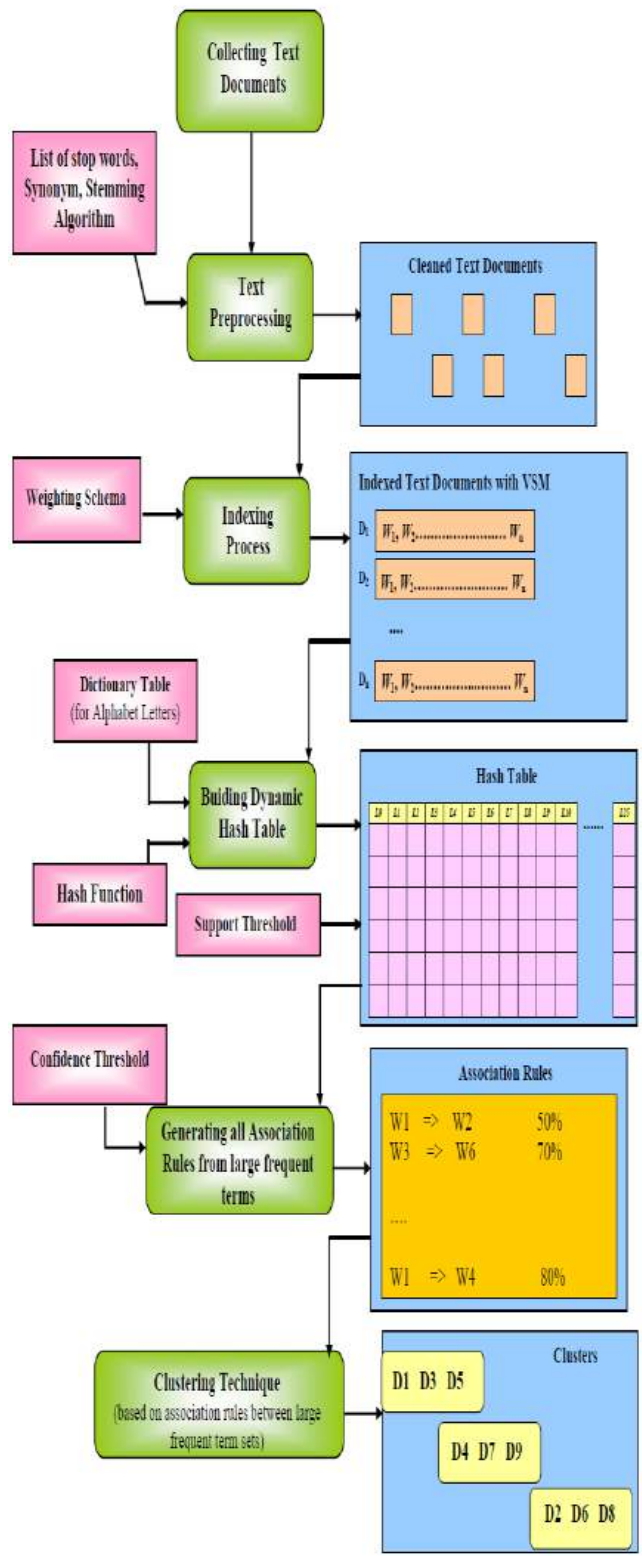


Figure 1: The proposed document clustering approach

1.1) The Association Rule Mining Phase

Association Rule Mining (ARM) is a promising field in text mining area. Such rules are very useful for users in different domains such as medical, News and stock to extract more useful information. In the case of text mining, extracted rules are deduced as co-occurrences of terms in texts and therefore are able to return semantic relations among the terms.

There are many algorithms have been developed for generating large frequent itemsets from datasets [13, 32]. Apriori algorithm considered to be the basic for all developed ARM algorithms. The drawbacks of the Apriori algorithm are: 1) Make multiple scanning on the original documents to generate frequent itemsets. As the documents become large, it gives worse performance. 2) It is time consuming to extract the association rules. Although the drawbacks of the Apriori algorithm, it still use for generating the frequent term sets that used in the document clustering.

1.1.1) The Proposed HARMT Algorithm

As we know the text documents have huge numbers of words which are very important, and dealing with words is difference from items. Moreover the time is a critical factor in the mining process. Consequently we proposed a novel Hash-based Association Rule Mining in Text (HARMT) algorithm to reflect on all these factors.

HARMT algorithm overcomes the drawbacks of the Apriori algorithm by employing the power of data structure called Dynamic Hash Table. Moreover it used new methodology for generating frequent term sets by building the hash table during the scan of documents only one time.

The two main key factors in hash table building and search performance are: 1) the number of the English alphabet letters (A to Z) $N = 26$, and 2) the hashing function $h(v) = v \bmod N$. The flowchart of the proposed HARMT algorithm is shown in Fig. 2. The HARMT algorithm employs the following two main steps:

- 1) Since every English word can begin with any alphabet letter from A to Z, subsequently we construct the dictionary table based on the number of alphabet letters $N = 26$ and give each character a numeric number from 0 to 25, and
- 2) There are also two main processes for a dynamic hash table: a) the Building Process, and b) the Scanning Process.

3.1.1.1) The Building Process

In the dynamic hash table, a primary bucket is only built at the first. Its size is equal to the number of the English alphabet letters N . Their locations in the hash table are determined using the division method of hash function that is $h(v) = v \bmod N$. For example, the alphabet letter E takes the numeric number 4 in the dictionary table, and their location is determined by applying the hash function so that its location is also 4 and so on.

3.1.1.2) The Scanning Process

After building a primary bucket, each document is scanned only once as follows:

- 1) For each document, select all terms and make all possible combinations of concepts then determine their locations in the dynamic hash table using the hash function $h(v)$.
- 2) Insert all terms and term sets in a hash table and update their frequencies, the process continues until there is no document in the collection.
- 3) Save the dynamic hash table into secondary storage media.
- 4) Scan the dynamic hash table to determine the large frequent term sets that satisfy the threshold support and generate all association rules.

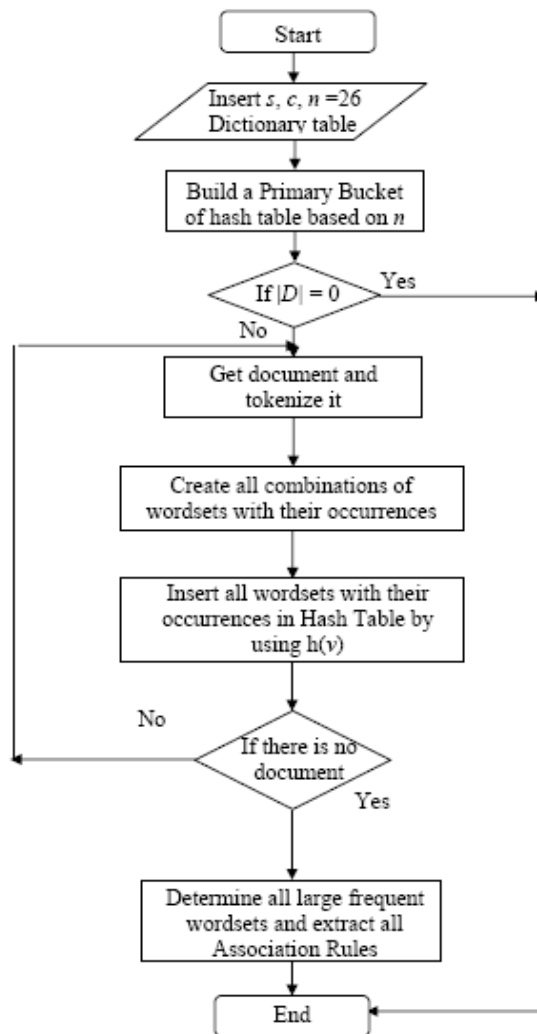


Figure 2: The flowchart of the HARMT algorithm

3.1.2) The Advantages of the HARMT algorithm

The advantages of the HARMT algorithm summarized as follows:

1) Reduce the Scanning Process on the Documents:

The algorithm scans the collection of documents only on time to build the dynamic hash table without returning to the original documents so that it takes $O(m*n)$ time since m is the number of words and n is the number of documents.

2) Small Size of Dynamic Hash Table:

It is not dependent on the number of words in the building process. As the number of words increase, the size of primary bucket of the dynamic hash table will not change.

3) Saving the Memory Space:

The HARMT algorithm does not require the predetermination of hash table size in advance before the creation process. Therefore there is no term sets with zero occurrence will occupy a size in a dynamic hash table.

4) Improve the Execution Time:

The algorithm permits the end user to change the threshold support and confidence factor without re-scanning the original documents: Since the algorithm saves the dynamic hash table into secondary storage media.

3.2) The Document Clustering Phase

Here, we have proposed an effective approach for clustering a text corpus with the aid of association rules between the large frequent term sets.

The proposed approach consists of the following major two steps: 1) Based on the strong association rules, the text documents are partitioned, and 2) Clustering of text documents from the partitions by means of represented words.

3.2.1) Partitioning the Text Documents based on the Association Rules

All strong association rules generated from the HARMT algorithm are used as input to this step. Strong association rules mean that all association rules that satisfies the confidence threshold.

The methodology of partitioning of text documents based on the generated association rules is as follows: Initially, we sort the set of generated association rules in descending order in accordance with their confidence level. Secondly, the first association rule from the sorted list is selected. Subsequently, an initial partition P1 which contains all the documents including the both term sets is constructed. Then we take the second association rule whose confidence is less than the previous one to form a new partition P2. This partition is formed by the same way of the partition P1. This procedure is repeated until every association rules are moved into partition P(i). Finally, all partitions that share the same documents are grouped in one partition.

3.2.2) Clustering of Text Documents from the Partition

In this step, we first identify the documents and the words $K_d[D_{c(i)}^{(x)}]$ that used for constructing each partition P(i). The words are obtained from all association rules that combined in one partition. Subsequently the support of each unique word is computed within the partition.

The set of words satisfying the partition support (par_sup) are formed as representative words $R_w[c(i)]$ of the partition P(i). Subsequently, we find the similarity of the partitions with respect to the representative words. The definition of the similarity measure plays an importance role in obtaining effective and meaningful clusters. The similarity between two partitions S_m is computed as follows [11],

$$S(K_d[D_{c(i)}^{(x)}], R_w[c(i)]) = |K_d[D_{c(i)}^{(x)}] \cap R_w[c(i)]| \quad (2)$$

$$S_m(K_d[D_{c(i)}^{(x)}], R_w[c(i)]) = \frac{S(K_d[D_{c(i)}^{(x)}], R_w[c(i)])}{|R_w[c(i)]|} \quad (3)$$

Based on the similarity measure, a new cluster is formed from the partitions i.e. each cluster will contain all partitions that have the similar similarity measures.

4 EXPERIMENTAL RESULTS

For experimentation with our dataset, we take 11 documents from various topics namely, Medical (D1 to D3), Sports (D4, D5), Association rule mining (D6 to D8) and Economics (D9 to D11). After the preprocessing and indexing process, each document is indexed by a set of weighted words. The total number of words in the collection of documents is 38 words without occurrence. The large frequent term sets are mined from the hash table of varying length from 1 to 4 (which satisfy minimum support 30%).

Then we get all association rules from only 2- large frequent term sets and select only the strong ones (which satisfying the minimum confidence 50%) and sort the rules in descending order. Subsequently, initial partition is constructed using these strong association rules shown in Table 1.

After that, the support of the representative words of each partition is computed. The similarity measure as shown in Table 2 is calculated for each partition by means of derived keywords

TABLE I
GENERATED PARTITIONS OF TEXT DOCUMENTS

Partition	Text Documents
P ₁	D ₁ , D ₂ , D ₃
P ₂	D ₁ , D ₃
P ₃	D ₆ , D ₇ , D ₈
P ₄	D ₉ , D ₁₀ , D ₁₁
P ₅	D ₆
P ₆	D ₄ , D ₅
P ₇	D ₆ , D ₈
P ₈	D ₉ , D ₁₁
P ₉	D ₄
P ₁₀	D ₁ , D ₂

P ₁₁	D ₅
P ₁₂	D ₆ , D ₇

From Table 2, we noticed that there are more than one partition have the same similarity measures such as P₁, P₂, and P₁₀ have similarity value equal to 0.55 and so on. Finally, all partitions that have the similar similarity measure are grouped. So we get four clusters (C₁, C₂, C₃, and C₄) from twelve partitions without overlapping as shown in Table 3.

TABLE 2:

SIMILARITY MEASURE OF TEXT DOCUMENTS

Partition	Text Documents	Similarity measures
P ₁	D ₁ , D ₂ , D ₃	0.55
P ₂	D ₁ , D ₃	0.55
P ₃	D ₆ , D ₇ , D ₈	0.84
P ₄	D ₉ , D ₁₀ , D ₁₁	1.0
P ₅	D ₆	0.84
P ₆	D ₄ , D ₅	0.3
P ₇	D ₆ , D ₈	0.84
P ₈	D ₉ , D ₁₁	1.0
P ₉	D ₄	0.3
P ₁₀	D ₁ , D ₂	0.55
P ₁₁	D ₅	0.3
P ₁₂	D ₆ , D ₇	0.84

TABLE 3:

RESULTANT CLUSTER

Cluster	Partition	Text Documents
C ₁	P ₁ , P ₂ , P ₁₀	D ₁ , D ₂ , D ₃
C ₂	P ₃ , P ₅ , P ₇ , P ₁₂	D ₄ , D ₅
C ₃	P ₄ , P ₈	D ₆ , D ₇ , D ₈
C ₄	P ₆ , P ₉ , P ₁₁	D ₉ , D ₁₀ , D ₁₁

5 CONCLUSIONS

In this paper, we presented a novel approach for document clustering based on association rules. The novelty of this approach is that: Firstly, introduce an efficient HARMT algorithm for generating association rule mining. The algorithm introduced a novel methodology for generating frequent term sets by scanning the documents only one time and stores all termsets and their combinations in dynamic hash table.

Furthermore it provides a possibility to generate more than frequent termsets at different minimum support without needing to rescan the documents. It speeds up the mining process. Secondly, it exploits association rules for defining a partition, organizing the partitions and clustering them by means of representative words furthermore reducing the dimensionality of document sets. Thirdly, the using of HARMT algorithm in the mining process speeds up the clustering process.

The experimental evaluation on text data sets demonstrated that our approach outperforms in terms of accuracy, efficiency, and scalability. Moreover, it automatically generates a natural description for the generated clusters by a set of association rules.

We are currently in the stage of implementing and optimizing the performance of the approach by C#.net language to compare it with the other approaches. Moreover we intend to carry out extensive testing of applying our approach.

REFERENCES

- [1] K. Raja, C. Narayanan, "Clustering Technique with Feature Selection for Text Documents", *International Conference on Information Science and Applications ICISA*, 2010, pp.506-514.
- [2] A. Luiz, V. Ana, and M. Marco, "A Simple and Fast Term Selection Procedure for Text Clustering", *International Conference on Intelligent Systems Design and Applications*, 2007, pp. 777 - 781
- [3] S. Fabrizio, "Machine Learning in Automated Text Categorization", *International Conference on ACM Computing Surveys*, Vol. 34, No. 1, 2002, pp.
- [4] K. Jain, N. Murty, and J. Flynn, "Data clustering: a review", *International Conference on ACM Computing Surveys*, Vol. 31, No. 3, 1999, pp. 264-323.

- [5] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques", *KDD Workshop on Text Mining*, 2000Tavel, P. 2007 Modeling and Simulation Design. AK Peters Lid.
- [6] P. Berkhin, Survey of clustering data mining techniques, 2004, Site: http://www.accrue.com/products/rp_cluster_review.pdf.
- [7] http://www.accrue.com/products/rp_cluster_review.pdf.
- [8] X. Rui, "Survey of Clustering Algorithms", *International Conference of IEEE Transactions on Neural Networks*, Vol.15, No. 3, 2005, pp. 634-678.
- [9] M. W. Berry Editor, *Survey of Text Mining: Clustering, Classification, and Retrieval*, Springer-Verlag New York, Inc., 2004.
- [10] C. Chen, F. tseng and T. Liang, "Hierarchical Document Clustering using Fuzzy Association Rule Mining", *International Conference on Innovative Computing Information and Control, IEEE*, 2008.
- [11] F. Benjamin, W. Ke, and E Martin., *Hierarchical Document Clustering*, Simon Fraser University, Canada, 2005.
- [12] J. Ashish, J. Nitin, "Hierarchical Document Clustering: A Review", *2nd National Conference on Information and Communication Technology*, 2011, Proceedings published in International Journal of Computer Applications.
- [13] B. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets", *International Conference on Data Mining*, 2003, pp. 59-70
- [14] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", *International Conference on Management of Data*, Washington, 1993, pp. 207-216.
- [15] O. Zaiane and M. Antonie, "Classifying text documents by association terms with text categories", *International Conference of Australasian Database*, pp. 215-222, 2002.
- [16] B. Liu, W. Hsu and Y. Ma, "Integrating classification and association rule mining", *International Conference of ACM SIGKDD on Knowledge Discovery and Data Mining*, pp. 27-31, 1998.
- [17] O. Zamir and O. Etzioni, "Web document Clustering: A Feasibility Demonstration", *International Conference of ACM SIGIR 98*, 1998, pp. 46-54.
- [18] M. Beil, and X. Xu, "Frequent term-based text clustering", *International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 436- 442
- [19] M. Hassan and K. John, "High Quality, Efficient Hierarchical Document Clustering using Closed Interesting Itemsets", *International IEEE Conference on Data Mining*, 2006.
- [20] L. Xiangwei, H. Pilian, *A study on text clustering algorithms based on frequent term sets*, Springer-Verlag Berlin Heidelberg, 2005.
- [21] Y.J. Li, S.M. Chung, J.D. Holt, "Text document clustering based on frequent word meaning sequences", *Data & Knowledge Engineering*, Vol. 64, 2008, pp. 381–404.
- [22] H. Edith, A.G. Rene, J.A. Carrasco-Ochoa, and J.F. Martinez-Trinidad, "Document Clustering based on Maximal Frequent Sequence", *FinTal 2006*, LNAI, Vol. 4139, 2006, pp. 257-267.
- [23] Z. Chong, L. Yansheng, Z. Lei and H. Rong, "FICW: Frequent itemset based text clustering with window constraint", *Wuhan University journal of natural sciences*, Vol. 11, No. 5, pp. 1345-1351, 2006.
- [24] L. Wang, L. Tian, Y. Jia and W. Han, "A Hybrid Algorithm for Web Document Clustering Based on Frequent Term Sets and k-Means", *Lecture Notes in Computer Science*, Springer Berlin, Vol. 4537, pp. 198-203, 2010.
- [25] Z. Su, W. Song, M. Lin, and J. Li, "Web Text Clustering for Personalized E-Learning based on Maximal Frequent Itemsets", *International Conference on Computer Science and Software Engineering*, Vol. 06, pp. 452-455, 2008.
- [26] Y. Wang, Y. Jia and S. Yang, "Short Documents Clustering in very Large Text Databases", *Lecture Notes in Computer Science*, Springer Berlin, Vol. 4256, pp. 38-93, 2006.
- [27] W. Liu and X. Zheng, "Documents Clustering based on Frequent Term Sets", *Intelligent Systems and Control*, 2005.
- [28] H. Anaya, A. Pons and R. Berlanga, "A document clustering algorithm for discovering and describing topics", *Pattern Recognition Letters*, Vol. 31, No. 6, pp. 502-510, April 2010.
- [29] R. Kiran, S. Ravi, and p. Vikram, *Frequent Itemset Based Hierarchical Document Clustering Using Wikipedia as External Knowledge*, Springer-Verlag Berlin Heidelberg, 2010.
- [30] R. Baghel and Dr. R. Dhir, "A Frequent Concept Based Document Clustering Algorithm", *International Journal of Computer Applications*, Vol. 4, No. 5, 2010.
- [31] A. Sharma and R. Dhir, "A Wordsets based Document Clustering Algorithm for Large datasets", *International Conference on methods and Models in Computer Science*, 2009.
- [32] M. Berry, *Survey of Text Mining: Clustering Classification, and Retrieval*, Springer-Verlag New York, Inc., 2004.
- [33] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *International Conference of very Large Data Bases*, 1994, pp. 487-499.

Adaptive English Pronunciation Errors for Arab Learners of English

Hassanin Al-Barhamtoshy, Kamal Jambi, Wajdi Al-Jedaibi
Faculty of Computing & Information Technology
King Abdulaziz University, Saudi Arabia
(hassanin , kjambi, waljedaibi)@kau.edu.sa

Sherif Abdou¹, Mohsen Rashwan²
1: Faculty of Computers at Cairo University Egypt
2: Faculty of Engineering at Cairo University Egypt
(sheriff.abdou , mrashwan)@rdi-eg.com

Abstract - The paper introduces the problems of detecting and correcting accent defects, and also summarizes typical Arabic speakers in English spoken language. Then, it introduces the essential probabilistic architecture that will be used to solve both accent defects and pronunciation problems.

Consequently, some of important kinds of variations in pronunciation that are important for speech recognition are carried out and presents accent defects describing such variations. Therefore, high quality speech algorithms need to know when to use particular pronunciation variants. Also, the paper focuses on the essential programming algorithm and various instantiations and also a probabilistic of artificial neural network, HMM, or weighted augmented transition network called *weighted ATN*.

The proposed method improves the accuracy of correct speaker. Using the *SpeakCorrect*, 100 hours from speakers phonetically prerecorded are used to create pronunciation training database. HMM and the weighted ATN models are trained using such prerecorded sets. Their speeches are very clear, acceptable as natural speeches. The proposed framework is optimized to suit embedded the phonetic pronounced database. This proposed frame work has been tested for Saudi and Egyptian accents.

This paper is a practical reference guide for teaching of English as a foreign language (Saudi and Egyptian accents). It is used to help learners to anticipate the characteristic difficulties of English who speak Arabic mother tongues, and how these difficulties overcome.

1. INTRODUCTION AND KEY DEFINITIONS

Any speech engine is composed of two or three sub-systems; the first is word recognizer that converts any given utterance into its underlying sequence of words. The second is phoneme recognizer based on HTK, which converts the given utterance into a sequence of phonemes. This sequence is then processed using matching algorithm and its most important keywords are extracted.

Automated speech recognition and interaction with the user is an essential speech applications system. Voice-driven navigation devices and hands-free telephony are other applications where the user is enabled to enter telephone numbers by speaking sequences of digits, or to enter city names for navigation.

The acoustic input O is treated as sequence of individual “symbols” or “observations” (by slicing up the input every 10ms, and representing each slice by values or frequencies of that slice). Each index represents time interval for slices of the input, therefore; capital letters will stand for sequences of symbols and lower-case letters represents symbols: $O = o_1, o_2, o_3, \dots, o_t$

Similarly, sentence will be treated as string of words: $W = w_1, w_2, w_3, \dots, w_n$.
Some of general terminologies used throughout this document, are explained in this section, [1]-[4].

1.1 APPROACH AND METHODOLOGY

The paper of the project is prepared especially for participating non-specialist learner/student who needs to speak English correctly. Technical linguistic terminology has been kept to a minimum, and proposed system has in general aimed at producing clear simple descriptions of usage rather than detailed of evaluation in visual output. This is particularly the case in the area of pronunciation, where excessive technical detail can be confusing for the non specialist. Within this approach, however, we believe that the descriptions, the technical and the technology points given here are valid and reasonably comprehensive.

The activities of this paper involve designing, implementing, and testing a prototype system that can be delivered and can support the daily teaching-based activities of the adult students. This system is not intended to be a complete substitute for the human class teacher but will be an auxiliary tool to help him/her teach the basic skills of reading, and participate in the different fields of comprehensive development.

In order to achieve the propose system, research and development is needed in five main components:

1. Developing high quality multimedia based lessons and comprehension questions.
2. Developing a reading training module
3. Developing a recording training module
4. Developing educational evaluation module
5. Integrating the developed modules in a deliverable application prototype.

The project has been designed to keep things simple even when dealing with complexity. Following this approach a simple structure has been drawn that comprises the main activities of speech processing. Therefore the project will be structured in the following set of work packages:

- 1- Design the applications according to the user requirements
- 2- Test a retype before the final release
- 3- Get the experience of previous similar projects

1.2 KEY DEFINITIONS

Phoneme: the smallest unit of speech is a phoneme; it is used to distinguish meaning. The phoneme is the important unit in the word, each word consists of phonemes, and replacing them causes change in the meaning of a word. If the sound [b] is replaced by [p] in the word “pin”, the word changed to “bin”. Therefore /b/ is a phoneme [20].

Phone: It is the smallest physical segment of sound. And therefore, phones are the physical realization of phonemes. Also, **allophones** are described as phonic variety of phonemes [25].

Phonetics: is the study of human speech, and is concerned with properties of speech sounds.

Phonology: is used to study sound systems and to abstract sound units; i.e., phonemes and phonological rules. Therefore, phonetics definitions apply across languages, and phonology is language based. The phonetic of a sound represented by [20], and the phoneme is represented using //.

Syllable: is defined as a unit of pronunciation. It is generally larger than a single sound and smaller than a word. The *syllable* is start and end with mostly consonants, and is made up using vowels.

1.3. LITERATURE REVIEW AND RELATED WORKS

The paper of Macherey and et. al.; (2009) investigates two statistical methods for spoken language understanding based on statistical machine translation. The first approach employs the source-channel paradigm, whereas the other uses the maximum entropy framework. Starting with an annotated corpus, it describes the problem of translation from a source sentence to a target sentence. Also, it analyzes the quality of different alignment models and feature functions and shows that the direct maximum entropy approach outperforms the source channel-based method. Finally, it investigates a new approach to combine speech recognition and spoken language understanding. For this purpose, it employs minimum error rate training which directly optimizes the final evaluation criterion. Experiments were carried out on two German inhouse for spoken dialogue systems [1].

Computational semantics performs a conceptualization of the world for composing a meaning representation structure from available signs and their features present, for example, in words and sentences (De Mori and et. al., 2008), [2]. Spoken language understanding (SLU) is the interpretation of signs conveyed by a speech signal. SLU and natural language understanding (NLU) share the goal of obtaining a conceptual representation of natural language sentences. Signs are used for interpretation and can be coded into signals along with other information such as speaker identity. Furthermore, SLU systems contain automatic speech recognition (ASR) component and must be robust to noise due to the nature of spoken language and the errors introduced by ASR.

Dinarelli and et. al.; (2010) presents a call routing application for complex problem solving tasks. Up to date work on call routing has been mainly dealing with call-type classification, [3]. In this paper they take call routing further: Initial call classification is done in parallel with a robust statistical Spoken Language Understanding module. This is followed by a dialogue to obtain further task-relevant details from the user before passing on the call. The dialogue capability also allows them to obtain clarifications of the initial classifier guess. Based on an evaluation, they show that conducting a dialogue significantly improves upon call routing based on call classification alone. They present both subjective and objective evaluation results of the system according to standard metrics on real users.

The paper of Camelin and et al.; (2010) describes a system for automatic opinion analysis from spoken messages collected in the context of a user satisfaction survey. A process is used for detecting segments expressing opinions in a speech signal. Methods are proposed for accepting or rejecting segments from messages that are not reliably analyzed due to the limitations of automatic speech recognition processes, for assigning opinion hypotheses to segments and for evaluating hypothesis opinion proportions. Specific language models are introduced for representing opinion concepts. These models are used for hypothesizing opinion carrying segments in a spoken message, [4]. The different processes are trained and evaluated on a telephone corpus collected in a deployed customer care service. The proportions estimated with such a low divergence are accurate enough for monitoring user satisfaction over time.

Automatic segmentation and classification of dialog acts is important for spoken language understanding (SLU: Laskowski, Kornel; Shriberg, Elizabeth; 2010). Such paper proposes a framework for employing both speech/non-speech-based (“contextual”) features and prosodic features, and applies it to segment and classify in multiparty meetings. They find that: (1) contextual features are better for recognizing, while prosodic features are better for finding

base mechanisms and backchannels; (2) the two knowledge sources are complementary for most of the studied types; and (3) the performance of the resulting system approaches that achieved using oracle lexical information for several types, [5].

Natural human-robot interaction (HRI) requires different and more robust models of language understanding (NLU) than non-embodied NLU systems as it is described in Cantrell and et. al., 2010. In particular, architectures are required that (1) process language incrementally; (2) use pragmatic contexts throughout the understanding process to infer missing information; and (3) handle the underspecified, fragmentary, or otherwise ungrammatical utterances that are common in natural speech. In this paper [6], they describe attempts at developing an integrated natural language understanding architecture for HRI, and demonstrate its novel capabilities using challenging data collected in human-human interaction experiments.

Using unstructured queries to search a structured database in voice search applications addressed by Young-In and et. al., (2009). By incorporating structural information in music metadata, the end-to-end search error has been reduced by 15% on text queries and up to 11% on spoken queries. Based on that, an HMM sequential rescoring model has reduced the error rate by 28% [7] on text queries and up to 23% on spoken queries compared to the baseline system. A phonetic similarity model has been introduced to compensate speech recognition errors, which has improved the end-to-end search accuracy consistently across different levels of speech recognition accuracy.

In the paper of Heracleous and et. al.; (2009), Hidden Markov Models (HMM)-based vowel and consonant automatic recognition in cued speech for French are presented. Cued speech is a visual communication mode which uses hand-shapes in different positions and in combination with lip-patterns of speech, makes all the sounds of spoken language clearly understandable to deaf and hearing-impaired people. The aim of cued speech is to overcome the problems of lip reading and thus enable deaf children and adults to fully understand a spoken language. This study investigates and also reports automatic consonant recognition experiments in cued speech for French. In addition, isolated word recognition experiments both in normal-hearing and deaf subject are presented, showing a promising word accuracy of 92% on average, [8].

Using speech transformation to increase speech intelligibility for the hearing- and speaking-impaired is presented using two speech transformation approaches designed to increase the intelligibility of speech, [9]. The first approach is used in the context of increasing the intelligibility of conversationally spoken speech for hearing-impaired listeners. The second approach aims to increase the intelligibility of speaking-impaired individuals by the general population. Results of listening tests indicated that an intelligibility increase was not achieved; listeners preferred the transformed speech of the proposed system over that of an alternative system.

Mobile augmented reality (AR) translation system, using a smart phone's camera and touch-screen, that requires the user to simply tap on the word of interest once in order to produce a translation, presented as an AR overlay (Victor Fragoso and et. al., 2011), [10]. The translation replaces the original text in the live camera stream, matching background and foreground colors estimated from the source images. For this purpose, they developed an efficient algorithm for accurately detecting the location and orientation of the text in a live camera stream that is robust to perspective distortion, and they combine it with OCR and a text-to-text translation engine.

Their experimental results, using the ICDAR 2003 dataset and their own set of video sequences, quantify the accuracy of detection and analyze the sources of failure among the system's components. With the OCR and translation running in a background thread, the system runs at 26 fps on a current generation smart-phone (Nokia N900) and offers a particularly easy-to-use and simple method for translation, especially in situations in which typing or correct pronunciation (for systems with speech input) is cumbersome or impossible [10].

The paper of [11] presents a simple and efficient feature modeling approach for tracking the pitch of two active speakers. The paper indicates that, they employ the mixture maximization model (MIXMAX) in addition to linear interaction model. A factorial hidden Markov model is applied for tracking pitch over time. This statistical model can be used for applications beyond speech, whenever the interaction between individual sources can be represented as MIXMAX or linear model. They demonstrate experimental results using Mocha-TIMIT as well as data from the speech separation challenge. The paper showed the excellent performance of the proposed method in comparison to a well-known multi-pitch tracking algorithm. Using speaker-dependent models, the proposed method improves the accuracy of correct speaker assignment, which is important for single-channel speech separation. Moreover, they demonstrate the beneficial effect of correct speaker assignment on speech separation performance.

Particle filtering has been shown to be an effective approach to solving the problem of acoustic source localization in reverberant environments, [12]. In such environment multiple-hypothesis model associated with these arrivals can be used to alleviate the unreliability often attributed to the acoustic source localization problem. Recently, the extended Kalman particle filter (EPF) scheme for the localization problem. Due to this, the extension of the multiple-hypothesis model for this scheme is not trivial. In this paper [12], the EPF scheme is adapted to the multiple-hypothesis model to track a single acoustic source in reverberant environments. Such work is supported by an extensive experimental study using both simulated data and data recorded in their acoustic lab. Various algorithms and array constellations were evaluated. The results demonstrate the superiority of the proposed algorithm in both tracking and switching scenarios. It is further shown that splitting the array into several sub-arrays improves the robustness of the estimated source location.

The paper of [13] presents a method of both separating audio mixtures into sound sources and identifying the musical instruments of the sources. A statistical tone model of the power spectrogram, called an integrated model, is defined and source separation and instrument identification are carried out on the basis of Bayesian inference. Since, the parameter distributions of the integrated model depend on each instrument, the instrument name is identified by selecting the one that has the maximum relative instrument weight. Experimental results showed correct instrument identification enables precise source separation even when many overtones overlap.

In the paper of [14], they present a method for automatically generating acoustic sub-word units that can substitute conventional phone models in a query-by-example spoken term detection system. They generate the sub-word units with a modified version of their speaker diarization system. Given a speech recording, the original diarization system generates a set of speaker models in an unsupervised manner without the need for training or development data. Modifying the diarization system to process the speech of a single speaker and decreasing the minimum segment duration constraint allows detecting speaker-dependent sub-word units. For the task of query-by-example spoken term detection, they show that the proposed system

performs well on both broadcast and non-broadcast recordings, distinct conventional phone-based system trained solely on broadcast data. A mean average precision of 0.28 and 0.38 was obtained for experiments on broadcast news.

Another paper [15] describes a modular, unit selection based TTS framework, which can be used as a research bed for developing TTS in any new language, as well as studying the effect of changing any parameter during synthesis. Using this structure, TTS has been developed for Tamil. Synthesis database consists of 1027 phonetically rich prerecorded sentences. This framework has already been tested for Kannada. Their TTS synthesizes intelligible and acceptably natural speech, as supported by high mean opinion scores. The framework is optimized to suit embedded applications like mobiles and PDAs. They compressed the synthesis speech database with standard speech compression algorithms used in commercial GSM phones. Even with a highly compressed database, the synthesized output is perceptually close to that with uncompressed database. Through experiments, they explored the ambiguities in human perception when listening to Tamil phones and syllables uttered in isolation, thus proposing to exploit the misperception to substitute for missing phone contexts in the database. Listening experiments have been conducted on sentences synthesized by replacing phones with their confused ones.

The paper of [16] addresses non-native accent issues in large vocabulary continuous speech recognition. It proposes to analyze the transformation rules of non-native Mandarin speech spoken by native speakers of *Naxi* and *Dai* in *Yunnan* at the level of initials and finals. Firstly, baseline HMM models are trained using the standard Mandarin corpus to test their performance on non-native speech recognition. Secondly, the non-native speech data is transcribed based on the baseline HMM models. In more detail, they analyze the error recognition rates of all initials and all finals, and their typical substitute error. The results obtained from their experiments might be useful for adapting a native speaker ASR system to model nonnative accented data.

The paper with the title of “Vowel Effects towards Dental Arabic Consonants based on Spectrogram” [17] discussed the effect of vowel (fatha, kasra and damma) in Arabic consonants. These vowels are added to the basic consonants with three simple diacritics using the utterances of every dental consonant concerned by Malaysian children. The dental consonants refer to the consonants utter using dental medium. It is called the place of articulation at labiodentals, dental and interdental. The formant frequencies produced for each place of articulation are based on their spectrogram. The visualization of the spectrogram makes the formants easily being identified by normal human being’s vision. The formants (F1, F2 and F3) are averaged and the results show the different increment and decrement compare to utterances of single phoneme. F1 is decreased for all observed consonants. F2 and F3 changes according to its manner of articulation, where in this study the tabulation are plosive and fricative. The paper noted that, most of the speakers having decreasing frequency of the second and third formant for all consonants pronunciation with vowel effect.

Kensaku Fujii and et. al. [18] proposed a step size control method capable to cancel acoustic echo even when double talk continues from echo path change. This method controls the step size by substituting the difference between the coefficients of a main adaptive filter (Main-ADF) and a sub-adaptive filter (Sub-ADF) for the estimation error in the former. The speed size control can be improved by utilizing the difference for the step size control. The paper shows that in single talk within the proposed method can provide almost the estimation speed as the method

whose step size is fixed at the optimum one and verify that even in double talk the estimation error quickly decreases.

However *Juraj Kac* and *Gregor Rozinaj* discussed the impact of substituting some of the basic speech features with the voiced/ unvoiced information and possibly with the estimated pitch value, [19]. As a good measure of the signal's voicing the average magnitude difference function was assumed, especially the ratio of its average value to its local minima found within the accepted ranges of the pitch. Furthermore, the pitch itself was used as an auxiliary feature to the base speech features. Experiments were performed on the professional database for mobile applications working in harsh conditions, using various HMM models of context dependent and independent phonemes. All models were trained following the training scheme. In all cases the voicing feature brought improved results by more than 9% compared to the base systems. However the role of the pitch itself in the case of speaker independent ASR system evaluated over different tasks was not always so beneficial.

2 PRONUNCIATIONS

This section illustrates how pronunciation can vary, and how can realize a phoneme as different allophones in phonetic environment. This section has also shown how to write transducer rules to model these changes for speech, also accent defects, pronunciation errors and common pronunciation errors will be presented.

Lexical variation and allophonic variation are two classes of pronunciation variation. The lexical variation is used to represent the word in the lexicon, while the allophonic variation is a difference in how individual segments change their value in different segments, [20]. Most of the variation in pronunciation is allophonic, according to the influence of surrounding sounds and syllable structure.

The lexical variation is related to sociolinguistic variation. It is due to extra linguistic factors such as dialect variation. Other socio linguistic differences are due to register or style rather than dialect. One of the most well-studied example of style-variation is the suffix -ing (as something), which can be pronounced somethin (without g) [20, 25].

The proposed rules are dependent on a complicated set of factors that must be interpreted probabilistically. Most allophonic rules relating English can be grouped into number of types: assimilation, dissimilation, deletion, flapping, vowel reduction, and epenthesis (insertion of an extra sound into a word), [20].

Assimilation is the change in segment sound to make it more like a neighboring segment, e.g.; dentalization and palatalization. As an example of palatalization rule as follows:

$$\left\{ \begin{array}{c} [s] \\ [z] \\ [t] \\ [d] \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} [ʃ] \\ [ʒ] \\ [tʃ] \\ [dʒ] \end{array} \right\} / - \{y\}$$

Deletion is the removal of a sound from a word. The following rule includes /t/ and /d/ is deleted before consonants:

$$\left\{ \begin{array}{c} t \\ d \end{array} \right\} \Rightarrow \emptyset / V - C$$

The *flapping* is happening when the speaker is speaking more quickly, and is more likely to happen at the end of a word when it forms a collection. The most important phonological process is vowel reduction or reduced vowel.

2.1 ACCENT DEFECTS AND PRONUNCIATION ERROR CATEGORIZATION

Some errors that may be noticeable will not cause any difficulties in understanding from a native listener's point of view, whereas other types of errors will cause serious problems for the intelligibility of an utterance, especially for the non native speakers. Regardless of what the final aim is, any learner will benefit from realizing the impact of various errors.

According to previous studies and works; this paper investigates pronunciation difficulties in second language learners from Egypt and Saudi Arabia citizens, with English as target language. The main motivation for this work was to create guidelines for teachers of English as a foreign language. In order to get recordings that were representative for each language group, and that covered all aspects of pronunciation without making the material.

A. General Phonology

The Arabic and English phonological systems are different, in the range of sounds used in emphasizes of the position on vowels and also in consonants in expressing meaning.

English has 22 vowels and diphthongs to 24 consonants. Arabic has only eight vowels and diphthongs to 32 consonants [25]. The Arabic vowels include three short, three long and two diphthongs. Therefore, Arabic speakers tend to gloss over and confuse English short vowel sounds, while unduly emphasizing consonants, avoiding elisions and shortened forms [16, 17, 25]. Within each country, a wide variety of colloquial dialects have developed, differing one from another in pronunciation, common lexical items, and in structure, [25].

The informants made recordings that were both read speech, and free speech guided by pictures and sequences of pictures. Both sentences and isolated words were recorded in each of the two categories. The read texts were designed to cover various aspects of the pronunciation accented errors and highlight pronunciation difficulties.

A comprehensive table listing the difficulties for each of the category, and in addition attention was given to re-occurring difficulties, as well as a categorization of errors based on the seriousness from an intelligibility point of view. Based on this analysis supervisor of acoustic (Adel Al-Sheikh) sorted errors on an intelligibility scale as a guideline for what aspects of pronunciation should be prioritized in pronunciation teaching. The most serious errors in ascending order according to Al-Sheikh are shown in Table 1.

Table 1

The most serious errors for learners' of English (Egyptian and Saudi Arabian Peoples)

No.	Error type or Category	Example
1	Labial-dental Fricatives:	Substituting: /f/ and /v/. - /v/ for /f/.
2	Rolling the	Library, Ruler, Lorry, Liberian, and Reroofing
3	Replacing	Replace /θ/ with /s/, as in sin for thin. Replacing /tʃ/ sound with /ʃ/ as in sheep for cheap
4	Dental Fricatives	Dental Fricatives: /θ, ð /
5	Pronouncing the grapheme	Pronouncing /g/ only as /ʒd/ or as in gentle/ʒd /. Bilabial Plosives: /p, b/. Alveolar Fricatives: /s, z/.

The initial work on creating pronunciation error detectors for the proposed framework is inspired by paper work of as will be described in the table (2).

Table 2

The most serious errors for learners' of English

No.	Error Type or Category	Error Description
1	Lexical stress	Insufficient stress marking, or stress on the wrong syllable
2	Syllable structure	Incorrect number of syllables in a word.
3	Consonant clusters	Vowel insertion (epenthesis) in, or before a consonant cluster, or consonant deletion in a consonant cluster before a stressed vowel.
4	Rhythm	The relationship between stressed and unstressed syllables in a sentence is wrong

To complete speech correct tasks, there are many sources of “defects, pronounced errors and acoustic variation” – in Saudi and Egyptian accents. For spelling error detection, what we mean by defects in pronounced text, which mask the correct spelling of the text. Therefore, the following subsection breaks the field down into four increasingly boarder problems:

1. Substituting /v/ for /f/, such as saying the word: vat/fat, very/ferry, belief/ believe, vast/fast and van/fan.
2. Rolling the /r/ as an examples: Library, Ruler, Lorry, Liberian, and Reroofing.
3. Replacing /θ/ with /s/, as in sin for thin: thong/ song, thank /sank , theme/seem, / thin sin and thought / sought.
4. Dental Fricatives: /θ, ð /: Replacing /ð/ with /z/ or /d/, as in dat or zat for that, and / ð / with /θ/. Therefore Ss may replace the / ð / sound as in “brother”, “they” and “these”, with the /θ/ sound. As examples: another, blithering, bother, brother, and father.

Other researches discuss kinds of spelling error patterns that occur in typed text and speech-recognition [20, 22]. Single-error misspellings induced by one the following errors: insertion, deletion, substitution, and transposition. While typing errors are usually characterized as substitutions, insertions, deletions, or transpositions, OCR errors are usually grouped into five classes: substitutions, multi-substitutions, space deletions or insertions, and failures, [20].

B. Vowels

Some of English phonemes have equivalent or near equivalent in Arabic and therefore be perceived and articulated without great difficulties. Some English phonemes may cause problems, the following are most confusion, [25]:

- a. /e/ and /ɪ / are often confused; for example bit for bet.
- b. The two phonemes /ɒ/ and /ɔ:/ are often confused ; e.g.; cot for caught.
- c. Diphthong /əʊ / and /eɪ/ are usually pronounced rather short, and may are confused with /e/ and /ɒ/; as an example red for raid.

C. Consonants

Many of English phonemes have equivalent or near equivalent in Arabic and therefore can be articulated without difficulties. Although some confusion may still arise, few of phonemes may cause problems, the following comments illustrate examples of such problems:

- a. The Arabic letter /g/ is pronounced as /g / in Egyptian accent, and /dʒ / in Saudi accent, and sometimes even as /j/ according to local dialects.
- b. The two letters /v/ and /f/ are often confused, especially in Saudi accent; e.g.; It is a fery nice fillage.
- c. The two allophones /p/ and /b/ tend to be used rather randomly: I baid ten bense for a bicture.
- d. /θ/ and /ð/ occur and dialect pronounce them as /t/ and /d/ respectively- especially in Egyptian accent- I tink dat dey

- e. The rolling of /r/ is voiced flap, Arabic speakers overpronounce the post-vocalic r; as in car park.
- f. Sometimes /g/ and /k/ are often confused, especially whose dialects do not include the phoneme /g/, as in goat/coat and bag/bak.

D. Consonant Clusters

The number of consonant clusters occurring in English is greater than in Arabic. Initial two segment clusters rarely occurring in Arabic [25]: pr, pl, gr, gl, thr, thw, and sp.

Initial three-segment clusters do not occur in Arabic: spr, skr, str, and spl. According to these clusters, there is tendency among Arabic speakers to insert short vowels to assist pronunciation:

ispring or sipring for spring.

perice or pirice for price.

And also, for the range of final clusters [25]:

monthiz for months

Neckist for next.

E. Rhythm, Stress and Intonation

Arabic speakers have problems grasping the unpredictable nature of English word stress, because the Arabic is stress-timed language, and word stress is predictable and regular.

Rhythm is similar in Arabic and English, and sometimes causes few problems. Primary stress, occur in Arabic and unstressed syllables are pronounced more clearly in English.

Whereas intonation patterns are similar in Arabic and English - using rising tune (questions, suggestions, etc.).

2.2 COMMON PRONUNCIATION ERRORS

Figure 1 illustrates almost pronunciation errors that are needed to be addressed in successful training and assessment models, [23]. As shown in the figure, it can be classified into phonemic and prosodic error types.

- (1) The phonemic errors can be divided into substituted, deleted or inserted. Also, there are errors on a little scale “where the correct phoneme is more or less being spoken”, [23].
- (2) The prosodic errors can be categorized into stress, rhythm and intonation.

Therefore, these two types of errors make pronunciation a multi-dimensional problem. Consequently, large number of metrics is used to measure these dimensions [23, 24].

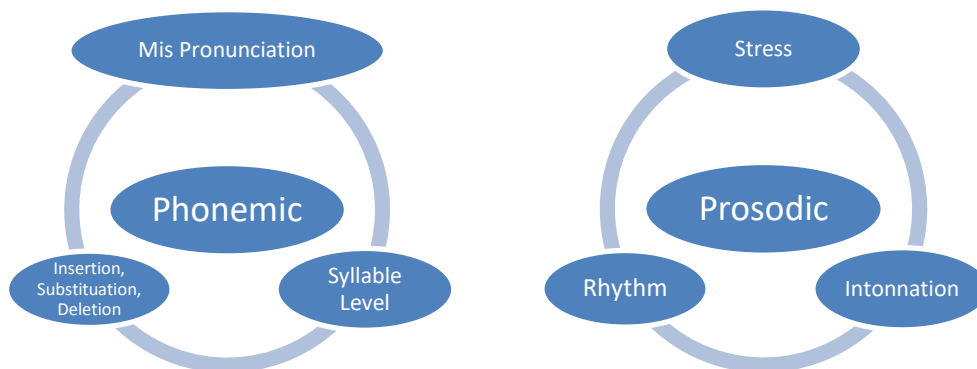


Figure (1): Classification of Pronunciation Errors [23]

During the development of Speak Correct we found a significant body of literature describing the typical patterns of error made by Korean Learner Segmental Errors (KLEs), [24]. A pilot corpus is collected and phonetically annotated consisting of prompted English speech data from an assortment of different types of content. The corpus includes short paragraphs of text,

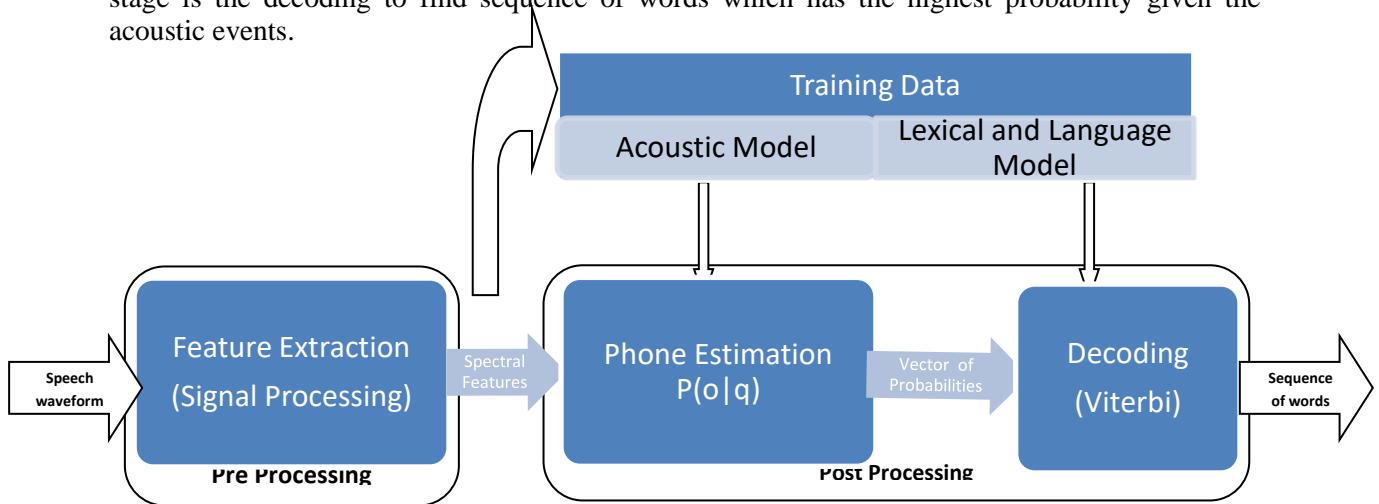
sentence prompts, and words with particularly difficult consonant clusters (e.g., refrigerator). In total, the pilot corpus collected 25,000 total speech samples from 111 learners who reside in Korea. The corpus provides a direct comparison of realized phone sequences compared to expected canonical sequences from native speakers. A summary of Korean to English speakers also makes no distinction between fricative /f/ and /v/, substitute /p/ and /b/ instead. Other common reported errors include the substitution of aspirated /t/ for /θ/ and un-aspirated /t/ for /ð/, [23]. Table (3) illustrates the most frequent segmentation observed errors, [24].

Table 3
Frequent Korean Learner Segmental Errors [24]

<i>Error Pattern Description</i>	<i>Error %</i>
Unstressed to Stressed Vowel (e.g., /i/ → /ɪ/)	14%
/ɪ/ Issue (deletion, substitution, etc.)	11%
Dental to Alveolar (e.g., /θ/ → /t/)	9%
Consonant Deletion (word initial /j/ → /j/)	8%
Vowel Insertion (e.g., large → largɪ)	7%
Diphthong to Monophthong (e.g., /eɪ/ → /ɛ/)	5%
Consonant Cluster Simplification	5%
Vowel to Central Vowel (e.g., /u/ → /ɪ/)	5%
Vowel Deletion (e.g., /ə/ → / /)	4%
/æ/ to /ɛ/	4%

3. SPEAK CORRECT PROCESSING ENGINE

Any speech recognition system consists of two main modules: The first module is called pre-processing or feature extraction and the second module is post-processing divided into acoustic, lexical and language modeling [44]. Figure (2) shows an outline of speech recognition system components. Such system is broken down into three stages. The first stage is used for signal processing or feature extraction, the acoustic waveform is sliced up into frames which are transformed into spectral features. The second stage, phone estimation that includes statistical techniques (e.g., neural networks or Gaussian models) to recognize individual speech sounds like *f* or *s*. The output of this stage is a vector of probabilities over phones for each frame. The last stage is the decoding to find sequence of words which has the highest probability given the acoustic events.



Figure(2): Architecture for Simplified Speech Recognizer

To summarize the process of extraction features, starting from sound waves and ending with feature vector. First, an input sound wave is *digitized* (it is analog-to-digital conversion), and it has two steps: *Sampling* and *quantization*.

The *sampling* rate is the number of samples taken per second (8000 Hz and 16000 Hz are two common sampling rates). At least two samples in each cycles are needed to measure a wave accurately: one for positive part of the wave and one for the negative part, and more than two samples per cycle increases the amplitude accuracy.

At each sampling rate (8000 Hz or 16000 Hz), there are amplitude values for each second of speech. The process of representing such values as integers is called *quantization*. After the *digitization* of the waveform, it is converted to set the spectral features. It is possible to use any popular feature set (Linear Predictive Coding (LPC) or Perceptual Linear Predictive (PLP)) directly to observe symbols of an HMM [1, 6, 7, 21]. Further processing is often done to the features; like cepstral, which are computed from the LPC coefficients by taking the Fourier transform of the spectrum.

The *phones estimation* is an efficient way to compute the likelihood of an observation sequence given weighted automata. HMM allows us to sum multiple paths that each account for the same observation sequence.

The *decoding* stage is the problem of finding determining the correct “*underlying*” sequence of symbols. Therefore, the *Viterbi* algorithm is an efficient way of solving the decoding problem by considering all possible strings and using addition rules (like Bays rule [20]) to compute their probabilities of generating the observer sequence.

3.1. Speak Correct Background Architecture

Many of researchers have introduced many of core algorithm used in speech recognition. Therefore, the notions of phone and syllable are introduced [8, 11, 19]. In addition, N-gram language model and the Hidden Markov Model (HMM) discussed in more details [8,11].

HMM were first described by Leonard E. Baum and others in 1960s. The HMM is stochastic methods to model temporal pattern recognition and sequence data. One of the first application fields of HMMs was speech recognition in the med of 1970s. Therefore, tutorial on HMMs was published by Lawrence R. Rabiner, so, analysis of biological sequences (DNA) began to be applied at the second half of 1980s. The HMMs can be illustrated using finite state machines, at each transition there is an observation from specific state, for each state there is output symbol emission. Figure (3) summarizes the overall definition of the HMM, [21].

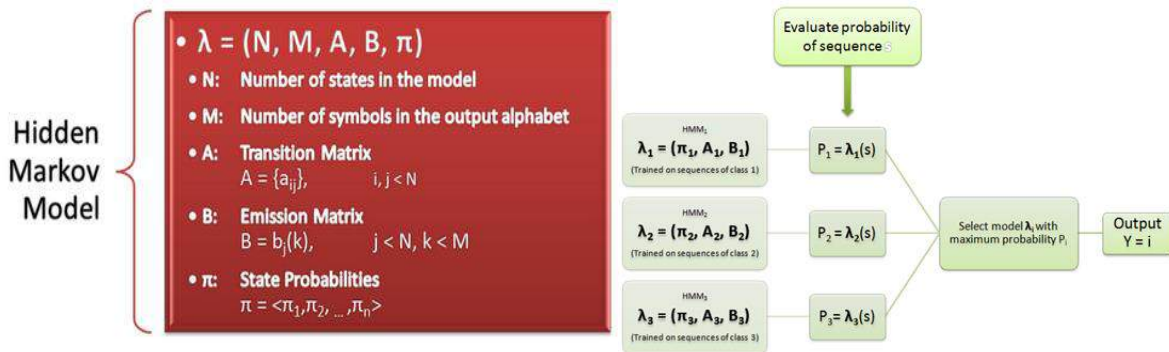


Figure (3-a): HMM Overall Definition [21] Figure (3-b): HMM Overall Computation [21]

In other words, to choose the word which is most probable given the observation, the single word such that $P(\text{word} | \text{observation})$ is highest. If w is the estimated correct word and if the O is the observed sequence (individual observation), then the equation for picking the best word given is:

$$W = \operatorname{argmax} P(o|w) P(w)$$

Where: $P(o|w)$ represents likelihood, $P(w)$ represents prior, w is vocabulary, w is correct word, o is observation. How to compute these two probabilities will be discussed in the next section. Once the likelihood- computation has been solved, and decoding problems for a simplified input consisting of strings of phones, feature extraction will quickly be involved.

3.2 Acoustic Probabilities Counting

As mentioned before, the speech input can be passed through signal processing transformations and converted into series of vectors of features, each vector representing one time-slice of the speech input signal. One of the popular ways to compute probabilities on feature vectors is to first cluster such feature vectors into discrete counted symbols. Therefore, the probability of a given cluster can be calculated (number of times it occurs in some training set). This methodology is called **vector quantization**; and it is developed into computing observation probabilities or probability density function (pdf). There are two common approaches; **Gaussian pdfs** that maps the observation vector O_t to a probability. The second alternative is the use of neural networks or multi-layer perceptions, that can be trained to assign a probability to speech real-valued feature vector. The neural network is a set of small computation units connected by weighted links. The network is given a vector values and computes a vector of output values.

A standard model based on probabilistic neural network is proposed in [21], it is suitable for testing and pattern classification. The structure of such probabilistic neural network used in this report is shown in figure (4). The number of input speech variable is M , the number of identification patterns are needed is N , the training samples for each patterns are represented by S_1, S_2, \dots, S_N . There are four layers: input layer, model layer, summation layer and output layer, the weights between summation layer and output layer is computed by:

$$W(M) = S_i / \sum_{i=1}^j S_i$$

3.3 SPEAK CORRECT PRINCIPLES MODULES

Many of researches have introduced many core algorithm used in speech recognition. Therefore, the notions of phone and syllable are introduced. This in added to N-gram language model and the Hidden Markov Model (HMM). Our goal is to build a model, so that we can figure out how it modified this “true” word and hence recover it. For the complete speak correct tasks, there are many sources of “defects”: Substituting /v/ for /f/, Rolling the /r/, Replacing /θ/ with /s/, Dental Fricatives: /θ, ð /: Replacing /ð/ with /z/ or /d and acoustic variation due to the channel (Microphone, networks, etc).

Consequently, the operation speed and the use effect should be affected. Therefore, the proposed technology has good ability of eliminating the data correlation, so, a speaker recognition pattern based on the combination of PRS, HMM, ASR, intonation analyzer and pronunciation generator is proposed in this report. The basic recognition processes are as the following.

(A) Main Module

Step 1: Gathering and collecting the speech inputting samples.

Step 2: Dividing such samples into two parts, one part is for training samples and the second part is for testing.

(B) Training Module

Step 3: Do the following:

- 3.1 Speaker Adaption.
- 3.2 Confidence measuring.
- 3.3 Tuning the native Arabic speaker accent.

- a. Tuning Saudi accent
- b. Tuning Egyptian accent.

3.4 Intonation training and teaching the prosodic effects.

Step 4: Using the feature vector of training samples to train the *SpeakCorrect* model.

(C) Testing Module

Step 5: Do the following steps:

Step 5.1: Establishing PSR with the associated ATN neural network and HTK mechanism.

Step 5.2: Inputting the feature vectors of test samples into PSR network which has been trained.

Step 5.3: Judging the corresponding speech signal category and the speaker identity according to the output values.

The coming sections include detailed description in more details.

3.4 FINITE STATE/WEIGHTED FINITE STATE AND WEIGHTED ATN/LATTICE

Computational linguistics and automata theory were used to predict letter sequences, describe natural language, employ context-free grammars (CFG), introduce the theory of the tree transducers, and parse automatic natural language text [28-35]. In the 1970s, speech recognition researchers captured NLP grammar with weighted finite state acceptors (FSAs), by employing transition weights that could be trained on machine-readable using dictionary, corpus, and corpora [36,37,39-43]. In the remainder of this section, we discuss how natural language applications use tree automata.

In the 1990s, combination of finite state and large training corpora became the dominant paradigm in speech and text processing; software toolkits for weighted finite state acceptors and transducers (WFSTs and WFSTs) were developed [28]. The 21s century has seen generic tree automata toolkits [36] that have been developed to support investigations.

The single WFST or augmented transition network (ATN) that represents P(S|E) is still complex, model transformation can be made into chain of transducers in the following:

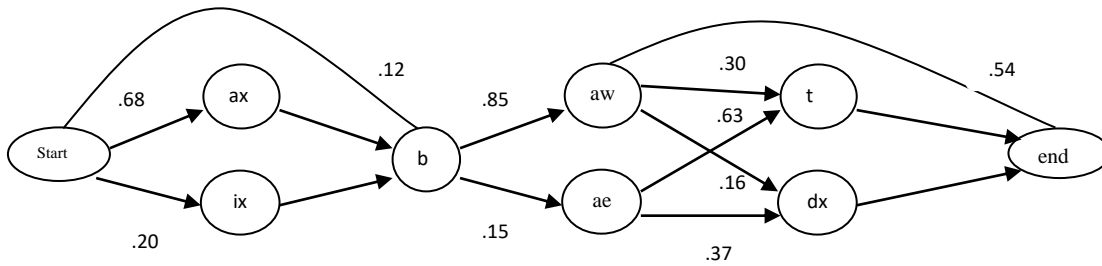
$$\text{WFSA}_a (\text{English}_{\text{text}}) \leftrightarrow \text{WFSA}_b (\text{English}_{\text{sound}})$$

Therefore, simple model can be used to calculate 1-gram, 2-gram, and n-gram language model of characters [28]. If an corpus includes 1,000,000 characters, the letter e occurs 127,000 times, the probability P(e) estimated as 0.127.

In case of 2-gram model, it can be calculated by remember the previous letter context- its WFSA state. As an example the transition between state r and state e outputs the letter e can be calculated by the probability P(e|r). The n-gram model generates more word-like items than (n-1)-gram model does. The weighted ATN is simple automaton in which each arc is associated with a transition, this transition can be represented by probability value, indicating how likely that path is to be taken. The probability on all the arcs leaving a node must sum to 1. Figure 4 shows weighted ATN for the English word “about” which is trained on actual pronunciation example.

This model is an instance of a Hidden Markov Model (HMM). Such figure illustrates graphically the behavior transition in the weighted ATN. The rule of the transition according to the following:

- Starts in some initial state (start: s_1) with probability $p(s_1)$,
- On each move goes from state s_i to state s_j according to transition probability $p(s_i, s_j)$.
- At each state s_i , it emits a symbol w_k according to the emit probability $p'(s_i, w_k)$.



Legends: $P(w | ax) = .68$ $P(w | ix) = .20$

Figure (4): A Pronunciation Network (Weighted ATN) for word “*about*”

The use of *SpeakCorrect* is often called hybrid approach, since it uses elements of the HMM or weighted state-graph representation of the pronunciation of a word, also it uses observation-probability computation using multilayer perception. The input to this multilayer perception is a representation of the signal at a time t , vector of spectral features for time t , and eight additional vectors for times $t+10$ ms, $t+20$ ms, $t+30$ ms, $t+40$ ms, $t-10$ ms, and so on. The network has one output unit for each phone; by summing the values of all the output units to 1, the *SpeakCorrect* can be used to compute probability of a state j given an observation vector O_t , or $P(j|O_t)$, or $P(O_t | q_j)$.

Therefore, receiving the sequence of spoken words that generated a given acoustic speech signal, a standard model- like described in figure 5- is used. The model generates $P(E|S)$ for a received speech signal S , and such model is described as the following:

1. For each word/phonetic in S , a variety of individual units of speech (sequence of phonemes), may be observed with varying probabilities, and therefore, can be interpreted as the word.
2. For each word, a word-to phone is constructed.
3. Each phone can be expressed as a variety of audio signal.

Once defined, the chain of audio signal and the final language model are weighted with the method of likelihood, and observing probabilities from the training data.

3.5 TRAINING THE *SPEAKCORRECT*

A brief sketch of the embedded training procedure is used in most of ASR systems. Some of the details of the algorithm have been introduced in [8, 11, 16, 21, 22]. Four probabilistic models are needed to train *SpeakCorrect* system:

- Language model probabilities: $P(w_i | w_{i-1} w_{i-2})$
- Likelihood observation : $b_j(O_t)$
- Transition probabilities: a_{ij}
- Pronunciation Lexicon: Weighted ATN of HMM state graph structure.

In order to train the previous probabilities component the *SpeakCorrect* has the following corpuses:

- Training corpus of speech wave files: which are collecting from news web site of the internet, individual peoples ... etc. This speech wave files are collected together with word- transaction.
- Large corpus of text: including the word-transaction from speech corpus together with many other similar texts.
- Smaller training corpus of speech: which is phonetically labeled, i.e. frames are hand-annotated with phonemes.

The HMM lexicon structure is built, by taking an off-the shelf pronunciation dictionary.

Therefore, the training is beginning by run the model on the observation and seeing which transitions and observations were used. Any state can generate one observation symbol; the observation probabilities are all 1.0. The probability p_{ij} of a particular transition between states i and j can be computed by counting the number of transition was taken; $c(i \rightarrow j)$. Normalize such value by using the following:

$$a_{ij} = c(i \rightarrow j) / \sum_{q \in Q} c(i \rightarrow q)$$

For the weighted ATN and HMM, two methods are used, the **first** idea is to *iteratively* estimate the counts, observation probabilities, and the use such estimated probabilities to derive better and better probabilities. The **second** idea is get estimated probabilities by computing forward probability among all different paths. Define the forward probability in state i after seeing the first t observation, given the automaton A .

$$a_t(i) = P(o_1, o_2, o_3, \dots, o_t, q_t = i | A)$$

Formally, define the following iteration:

1. Initialization:

$$a_n(1) = a_{1j} * b_j(o_1) \quad \dots\dots\dots 1 < j < N$$

2. Iteration:

$$a_j(t) = [\sum_{i=2}^{N-1} a_i(t-1) * a_{ij}] b_j(o_t) \quad \dots\dots\dots 1 < j < N, 1 < t < T$$

3. Termination:

$$p(o|A) = a_N(T) = \sum_{i=2}^{N-1} a_i(T) * a_{iN}$$

The forward algorithm can be run to compute the candidate words was most probable given the observation sequence [ax b], the product $P(o | w) P(w)$ is computed for each candidate word. So, the likelihood of observation sequence o given the word w times the prior probability of the word is computed for each word, and choose the word with the highest value.

The forward algorithm is an edit distance algorithm, it uses a table to store intermediate values as it builds up the probability of the observation sequence. The data is represented in the table by rows oriented; the rows are labelled by state-graph which has many ways of getting from one state to another. The table is filled as a matrix by computing the value of each cell from the three cells around it. Furthermore, the forward algorithm computes the sum of probabilities of all possible paths that could generate the observation sequence.

Each cell of the forward algorithm matrix, $forward[t, j]$ represents the probability of being in state j after seeing the first t observations, given the automaton A . Formally, each cell expresses the following probability:

$$forward[t, j] = P(o_1, o_2 \dots o_t, q_t = j | A) P(w)$$

The following pseudo code describes the forward algorithm applied to any word.

forwardAlgorithm (observation, state-graph)

begin

ns = numOfStates(state-graph);

no= length(observation);

/ create probability matrix */*

forward [ns+2 , no + 2];

```

forward [0,0] = 1.0;
foreach time step t from 0 to no do
  foreach states from 0 to ns do
    foreach transition s' from s specified by state-graph
      forward [ s' , t + 1 ] = forward [ s , t ] * a[s , s'] * b [s' , ot];
  return sum of the probabilities in the final column of forward;

```

end.

Where:

- a [s , s'] represents transition probability from current state s to next state s'
- b [s' , o_t] is the observation likelihood of s' given o_t
- b [s' , o_t] is equal 1 if the observation symbol matches the state, and is equal 0 otherwise.

The part of the forward-backward algorithm is the backward probability. This backward algorithm is almost the mirroring of the forward probability [21]. It computes the probability of the observations from t+1 to the end. Suppose that we are in state j at time t to given automaton **A**; then:

$$\beta_i(o_t) = P(o_{t+1}, o_{t+2}, o_{t+3}, \dots, o_T | q_t = j, \mathbf{A})$$

The backward computation is defined as the following:

1. Initialization:

$$\beta_i(t_1) = a_{iN} \quad \dots\dots\dots 1 < i < N$$

2. Iteration:

$$\beta_i(t) = \sum_{i=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta_j(t+1) \quad \dots\dots\dots 1 < j < N, T > t < = 1$$

3. Termination:

$$p(o|A) = a_N(T) = \beta_1(T) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) * \beta_j(1)$$

Therefore, the transition probability a_{ij} and observation probability $b_i(o_t)$ will be computed from an observation sequence.

4. IMPLEMENTATION AND TESTING

The following developing model is based on components to generate pitch contour for pronunciation analysis and pronunciation adaption.

4.1 THE SPEAKCORRECT CORPUS ARCHITECTURE

The SpeakCorrect corpus is based on annotated speech; it will be designed to provide data for the acquisition of acoustic-phonetic knowledge and to support the development and evaluation of automatic speech recognition systems.

A. The SpeakCorrect Structure

Like the Brown Corpus, *SpeakCorrect* includes a balanced selection of dialects, speakers, and materials. It contains three dialect regions, 100 male and female speakers having a range of ages and educational backgrounds each read 150 carefully chosen words. The words were chosen to be phonetically rich and cover all the pronunciation defects of Arabic speakers (Saudi and Egypt regions). Additionally, the design walkouts equilibrium between multiple speakers saying the

same word in order to permit comparison across speakers, and having a large range of words covered by the corpus to get maximal coverage of defects. One hundred of the speakers were read by each region, therefore, 100 hundred recorded utterances are stored in the corpus, each file name has internal structure, as shown in Figure 5.



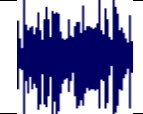

Speaker ID	Gender	Word ₁	Word ₂	...	Word ₁₄₉	Word ₁₅₀
001	Female			...		
002	Male
...
150	Female

Figure (5): Structure of SpeakCorrect Structure

Each item has a phonetic transcription which can be accessed, the corresponding word tokens.

B. The SpeakCorrect Design Features

SpeakCorrect includes features of corpus design. First, such corpus contains two layers of annotation, the phonetic and orthographic levels. At this level there are different labeling schemes. A second property of *SpeakCorrect* is its balance across multiple dimensions of variation, to cover dialect regions and diphones, which facilitate later uses of corpus for purposes, when the corpus was created, such as sociolinguistics.

A fourth feature of *SpeakCorrect* is related to hierarchical structure of the corpus. With 150 words/sentences, and 100 speakers there are 15,000 files. These are organized into a tree structure, shown schematically in Figure 6.

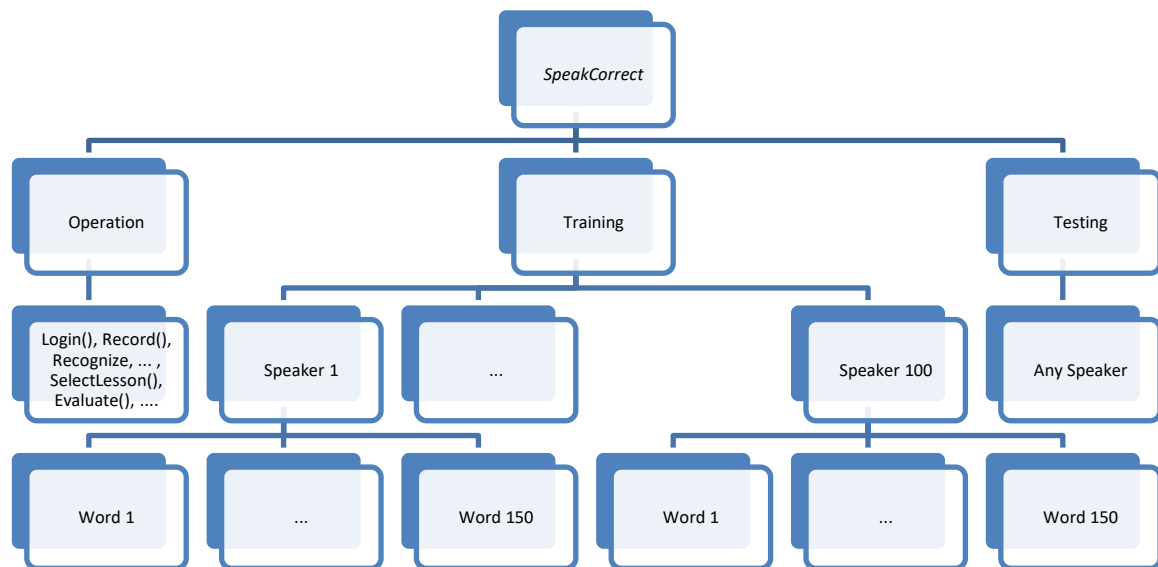


Figure (6): Structure of the Implemented *Speak Correct* Corpus

C. The SpeakCorrect Data Acquisition

The web is one source of rich repository of data for many natural language processing purposes. However, in our case large quantity of data samples are needed to obtain. Consequently, one of

such approach is to obtain a published data from the web. The advantage of using such well-defined web data is that they are documented, stable and reproducible experimentation.

4.2 THE SPEAKCORRECT DIAGRAMS

The Visual Studio is used to draw a *component diagram* to show the structure the *SpeakCorrect* system. Therefore, UML component diagrams are created to represent the **architecture** of the *SpeakCorrect* system.

A. SpeakCorrect Use Case Diagram

The *use case diagram* is used to summarize who uses the *SpeakCorrect* system. Such diagram is used to illustrate (see figure 7):

- The scenarios in which the *SpeakCorrect* system interacts with peoples, organizations, or external systems.
- The goals that it helps those actors achieve.
- The scope of the *SpeakCorrect* system.

Figure (7): The Proposed Use Case Diagram of the *SpeakCorrect* System

B. SpeakCorrect Class Diagram

The *UML class diagram* describes data types and their relationships separately from their implementation. The diagram is used to focus on the logical aspects of the classes, instead of their implementation. There are three standard kinds of classifier available on the toolbox of the UML tools. These are referred to as *types*: classes, interfaces, and an enumeration. The Classes is used to represent data or object types for most purposes. The Interfaces in a context is employed to differentiate between pure interfaces and concrete classes that have internal implementations. This difference is useful when the purpose of the diagram is to describe a software implementation. And, the Enumeration is used to represent a type that has a limited number of literal values. Figure 8 shows the proposed class diagram of the *SpeakCorrect* that includes 4 classes and one interface.

Figure (8): The Proposed Class Diagram of the *SpeakCorrect* System

C. **SpeakCorrect Sequence Diagram**

The *sequence diagram* is drawn to display an interaction. An interaction is a sequence of messages between typical instances of classes, components, subsystems, or actors. There are two kinds of sequence diagrams: UML sequence diagrams that are part of UML modeling projects, and Code-based sequence diagrams that can be generated from .NET program code. Figures (9-a, and 9-b) illustrate sequence diagrams of the *SpeakCorrect* system.

■

Figure (9-a): Sequence Diagram of the *SpeakCorrect* System

Figure (9-b): Login Sequence Diagram of the *SpeakCorrect* System

4.3 THE SPEAKCORRECT USER INTERFACE

The user interface in *SpeakCorrect*, as shown in Figure 10, is divided into three tiers. The top part contains the presentation tier; the middle part of the user interface includes the logical or business tier; which starts with registration where the login takes place login, microphone setting and

adaptation, language and speech lessons and finally evaluation. The third tier is the internal one, which hosts all the properties, databases, files, etc.

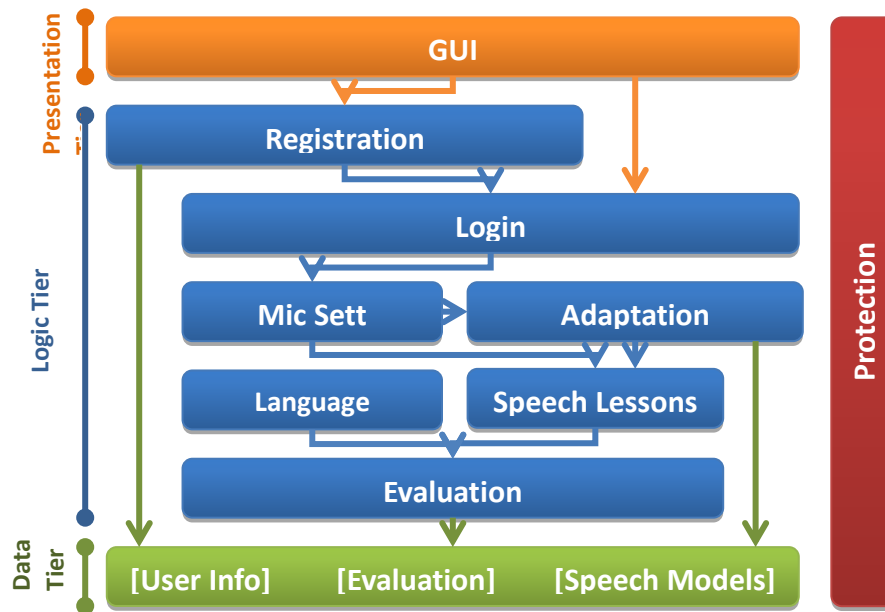


Figure (10): The Different Tiers of the *SpeakCorrect* System

4.4 THE SPEAKCORRECT TESTING

Due to previous difficulties found in Arabian accents for English pronunciation, the following testing model is based on component to evaluate and guide students for pronunciation analysis and pronunciation adaption. Figure 11 illustrates the login student information.



Figure (11): The Login Interface of the *SpeakCorrect* System

Consequently, the user interface is designed in Silverlight technology. Such user interface includes different visual properties to perform basic functions: Moving to previous and next demos playing sample (predefined example), user voice and recording user’s voice. Figure 12 illustrates device setting and microphone adjustment.



Figure (12): The Device Setting and Microphone adjustment of the *SpeakCorrect* System

The pitch contour is the fluctuation in frequency associated in human voices. The development of the proposed project includes “open source .Net Code” which included the pitch contour calculation, in added to HTK code that contains mathematical algorithms. The implementation code contains collaboration module between C# code (.Net Client/Server) and the HTK component code. The second component is trying to compare the input voice against the predefined trained voices and therefore providing a mistake-if any, as shown in figure 13.



Figure (13): The Levels and Associated Lessons Testing of the *SpeakCorrect* System

Consequently, the user interface of the *SpeakCorrect* is designed in Silverlight technology. Such user interface includes tabs to perform basic functions: Moving to previous and next demos playing sample (predefined example), user voice and recording user’s voice, as shown in different previous figures. Consequently, the Model-View-ViewModel (MVVM)

pattern is used to facilitate the interconnection “Click Event” for the tabs. Such visual elements properties are bounded in the underlying ViewModel class.

Below in figure 14 describes pronunciation guide with visually feedback. The feedback contains hints about things the user might try to say if the conversation has stalled. The things-tab holds a picture of all the items the user has managed to acquire. Finally, the evaluation offers lesson summarization to the students at different levels.



Figure (14): The Levels and Associated Lessons Testing of the *SpeakCorrect* System

ACKNOWLEDGEMENT

The teamwork of the *SpeakCorrect* project was funded as part of the Strategic technology project (10-INF-1406-03) held at the King Abdulaziz City for Science and Technology (KACST). Therefore; authors of the paper thankful to KACST through their grand’s number 10-INF-1406-03. Their financially and support during the period this research took place is greatly acknowledged.

REFERENCES

- [1] Macherey, K.; Bender, O.; Ney, H., (2009). Applications of Statistical Machine Translation Approaches to Spoken Language Understanding, Audio, Speech, and Language Processing, IEEE Transactions on Volume: 17 , Issue: 4.
- [2] De Mori, R.; Bechet, F.; Hakkani-Tur, D.; McTear, M.; Riccardi, G.; Tur, G.; (2008). Spoken language understanding, Signal Processing Magazine, IEEE Volume: 25 , Issue: 3.
- [3] Dinarelli, M.; Stepanov, E.A.; Varges, S.; Riccardi, G.; (2010). The LUNA Spoken Dialogue System: Beyond utterance classification, Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference.
- [4] Camelin, N.; Bechet, F.; Damnati, G.; De Mori, R.; (2010). Detection and Interpretation of Opinion Expressions in Spoken Surveys, Audio, Speech, and Language Processing, IEEE Transactions on Volume: 18 , Issue: 2.
- [5] Laskowski, Kornel; Shriberg, Elizabeth; (2010). Comparing the contributions of context and prosody in text-independent dialog act recognition. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference.
- [6] Cantrell, R.; Scheutz, M.; Schermerhorn, P.; Xuan Wu; (2010). Robust spoken instruction understanding for HRI. Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference.
- [7] Young-In Song; Ye-Yi Wang; Yun-Cheng Ju; Seltzer, M.; Tashev, I.; Acero, A.; (2009). Voice search of structured media data, Acoustics, Speech and Signal Processing, 2009. ICASSP, IEEE International Conference.

- [8] Heracleous, P.; Aboutabit, N.; Beautemps, D.; (2009). HMM-based vowel and consonant automatic recognition in Cued Speech for French, Virtual Environments, Human-Computer Interfaces and Measurements Systems. VECIMS '09. IEEE International Conference.
- [9] Kain, A.; van Santen, J.; (2009). Using speech transformation to increase speech intelligibility for the hearing- and speaking-impaired, Acoustics, Speech and Signal Processing. ICASSP 2009. IEEE International Conference.
- [10] Fragoso, V. Gauglitz, S. Zamora, S. Kleban, J. and Turk, M. (2011). TranslatAR: A mobile augmented reality translator. Applications of Computer Vision (WACV), 2011 IEEE Workshop on 5-7 Jan. 2011.
- [11] Michael Wohlmayr, Michael Stark, and Franz Pernkopf, (2011). A Probabilistic Interaction Model for Multi-pitch Tracking With Factorial Hidden Markov Models. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 19, No. 4, May 2011.
- [12] Levy, A.; Gannot, S.; Habets, E.A.P., (2011). Multiple-Hypothesis Extended Particle Filter for Acoustic Source Localization in Reverberant Environments. Audio, Speech, and Language Processing, IEEE Transactions on Volume: 19 , Issue: 6.
- [13] Itoyama, Katsutoshi Goto, Masataka Komatani, Kazunori Ogata, Tetsuya Okuno, Hiroshi G. Graduate School of Informatics, Kyoto University, Japan, (2011). Simultaneous processing of sound source separation and musical instrument identification using Bayesian spectral modeling. Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on 22-27 May 2011.
- [14] Huijbregts, Marijn McLaren, Mitchell van Leeuwen, David, (2011). Unsupervised acoustic subword unit detection for query-by-example spoken term detection. Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on 22-27 May 2011, page(s): 4436 – 4439, Prague, Czech Republic.
- [15] Sarathy, K.P.; Ramakrishnan, A.G.; (2008). A research bed for unit selection based text to speech synthesis, Spoken Language Technology Workshop, 2008. SLT 2008. IEEE, Page(s): 229 – 232.
- [16] Han Yang; Yuanyuan Pu; Hong Wei; Zhengpeng Zhao; (2004). An acoustic-phonetic analysis of large vocabulary continuous Mandarin speech recognition for non-native speakers, Chinese Spoken Language Processing, 2004 International Symposium on 2004. Page(s): 241 – 244.
- [17] N.A. Abdul-Kadir and R. Sudirman, (2011). Vowel Effects towards Dental Arabic Consonants based on Spectrogram, IEEE Second International Conference on Intelligent Systems, Modelling and Simulation, IEEE Computer Society 2011.
- [18] Kensaku Fujii, Takuto Yoshioka, Kana Yamasaki, Mitsuji Muneyasu and Masakazu Morimoto, (2011). A Double Talk Control Method Improving Estimation Speed by Adjusting Required Error Level, Workshop on Hands-free Speech Communication and Microphone Arrays, IEEE May 30 - June 1, 2011 .
- [19] Juraj Kac and Gregor Rozinaj, (2009). *Adding Voicing Features Into Speech Recognition Based on HMM in Slovak*, IEEE Conference, Systems, Signals and Image Processing, IWSSIP 2009. 16th International Conference.
- [20] Daniel Jurafsky & James H. Martin, University of Colorado, Boulder, (2008). *Speech and Natural Language Processing*. 2nd edition, Prentice Hall.
- [21] Yan Zhou and Li Shang, (2012). Speaker Recognition Based on Principal Component Analysis and Probabilistic Neural Network, Lecture Notes in Computer Science, 2012, Volume 6839, Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Pages 708-715.
- [22] Tobias Herbig, Franz Gerl, and Wolfgang Minker, (2012). Self-learning speaker identification for enhanced speech recognition. Computer Speech & Language, Volume 26, Issue 3, June 2012, Pages 210–227.
- [23] Silke M. Witt, (2012). Automatic Error Detection in Pronunciation Training: Where we are and where we need to go. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012 KTH, Stockholm, Sweden , (IS ADEPT, Stockholm, Sweden, June 6-8 2012).

- [24] Bryan Pellom, (2012). Rosetta Stone ReFLEX: Toward Improving English Conversational Fluency in Asia. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012 KTH, Stockholm, Sweden, (IS ADEPT, Stockholm, Sweden, June 6-8 2012).
- [25] Bernard Smith (2011). Arabic Speakers: Learner English, Cambridge Handbooks for Language Teachers, 2nd Edition, Series Editor Scott Thornbury.
- [26] ZHU, J., WANG, H., HOVY, E. H. (2010). Confidence-based Stopping Criteria for Active Learning for Data Annotation. ACM Transactions on Speech and Language Processing, Vol. 6, No. 3, Article 3, Publication date: April 2010.
- [27] ZHU, J., WANG, H., and HOVY, E. H. (2008). Multi-Criteria-Based strategy to stop active learning for data annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics*. 1129–1136.
- [28] Kevin Knight and Jonathan May, (2009). Handbook of Weighted Automata, Edited by Manfred Droste, Werner Kuich, Heiko Vogler, Springer. Chapter 14: Applications of Weighted Automata in Natural Language Processing.
- [29] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 858–867, Prague, June 2007.
- [30] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What’s in a translation rule? In HLT-NAACL Proceedings, 2004.
- [31] Daniel Gildea. Loosely tree-based alignment for machine translation. In ACL Proceedings, Sapporo, Japan, 2003.
- [32] Jonathan Graehl and Kevin Knight. Training tree transducers. In HLT-NAACL Proceedings, 2004.
- [33] Kevin Knight and Jonathan Graehl. An overview of probabilistic tree transducers for natural language processing. In CICLing Proceedings, 2005.
- [34] Shankar Kumar and William Byrne. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In HLT-NAACL Proceedings, 2003.
- [35] Jonathan May and Kevin Knight. A better n-best list: Practical determinization of weighted finite tree automata. In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pages 351–358, New York City, USA, June 2006. Association for Computational Linguistics.
- [36] Jonathan May and Kevin Knight. Tiburon: A weighted tree automata toolkit. In Oscar H. Ibarra and Hsu-Chun Yen, editors, Proceedings of the 11th International Conference of Implementation and Application of Automata, CIAA 2006, volume 4094 of Lecture Notes in Computer Science, pages 102–113, Taipei, Taiwan, August 2006. Springer.
- [37] I. Dan Melamed. Multi-text grammars and synchronous parsers. In NAACL Proceedings, 2003.
- [39] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In NAACL-HLT Proceedings, 2003.
- [40] Ivan A. Sag, Thomas Wasow, and Emily M. Bender. Syntactic Theory. CSLI Publications, 2nd edition, 2003.
- [41] Stuart M. Shieber. Synchronous grammars as tree transducers. In TAG+ Proceedings, 2004.
- [42] Stuart M. Shieber. Unifying synchronous tree adjoining grammars and tree transducers via bimorphisms. In EACL Proceedings, 2006.
- [43] Bowen Zhou, Stanley F. Chen, and Yuqing Gao. Folsom: A fast and memory efficient phrase-based approach to statistical machine translation. In Proceedings of the IEEE/ACL 2006 Workshop on Spoken Language Technology, pages 226–229, Palm Beach, Aruba, December 10–13 2006.
- [44] Sherif Abdou, Mohsen Rashwan, Hassanin Al-Barhamtoshy, Kamal Jambi, and Wajdi Al-Jedaibi, 2012. *Enhancing the Confidence Measure for an Arabic Pronunciation Verification System*. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012, KTH, Stockholm, Sweden.

Adaptive English Pronunciation Errors for Arab Learners of English

Hassanin Al-Barhamtoshy¹, Kamal Jambi², Wajdi Al-Jedaibi³

Faculty of Computing & Information Technology
King Abdulaziz University, Saudi Arabia

1 hassanin@kau.edu.sa

2 kjambi@kau.edu.sa

3 waljedaibi@kau.edu.sa

Sherif Abdou*¹, Mohsen Rashwan**²

* Faculty of Computers at Cairo University Egypt

** Faculty of Engineering at Cairo University Egypt

1 sheriff.abdou@rdi-eg.com ,

2 mrashwan@rdi-eg.com

Abstract - The paper introduces the problems of detecting and correcting accent defects, and also summarizes typical Arabic speakers in English spoken language. Then, it introduces the essential probabilistic architecture that will be used to solve both accent defects and pronunciation problems.

Consequently, some of important kinds of variations in pronunciation that are important for speech recognition are carried out and presents accent defects describing such variations. Therefore, high quality speech algorithms need to know when to use particular pronunciation variants. Also, the paper focuses on the essential programming algorithm and various instantiations and also a probabilistic of artificial neural network, HMM, or weighted augmented transition network called *weighted ATN*.

The proposed method improves the accuracy of correct speaker. Using the *SpeakCorrect*, 100 hours from speakers phonetically prerecorded are used to create pronunciation training database. HMM and the weighted ATN models are trained using such prerecorded sets. Their speeches are very clear, acceptable as natural speeches. The proposed framework is optimized to suit embedded the phonetic pronounced database. This proposed frame work has been tested for Saudi and Egyptian accents.

This paper is a practical reference guide for teaching of English as a foreign language (Saudi and Egyptian accents). It is used to help learners to anticipate the characteristic difficulties of English who speak Arabic mother tongues, and how these difficulties overcome.

1. INTRODUCTION AND KEY DEFINITIONS

Any speech engine is composed of two or three sub-systems; the first is word recognizer that converts any given utterance into its underlying sequence of words. The second is phoneme recognizer based on HTK, which converts the given utterance into a sequence of phonemes. This sequence is then processed using matching algorithm and its most important keywords are extracted.

Automated speech recognition and interaction with the user is an essential speech applications system. Voice-driven navigation devices and hands-free telephony are other applications where the user is enabled to enter telephone numbers by speaking sequences of digits, or to enter city names for navigation.

The acoustic input O is treated as sequence of individual “symbols” or “observations” (by slicing up the input every 10ms, and representing each slice by values or frequencies of that slice). Each index represents time interval for slices of the input, therefore; capital letters will stand for sequences of symbols and lower-case letters represents symbols: $O = o_1, o_2, o_3, \dots, o_t$

Similarly, sentence will be treated as string of words: $W = w_1, w_2, w_3, \dots, w_n$.

Some of general terminologies used throughout this document, are explained in this section, [1]-[4].

A. Approach and Methodology

The paper of the project is prepared especially for participating non-specialist learner/student who needs to speak English correctly. Technical linguistic terminology has been kept to a minimum, and proposed system has in general aimed at producing clear simple descriptions of usage rather than detailed of evaluation in visual output. This is particularly the case in the area of pronunciation, where excessive technical detail can be confusing for the non specialist. Within this approach, however, we believe that the descriptions, the technical and the technology points given here are valid and reasonably comprehensive.

The activities of this paper involve designing, implementing, and testing a prototype system that can be delivered and can support the daily teaching-based activities of the adult students. This system is not intended to be a complete substitute for the human class teacher but will be an auxiliary tool to help him/her teach the basic skills of reading, and participate in the different fields of comprehensive development.

In order to achieve the propose system, research and development is needed in five main components:

1. Developing high quality multimedia based lessons and comprehension questions.
2. Developing a reading training module
3. Developing a recording training module
4. Developing educational evaluation module
5. Integrating the developed modules in a deliverable application prototype.

The project has been designed to keep things simple even when dealing with complexity. Following this approach a simple structure has been drawn that comprises the main activities of speech processing. Therefore the project will be structured in the following set of work packages:

- 1- Design the applications according to the user requirements
- 2- Test a retype before the final release
- 3- Get the experience of previous similar projects

B. Key Definitions

Phoneme: the smallest unit of speech is a phoneme; it is used to distinguish meaning. The phoneme is the important unit in the word, each word consists of phonemes, and replacing them causes change in the meaning of a word. If the sound [b] is replaced by [p] in the word “pin”, the word changed to “bin”. Therefore /b/ is a phoneme [20].

Phone: It is the smallest physical segment of sound. And therefore, phones are the physical realization of phonemes. Also, **allophones** are described as phonic variety of phonemes [25].

Phonetics: is the study of human speech, and is concerned with properties of speech sounds.

Phonology: is used to study sound systems and to abstract sound units; i.e., phonemes and phonological rules. Therefore, phonetics definitions apply across languages, and phonology is language based. The phonetic of a sound represented by [20], and the phoneme is represented using //.

Syllable: is defined as a unit of pronunciation. It is generally larger than a single sound and smaller than a word. The *syllable* is start and end with mostly consonants, and is made up using vowels.

C. Literature Review and Related Works

The paper of Macherey and et. al.; (2009) investigates two statistical methods for spoken language understanding based on statistical machine translation. The first approach employs the source-channel paradigm, whereas the other uses the maximum entropy framework. Starting with an annotated corpus, it describes the problem of translation from a source sentence to a target sentence. Also, it analyzes the quality of different alignment models and feature functions and shows that the direct maximum entropy approach outperforms the source channel-based method. Finally, it investigates a new approach to combine speech recognition and spoken language understanding. For this purpose, it employs minimum error rate training which directly optimizes the final evaluation criterion. Experiments were carried out on two German inhouse for spoken dialogue systems [1].

Computational semantics performs a conceptualization of the world for composing a meaning representation structure from available signs and their features present, for example, in words and sentences (De Mori and et. al., 2008), [2]. Spoken language understanding (SLU) is the interpretation of signs conveyed by a speech signal. SLU and natural language understanding (NLU) share the goal of obtaining a conceptual representation of natural language sentences. Signs are used for interpretation and can be coded into signals along with other information such as speaker identity. Furthermore, SLU systems contain automatic speech recognition (ASR) component and must be robust to noise due to the nature of spoken language and the errors introduced by ASR.

Dinarelli and et. al.; (2010) presents a call routing application for complex problem solving tasks. Up to date work on call routing has been mainly dealing with call-type classification, [3]. In this paper they take call routing further: Initial call classification is done in parallel with a robust statistical Spoken Language Understanding module. This is followed by a dialogue to obtain further task-relevant details from the user before passing on the call. The dialogue capability also allows them to obtain clarifications of the initial classifier guess. Based on an evaluation, they show that conducting a dialogue significantly improves upon call routing based on call classification alone. They present both subjective and objective evaluation results of the system according to standard metrics on real users.

The paper of Camelin and et al.; (2010) describes a system for automatic opinion analysis from spoken messages collected in the context of a user satisfaction survey. A process is used for detecting segments expressing opinions in a speech signal. Methods are proposed for accepting or rejecting segments from messages that are not reliably analyzed due to the limitations of automatic speech recognition processes, for assigning opinion hypotheses to segments and for evaluating hypothesis opinion proportions.

Specific language models are introduced for representing opinion concepts. These models are used for hypothesizing opinion carrying segments in a spoken message, [4]. The different processes are trained and evaluated on a telephone corpus collected in a deployed customer care service. The proportions estimated with such a low divergence are accurate enough for monitoring user satisfaction over time.

Automatic segmentation and classification of dialog acts is important for spoken language understanding (SLU: Laskowski, Kornel; Shriberg, Elizabeth; 2010). Such paper proposes a framework for employing both speech/non-speech-based (“contextual”) features and prosodic features, and applies it to segment and classify in multiparty meetings. They find that: (1) contextual features are better for recognizing, while prosodic features are better for finding base mechanisms and backchannels; (2) the two knowledge sources are complementary for most of the studied types; and (3) the performance of the resulting system approaches that achieved using oracle lexical information for several types, [5].

Natural human-robot interaction (HRI) requires different and more robust models of language understanding (NLU) than non-embodied NLU systems as it is described in Cantrell and et. al., 2010. In particular, architectures are required that (1) process language incrementally; (2) use pragmatic contexts throughout the understanding process to infer missing information; and (3) handle the underspecified, fragmentary, or otherwise ungrammatical utterances that are common in natural speech. In this paper [6], they describe attempts at developing an integrated natural language understanding architecture for HRI, and demonstrate its novel capabilities using challenging data collected in human-human interaction experiments.

Using unstructured queries to search a structured database in voice search applications addressed by Young-In and et. al., (2009). By incorporating structural information in music metadata, the end-to-end search error has been reduced by 15% on text queries and up to 11% on spoken queries. Based on that, an HMM sequential rescoring model has reduced the error rate by 28% [7] on text queries and up to 23% on spoken queries compared to the baseline system. A phonetic similarity model has been introduced to compensate speech recognition errors, which has improved the end-to-end search accuracy consistently across different levels of speech recognition accuracy.

In the paper of Heracleous and et. al.; (2009), Hidden Markov Models (HMM)-based vowel and consonant automatic recognition in cued speech for French are presented. Cued speech is a visual communication mode which uses hand-shapes in different positions and in combination with lip-patterns of speech, makes all the sounds of spoken language clearly understandable to deaf and hearing-impaired people. The aim of cued speech is to overcome the problems of lip reading and thus enable deaf children and adults to fully understand a spoken language. This study investigates and also reports automatic consonant recognition experiments in cued speech for French. In addition, isolated word recognition experiments both in normal-hearing and deaf subject are presented, showing a promising word accuracy of 92% on average, [8].

Using speech transformation to increase speech intelligibility for the hearing- and speaking-impaired is presented using two speech transformation approaches designed to increase the intelligibility of speech, [9]. The first approach is used in the context of increasing the intelligibility of conversationally spoken speech for hearing-impaired listeners. The second approach aims to increase the intelligibility of speaking-impaired individuals by the general population. Results of listening tests indicated that an intelligibility increase was not achieved; listeners preferred the transformed speech of the proposed system over that of an alternative system.

Mobile augmented reality (AR) translation system, using a smart phone’s camera and touch-screen, that requires the user to simply tap on the word of interest once in order to produce a translation, presented as an AR overlay (Victor Frago and et. al., 2011), [10]. The translation replaces the original text in the live camera stream, matching background and foreground colors estimated from the source images. For this purpose, they developed an efficient algorithm for accurately detecting the location and orientation of the text in a live camera stream that is robust to perspective distortion, and they combine it with OCR and a text-to-text translation engine. Their experimental results, using the ICDAR 2003 dataset and their own set of video sequences, quantify the accuracy of detection and analyze the sources of failure among the system’s components. With the OCR and translation running in a background thread, the system runs at 26 fps on a current generation smart-phone (Nokia N900) and offers a particularly easy-to-use and simple method for translation, especially in situations in which typing or correct pronunciation (for systems with speech input) is cumbersome or impossible [10].

The paper of [11] presents a simple and efficient feature modeling approach for tracking the pitch of two active speakers. The paper indicates that, they employ the mixture maximization model (MIXMAX)

in added to linear interaction model. A factorial hidden Markov model is applied for tracking pitch over time. This statistical model can be used for applications beyond speech, whenever the interaction between individual sources can be represented as MIXMAX or linear model. They demonstrate experimental results using Mocha-TIMIT as well as data from the speech separation challenge. The paper showed the excellent performance of the proposed method in comparison to a well-known multi-pitch tracking algorithm. Using speaker-dependent models, the proposed method improves the accuracy of correct speaker assignment, which is important for single-channel speech separation. Moreover, they demonstrate the beneficial effect of correct speaker assignment on speech separation performance.

Particle filtering has been shown to be an effective approach to solving the problem of acoustic source localization in reverberant environments, [12]. In such environment multiple-hypothesis model associated with these arrivals can be used to alleviate the unreliability often attributed to the acoustic source localization problem. Recently, the extended Kalman particle filter (EPF) scheme for the localization problem. Due to this, the extension of the multiple-hypothesis model for this scheme is not trivial. In this paper [12], the EPF scheme is adapted to the multiple-hypothesis model to track a single acoustic source in reverberant environments. Such work is supported by an extensive experimental study using both simulated data and data recorded in their acoustic lab. Various algorithms and array constellations were evaluated. The results demonstrate the superiority of the proposed algorithm in both tracking and switching scenarios. It is further shown that splitting the array into several sub-arrays improves the robustness of the estimated source location.

The paper of [13] presents a method of both separating audio mixtures into sound sources and identifying the musical instruments of the sources. A statistical tone model of the power spectrogram, called an integrated model, is defined and source separation and instrument identification are carried out on the basis of Bayesian inference. Since, the parameter distributions of the integrated model depend on each instrument, the instrument name is identified by selecting the one that has the maximum relative instrument weight. Experimental results showed correct instrument identification enables precise source separation even when many overtones overlap.

In the paper of [14], they present a method for automatically generating acoustic sub-word units that can substitute conventional phone models in a query-by-example spoken term detection system. They generate the sub-word units with a modified version of their speaker diarization system. Given a speech recording, the original diarization system generates a set of speaker models in an unsupervised manner without the need for training or development data. Modifying the diarization system to process the speech of a single speaker and decreasing the minimum segment duration constraint allows detecting speaker-dependent sub-word units. For the task of query-by-example spoken term detection, they show that the proposed system performs well on both broadcast and non-broadcast recordings, distinct conventional phone-based system trained solely on broadcast data. A mean average precision of 0.28 and 0.38 was obtained for experiments on broadcast news.

Another paper [15] describes a modular, unit selection based TTS framework, which can be used as a research bed for developing TTS in any new language, as well as studying the effect of changing any parameter during synthesis. Using this structure, TTS has been developed for Tamil. Synthesis database consists of 1027 phonetically rich prerecorded sentences. This framework has already been tested for Kannada. Their TTS synthesizes intelligible and acceptably natural speech, as supported by high mean opinion scores. The framework is optimized to suit embedded applications like mobiles and PDAs. They compressed the synthesis speech database with standard speech compression algorithms used in commercial GSM phones. Even with a highly compressed database, the synthesized output is perceptually close to that with uncompressed database. Through experiments, they explored the ambiguities in human perception when listening to Tamil phones and syllables uttered in isolation, thus proposing to exploit the misperception to substitute for missing phone contexts in the database. Listening experiments have been conducted on sentences synthesized by replacing phones with their confused ones.

The paper of [16] addresses non-native accent issues in large vocabulary continuous speech recognition. It proposes to analyze the transformation rules of non-native Mandarin speech spoken by native speakers of *Naxi* and *Dai* in *Yunnan* at the level of initials and finals. Firstly, baseline HMM models are trained using the standard Mandarin corpus to test their performance on non-native speech recognition. Secondly, the non-native speech data is transcribed based on the baseline HMM models. In more detail, they analyze the error recognition rates of all initials and all finals, and their typical substitute error. The results obtained from their experiments might be useful for adapting a native speaker ASR system to model nonnative accented data.

The paper with the title of “Vowel Effects towards Dental Arabic Consonants based on Spectrogram” [17] discussed the effect of vowel (fatha, kasra and damma) in Arabic consonants. These vowels are added to the basic consonants with three simple diacritics using the utterances of every dental consonant concerned by Malaysian children. The dental consonants refer to the consonants utter using dental medium. It is called the place of articulation at labiodentals, dental and interdental. The formant frequencies produced for each place of articulation are based on their spectrogram. The visualization of the spectrogram makes the formants easily being identified by normal human being’s vision. The formants (F1, F2 and F3) are averaged and the results show the different increment and decrement compare to utterances of single phoneme. F1 is decreased for all observed consonants. F2 and F3 changes according to its manner of articulation, where in this study the tabulation are plosive and fricative. The paper noted that, most of the speakers having decreasing frequency of the second and third formant for all consonants pronunciation with vowel effect.

Kensaku Fujii and et. al. [18] proposed a step size control method capable to cancel acoustic echo even when double talk continues from echo path change. This method controls the step size by substituting the difference between the coefficients of a main adaptive filter (Main-ADF) and a sub-adaptive filter (Sub-ADF) for the estimation error in the former. The speed size control can be improved by utilizing the difference for the step size control. The paper shows that in single talk within the proposed method can provide almost the estimation speed as the method whose step size is fixed at the optimum one and verify that even in double talk the estimation error quickly decreases.

However *Juraj Kac* and *Gregor Rozinaj* discussed the impact of substituting some of the basic speech features with the voiced/ unvoiced information and possibly with the estimated pitch value, [19]. As a good measure of the signal’s voicing the average magnitude difference function was assumed, especially the ratio of its average value to its local minima found within the accepted ranges of the pitch. Furthermore, the pitch itself was used as an auxiliary feature to the base speech features. Experiments were performed on the professional database for mobile applications working in harsh conditions, using various HMM models of context dependent and independent phonemes. All models were trained following the training scheme. In all cases the voicing feature brought improved results by more than 9% compared to the base systems. However the role of the pitch itself in the case of speaker independent ASR system evaluated over different tasks was not always so beneficial.

2 PRONUNCIATIONS

This section illustrates how pronunciation can vary, and how can realize a phoneme as different allophones in phonetic environment. This section has also shown how to write transducer rules to model these changes for speech, also accent defects, pronunciation errors and common pronunciation errors will be presented.

Lexical variation and allophonic variation are two classes of pronunciation variation. The lexical variation is used to represent the word in the lexicon, while the allophonic variation is a difference in how individual segments change their value in different segments, [20]. Most of the variation in pronunciation is allophonic, according to the influence of surrounding sounds and syllable structure.

The lexical variation is related to sociolinguistic variation. It is due to extra linguistic factors such as dialect variation. Other socio linguistic differences are due to register or style rather than dialect. One of the most well-studied example of style-variation is the suffix -ing (as something), which can be pronounced somethin (without g) [20, 25].

The proposed rules are dependent on a complicated set of factors that must be interpreted probabilistically. Most allophonic rules relating English can be grouped into number of types: assimilation, dissimilation, deletion, flapping, vowel reduction, and epenthesis (insertion of an extra sound into a word), [20].

Assimilation is the change in segment sound to make it more like a neighboring segment, e.g.; dentalization and palatalization. As an example of palatalization rule as follows:

$$\left\{ \begin{array}{l} [s] \\ [z] \\ [t] \\ [d] \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} [ʃ] \\ [ʒ] \\ [tʃ] \\ [dʒ] \end{array} \right\} / - \{y\}$$

Deletion is the removal of a sound from a word. The following rule includes /t/ and /d/ is deleted before consonants:

$$\begin{Bmatrix} t \\ d \end{Bmatrix} \Rightarrow \theta / V - C$$

The **flapping** is happening when the speaker is speaking more quickly, and is more likely to happen at the end of a word when it forms a collection. The most important phonological process is vowel reduction or reduced vowel.

A. Accent Defects and Pronunciation Error Categorization

Some errors that may be noticeable will not cause any difficulties in understanding from a native listener's point of view, whereas other types of errors will cause serious problems for the intelligibility of an utterance, especially for the non native speakers. Regardless of what the final aim is, any learner will benefit from realizing the impact of various errors.

According to previous studies and works; this paper investigates pronunciation difficulties in second language learners from Egypt and Saudi Arabia citizens, with English as target language. The main motivation for this work was to create guidelines for teachers of English as a foreign language. In order to get recordings that were representative for each language group, and that covered all aspects of pronunciation without making the material.

1) **General Phonology.** The Arabic and English phonological systems are different, in the range of sounds used in emphasizes of the position on vowels and also in consonants in expressing meaning.

English has 22 vowels and diphthongs to 24 consonants. Arabic has only eight vowels and diphthongs to 32 consonants [25]. The Arabic vowels include three short, three long and two diphthongs. Therefore, Arabic speakers tend to gloss over and confuse English short vowel sounds, while unduly emphasizing consonants, avoiding elisions and shortened forms [16, 17, 25]. Within each country, a wide variety of colloquial dialects have developed, differing one from another in pronunciation, common lexical items, and in structure, [25].

The informants made recordings that were both read speech, and free speech guided by pictures and sequences of pictures. Both sentences and isolated words were recorded in each of the two categories. The read texts were designed to cover various aspects of the pronunciation accented errors and highlight pronunciation difficulties.

A comprehensive table listing the difficulties for each of the category, and in addition attention was given to re-occurring difficulties, as well as a categorization of errors based on the seriousness from an intelligibility point of view. Based on this analysis supervisor of acoustic (Adel Al-Sheikh) sorted errors on an intelligibility scale as a guideline for what aspects of pronunciation should be prioritized in pronunciation teaching. The most serious errors in ascending order according to Al-Sheikh are shown in Table 1.

TABLE 1
THE MOST SERIOUS ERRORS FOR LEARNERS' OF ENGLISH (EGYPTIAN AND SAUDI ARABIAN PEOPLES)

No.	Error type or Category	Example
1	Labial-dental Fricatives:	Substituting: /f/ and /v/. - /v/ for /f/.
2	Rolling the	Library, Ruler, Lorry, Liberian, and Reroofing
3	Replacing	Replace /θ/ with /s/, as in sin for thin. Replace /tʃ/ with /ʃ/ as in sheep for cheap
4	Dental Fricatives	Dental Fricatives: /θ, ð /
5	Pronouncing the grapheme	Pronouncing /g/ only as /ʒd/ or as in gentle/ʒd /. Bilabial Plosives: /p, b/. Alveolar Fricatives: /s, z/.

The initial work on creating pronunciation error detectors for the proposed framework is inspired by paper work of as will be described in the table (2).

TABLE 2
THE MOST SERIOUS ERRORS FOR LEARNERS' OF ENGLISH

No.	Error Type or Category	Error Description
1	Lexical stress	Insufficient stress marking, or stress on the wrong syllable
2	Syllable structure	Incorrect number of syllables in a word.
3	Consonant clusters	Vowel insertion (epenthesis) in, or before a consonant cluster, or consonant deletion in a consonant cluster before a stressed vowel.
4	Rhythm	The relationship between stressed and unstressed syllables in a sentence is wrong

To complete speech correct tasks, there are many sources of “defects, pronounced errors and acoustic variation” – in Saudi and Egyptian accents. For spelling error detection, what we mean by defects in

pronounced text, which mask the correct spelling of the text. Therefore, the following subsection breaks the field down into four increasingly boarder problems:

1. Substituting /v/ for /f/, such as saying the word: vat/fat, very/ferry, belief/ believe, vast/fast and van/fan.
2. Rolling the /r/ as an examples: Library, Ruler, Lorry, Liberian, and Reroofing.
3. Replacing /θ/ with /s/, as in sin for thin: thong/ song, thank /sank , theme/seem, / thin sin and thought / sought.
4. Dental Fricatives: /θ, ð /: Replacing /ð/ with /z/ or /d/, as in dat or zat for that, and / ð / with /θ/. Therefore Ss may replace the / ð / sound as in “brother”, “they” and “these”, with the /θ/ sound. As examples: another, blithering, bother, brother, and father.

Other researches discuss kinds of spelling error patterns that occur in typed text and speech-recognition [20, 22]. Single-error misspellings induced by one the following errors: insertion, deletion, substitution, and transposition. While typing errors are usually characterized as substitutions, insertions, deletions, or transpositions, OCR errors are usually grouped into five classes: substitutions, multi-substitutions, space deletions or insertions, and failures, [20].

- 2) **Vowels.** Some of English phonemes have equivalent or near equivalent in Arabic and therefore be perceived and articulated without great difficulties. Some English phonemes may cause problems, the following are most confusion, [25]:
 - a. /e/ and /ɪ / are often confused; for example bit for bet.
 - b. The two phonemes /ɒ/ and /ɔ:/ are often confused ; e.g.; cot for caught.
 - c. Diphthong /əʊ / and /eɪ/ are usually pronounced rather short, and may are confused with /e/ and /ɒ/; as an example red for raid.
- 3) **Consonants.** Many of English phonemes have equivalent or near equivalent in Arabic and therefore can be articulated without difficulties. Although some confusion may still arise, few of phonemes may cause problems, the following comments illustrate examples of such problems:
 - a. The Arabic letter /g/ is pronounced as /g / in Egyptian accent, and /dʒ / in Saudi accent, and sometimes even as /j/ according to local dialects.
 - b. The two letters /v/ and /f/ are often confused, especially in Saudi accent; e.g.; It is a fery nice fillage.
 - c. The two allophones /p/ and /b/ tend to be used rather randomly: I baid ten bense for a bicture.
 - d. /θ/ and /ð/ occur and dialect pronounce them as /t/ and /d/ respectively- especially in Egyptian accent- I tink dat dey
 - e. The rolling of /r/ is voiced flap, Arabic speakers overpronounce the post-vocalic r; as in car park.
 - f. Sometimes /g/ and /k/ are often confused, especially whose dialects do not include the phoneme /g/, as in goat/coat and bag/bak.
- 4) **Consonant Clusters.** The number of consonant clusters occurring in English is greater than in Arabic. Initial two segment clusters rarely occurring in Arabic [25]: pr, pl, gr, gl, thr, thw, and sp.

Initial three-segment clusters do not occure in Arabic: spr, skr, str, and spl. According to these clusters, there is tendency among Arabic speakers to insert short vowels to assist pronunciation:

ispring or sipring for spring.
perice or pirice for price.

And also, for the range of final clusters [25]:

monthiz for months
Neckist for next.

5) **Rhythm, Stress and Intonation.** Arabic speakers have problems grasping the unpredictable nature of English word stress, because the Arabic is stress-timed language, and word stress is predictable and regular.

Rhythm is similar in Arabic and English, and sometimes causes few problems. Primary stress, occur in Arabic and unstressed syllables are pronounced more clearly in English.

Whereas intonation patterns are similar in Arabic and English - using rising tune (questions, suggestions, etc.).

B. Common Pronunciation Errors

Figure 1 illustrates almost pronunciation errors that are needed to be addressed in successful training and assessment models, [23]. As shown in the figure, it can be classified into phonemic and prosodic error types.

- (1) The phonemic errors can be divided into substituted, deleted or inserted. Also, there are errors on a little scale “where the correct phoneme is more or less being spoken”, [23].
- (2) The prosodic errors can be categorized into stress, rhythm and intonation.

Therefore, these two types of errors make pronunciation a multi-dimensional problem. Consequently, large number of metrics is used to measure these dimensions [23, 24].

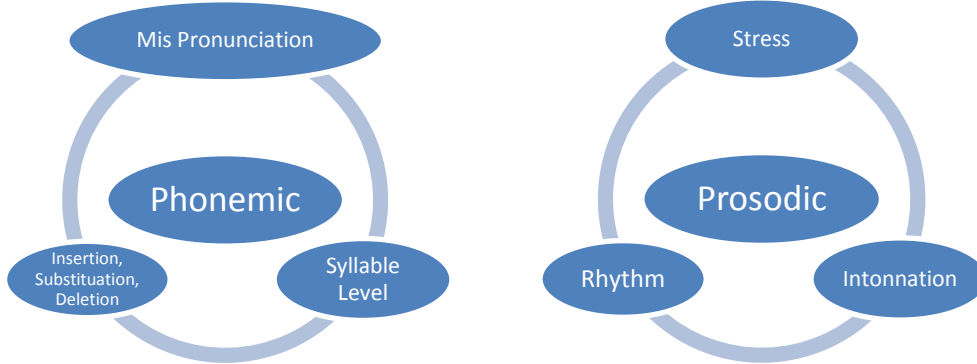


Figure (1): Classification of Pronunciation Errors [23]

During the development of Speak Correct we found a significant body of literature describing the typical patterns of error made by Korean Learner Segmental Errors (KLEs), [24]. A pilot corpus is collected and phonetically annotated consisting of prompted English speech data from an assortment of different types of content. The corpus includes short paragraphs of text, sentence prompts, and words with particularly difficult consonant clusters (e.g., refrigerator). In total, the pilot corpus collected 25,000 total speech samples from 111 learners who reside in Korea. The corpus provides a direct comparison of realized phone sequences compared to expected canonical sequences from native speakers. A summary of Korean to English speakers also makes no distinction between fricative /f/ and /v/, substitute /p/ and /b/ instead. Other common reported errors include the substitution of aspirated /t/ for /θ/ and un-aspirated /t/ for /ð/, [23]. Table (3) illustrates the most frequent segmentation observed errors, [24].

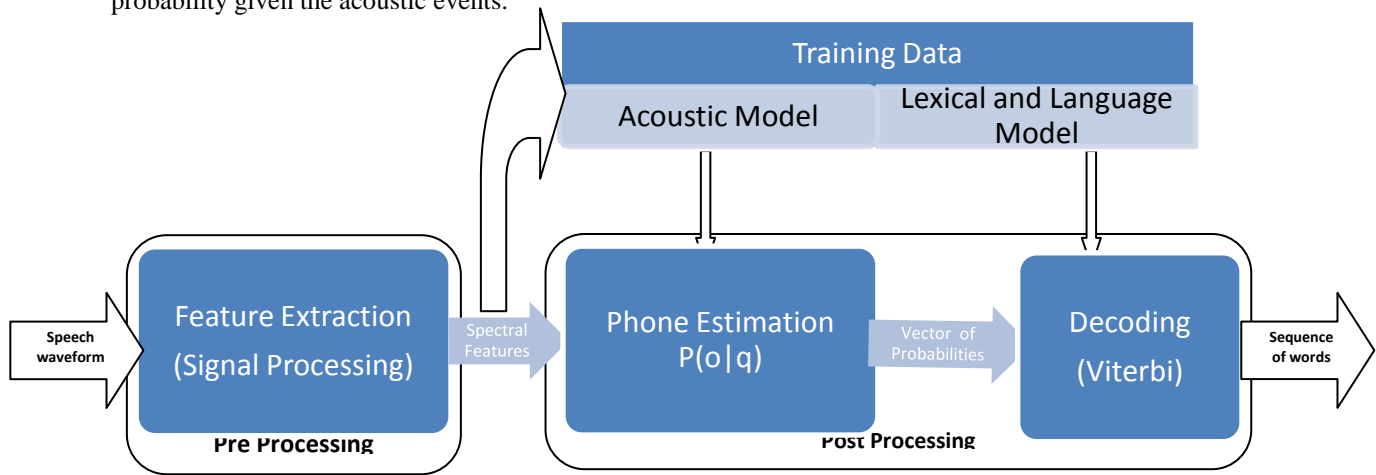
TABLE 3
FREQUENT KOREAN LEARNER SEGMENTAL ERRORS [24]

<i>Error Pattern Description</i>	<i>Error %</i>
Unstressed to Stressed Vowel (e.g., /ɪ/ → /i/)	14%
/ɹ/ Issue (deletion, substitution, etc.)	11%
Dental to Alveolar (e.g., /θ/ → /t/)	9%
Consonant Deletion (word initial /j/ → / /)	8%
Vowel Insertion (e.g., large → largɪ)	7%
Diphthong to Monophthong (e.g., /eɪ/ → /ɛ/)	5%
Consonant Cluster Simplification	5%
Vowel to Central Vowel (e.g., /u/ → /ɨ/)	5%
Vowel Deletion (e.g., /ə/ → / /)	4%
/æ/ to /ɛ/	4%

3. SPEAK CORRECT PROCESSING ENGINE

Any speech recognition system consists of two main modules: The first module is called pre-processing or feature extraction and the second module is post-processing divided into acoustic, lexical and language modeling [44]. Figure (2) shows an outline of speech recognition system components. Such system is broken down into three stages. The first stage is used for signal processing or feature extraction,

the acoustic waveform is sliced up into frames which are transformed into spectral features. The second stage, phone estimation that includes statistical techniques (e.g., neural networks or Gaussian models) to recognize individual speech sounds like *f* or *s*. The output of this stage is a vector of probabilities over phones for each frame. The last stage is the decoding to find sequence of words which has the highest probability given the acoustic events.



Figure(2): Architecture for Simplified Speech Recognizer

To summarize the process of extraction features, starting from sound waves and ending with feature vector. First, an input sound wave is *digitized* (it is analog-to-digital conversion), and it has two steps: *Sampling* and *quantization*.

The *sampling* rate is the number of samples taken per second (8000 Hz and 16000 Hz are two common sampling rates). At least two samples in each cycles are needed to measure a wave accurately: one for positive part of the wave and one for the negative part, and more than two samples per cycle increases the amplitude accuracy.

At each sampling rate (8000 Hz or 16000 Hz), there are amplitude values for each second of speech. The process of representing such values as integers is called *quantization*. After the *digitization* of the waveform, it is converted to set the spectral features. It is possible to use any popular feature set (Linear Predictive Coding (LPC) or Perceptual Linear Predictive (PLP)) directly to observe symbols of an HMM [1, 6, 7, 21]. Further processing is often done to the features; like cepstral, which are computed from the LPC coefficients by taking the Fourier transform of the spectrum.

The *phones estimation* is an efficient way to compute the likelihood of an observation sequence given weighted automata. HMM allows us to sum multiple paths that each account for the same observation sequence.

The *decoding* stage is the problem of finding determining the correct “*underlying*” sequence of symbols. Therefore, the *Viterbi* algorithm is an efficient way of solving the decoding problem by considering all possible strings and using addition rules (like Bays rule [20]) to compute their probabilities of generating the observer sequence.

A. Speak Correct Background Architecture

Many of researchers have introduced many of core algorithm used in speech recognition. Therefore, the notions of phone and syllable are introduced [8, 11, 19]. In addition, N-gram language model and the Hidden Markov Model (HMM) discussed in more details [8,11].

HMM were first described by Leonard E. Baum and others in 1960s. The HMM is stochastic methods to model temporal pattern recognition and sequence data. One of the first application fields of HMMs was speech recognition in the med of 1970s. Therefore, tutorial on HMMs was published by Lawrence R. Rabiner, so, analysis of biological sequences (DNA) began to be applied at the second half of 1980s. The HMMs can be illustrated using finite state machines, at each transition there is an observation from specific state, for each state there is output symbol emission. Figure (3) summarizes the overall definition of the HMM, [21].

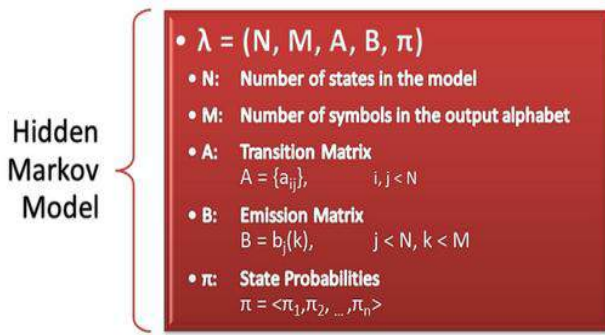


Figure (3-a): HMM Overall Definition [21]

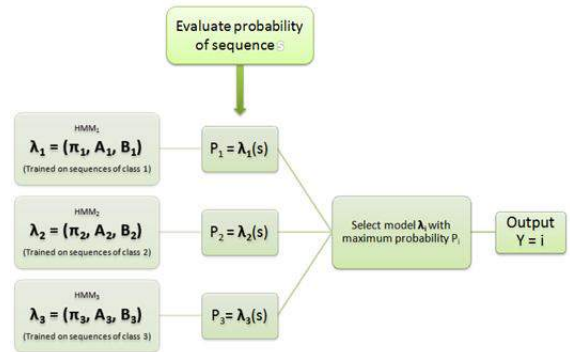


Figure (3-b): HMM Overall Computation [21]

In other words, to choose the word which is most probable given the observation, the single word such that $P(\text{word} | \text{observation})$ is highest. If w is the estimated correct word and if the O is the observed sequence (individual observation), then the equation for picking the best word given is:

$$W = \text{argmax} P(o|w) P(w)$$

Where: $P(o|w)$ represents likelihood, $P(w)$ represents prior, w is vocabulary, w is correct word, o is observation. How to compute these two probabilities will be discussed in the next section.

Once the likelihood- computation has been solved, and decoding problems for a simplified input consisting of strings of phones, feature extraction will quickly be involved.

B. Acoustic Probabilities Counting

As mentioned before, the speech input can be passed through signal processing transformations and converted into series of vectors of features, each vector representing one time-slice of the speech input signal. One of the popular ways to compute probabilities on feature vectors is to first cluster such feature vectors into discrete counted symbols. Therefore, the probability of a given cluster can be calculated (number of times it occurs in some training set).

This methodology is called **vector quantization**; and it is developed into computing observation probabilities or probability density function (pdf). There are two common approaches; **Gaussian pdfs** that maps the observation vector O_t to a probability. The second alternative is the use of neural networks or multi-layer perceptions, that can be trained to assign a probability to speech real-valued feature vector. The neural network is a set of small computation units connected by weighted links. The network is given a vector values and computes a vector of output values.

A standard model based on probabilistic neural network is proposed in [21], it is suitable for testing and pattern classification. The structure of such probabilistic neural network used in this report is shown in figure (4). The number of input speech variable is M , the number of identification patterns are needed is N , the training samples for each patterns are represented by S_1, S_2, \dots, S_N . There are four layers: input layer, model layer, summation layer and output layer, the weights between summation layer and output layer is computed by:

$$W(M) = S_i / \sum_{i=1}^j S_i$$

C. Speak Correct Principles Modules

Many of researches have introduced many core algorithm used in speech recognition. Therefore, the notions of phone and syllable are introduced. This in added to N-gram language model and the Hidden Markov Model (HMM). Our goal is to build a model, so that we can figure out how it modified this “true” word and hence recover it. For the complete speak correct tasks, there are many sources of “defects”: Substituting /v/ for /f/, Rolling the /r/, Replacing /θ/ with /s/, Dental Fricatives: /θ, ð /: Replacing /ð/ with /z/ or /d and acoustic variation due to the channel (Microphone, networks, etc).

Consequently, the operation speed and the use effect should be affected. Therefore, the proposed technology has good ability of eliminating the data correlation, so, a speaker recognition pattern based on the combination of PRS, HMM, ASR, intonation analyzer and pronunciation generator is proposed in this report. The basic recognition processes are as the following.

1) Main Module

Step 1: Gathering and collecting the speech inputting samples.

Step 2: Dividing such samples into two parts, one part is for training samples and the second part is for testing.

2) Training Module

Step 3: Do the following:

- 3.1 Speaker Adaption.
- 3.2 Confidence measuring.
- 3.3 Tuning the native Arabic speaker accent.
 - a. Tuning Saudi accent
 - b. Tuning Egyptian accent.
- 3.4 Intonation training and teaching the prosodic effects.

Step 4: Using the feature vector of training samples to train the *SpeakCorrect* model.

3) Testing Module

Step 5: Do the following steps:

- Step 5.1: Establishing PSR with the associated ATN neural network and HTK mechanism.
- Step 5.2: Inputting the feature vectors of test samples into PSR network which has been trained.
- Step 5.3: Judging the corresponding speech signal category and speaker identity according to the output values. The coming sections include additional description in more details.

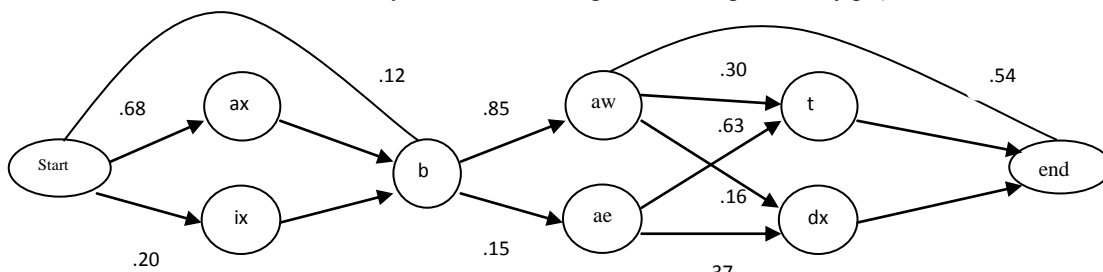
D. Finite State/Weighted Finite State and Weighted ATN/Lattice

Computational linguistics and automata theory were used to predict letter sequences, describe natural language, employ context-free grammars (CFG), introduce the theory of the tree transducers, and parse automatic natural language text [28-35]. In the 1970s, speech recognition researchers captured NLP grammar with weighted finite state acceptors (FSAs), by employing transition weights that could be trained on machine-readable using dictionary, corpus, and corpora [36,37,39-43]. In the remainder of this section, we discuss how natural language applications use tree automata.

In the 1990s, combination of finite state and large training corpora became the dominant paradigm in speech and text processing; software toolkits for weighted finite state acceptors and transducers (WFSTs and WFSTs) were developed [28]. The 21s century has seen generic tree automata toolkits [36] that have been developed to support investigations. The single WFST or augmented transition network (ATN) that represents P(S|E) is still complex, model transformation can be made into chain of transducers in the following: $WFSA_a(\text{English}_{\text{text}}) \leftrightarrow WFSA_b(\text{English}_{\text{sound}})$

Therefore, simple model can be used to calculate 1-gram, 2-gram, and n-gram language model of characters [28]. If an corpus includes 1,000,000 characters, the letter e occurs 127,000 times, the probability P(e) estimated as 0.127. In case of 2-gram model, it can be calculated by remember the previous letter context- its WFS state. As an example the transition between state r and state e outputs the letter e can be calculated by the probability P(e|r). The n-gram model generates more word-like items than (n-1)-gram model does. The weighted ATN is simple automaton in which each arc is associated with a transition, this transition can be represented by probability value, indicating how likely that path is to be taken. The probability on all the arcs leaving a node must sum to 1. Figure 4 shows weighted ATN for the English word “about” which is trained on actual pronunciation example. This model is an instance of a Hidden Markov Model (HMM). It illustrates the behavior transition in the weighted ATN. The rule of the transition according to the following:

- Starts in some initial state (start: s_1) with probability $p(s_i)$,
- On each move goes from state s_i to state s_j according to transition probability $p(s_i, s_j)$.
- At each state s_i , it emits a symbol w_k according to the emit probability $p'(s_i, w_k)$.



Legends: $P(w | ax) = .68$ $P(w | ix) = .20$

Figure (4): A Pronunciation Network (Weighted ATN) for word “about”

The use of *SpeakCorrect* is often called hybrid approach, since it uses elements of the HMM or weighted state-graph representation of the pronunciation of a word, also it uses observation-probability computation using multilayer perception. The input to this multilayer perception is a representation of the signal at a time t , vector of spectral features for time t , and eight additional vectors for times $t+10$ ms, $t+20$ ms, $t+30$ ms, $t+40$ ms, $t-10$ ms, and so on. The network has one output unit for each phone; by summing the values of all the output units to 1, the *SpeakCorrect* can be used to compute probability of a state j given an observation vector O_t , or $P(j|o_t)$, or $P(o_t | q_j)$.

Therefore, receiving the sequence of spoken words that generated a given acoustic speech signal, a standard model- like described in figure 5- is used. The model generates $P(E|S)$ for a received speech signal S , and such model is described as the following:

1. For each word/phonetic in S , a variety of individual units of speech (sequence of phonemes), may be observed with varying probabilities, and therefore, can be interpreted as the word.
2. For each word, a word-to phone is constructed.
3. Each phone can be expressed as a variety of audio signal.

Once defined, the chain of audio signal and the final language model are weighted with the method of likelihood, and observing probabilities from the training data.

E. Training the *SpeakCorrect*

A brief sketch of the embedded training procedure is used in most of ASR systems. Some of the details of the algorithm have been introduced in [8, 11, 16, 21, 22]. Four probabilistic models are needed to train *SpeakCorrect* system:

- Language model probabilities: $P(w_i|w_{i-1} w_{i-2})$
- Likelihood observation : $b_j(o_t)$
- Transition probabilities: a_{ij}
- Pronunciation Lexicon: Weighted ATN of HMM state graph structure.

In order to train the previous probabilities component the *SpeakCorrect* has the following corpuses:

- Training corpus of speech wave files: which are collecting from news web site of the internet, individual peoples ... etc. This speech wave files are collected together with word- transaction.
- Large corpus of text: including the word-transaction from speech corpus together with many other similar texts.
- Smaller training corpus of speech: which is phonetically labeled, i.e. frames are hand-annotated with phonemes.

The HMM lexicon structure is built, by taking an off-the shelf pronunciation dictionary.

Therefore, the training is beginning by run the model on the observation and seeing which transitions and observations were used. Any state can generate one observation symbol; the observation probabilities are all 1.0. The probability p_{ij} of a particular transition between states i and j can be computed by counting the number of transition was taken; $c(i \rightarrow j)$. Normalize such value by using the following:

$$a_{ij} = c(i \rightarrow j) / \sum_{q \in Q} (C(i \rightarrow j))$$

For the weighted ATN and HMM, two methods are used, the **first** idea is to *iteratively* estimate the counts, observation probabilities, and the use such estimated probabilities to derive better and better probabilities. The **second** idea is get estimated probabilities by computing forward probability among all different paths. Define the forward probability in state i after seeing the first t observation, given the automaton A .

$$a_t(i) = P(o_1, o_2, o_3, \dots, o_t, q_t = i | A)$$

Formally, define the following iteration:

1. Initialization:
 $\alpha_n(1) = a_{1j} * b_j(o_1) \dots\dots\dots 1 < j < N$
2. Iteration:
 $\alpha_j(t) = [\sum_{i=2}^{N-1} \alpha_i(t-1) * a_{ij}] b_j(o_t) \dots\dots\dots 1 < j < N, 1 < t < T$
3. Termination:
 $p(o|A) = a_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) * a_{iN}$

The forward algorithm can be run to compute the candidate words was most probable given the observation sequence [ax b], the product $P(o | w) P(w)$ is computed for each candidate word. So, the likelihood of observation sequence o given the word w times the prior probability of the word is computed for each word, and choose the word with the highest value.

The forward algorithm is an edit distance algorithm, it uses a table to store intermediate values as it builds up the probability of the observation sequence. The data is represented in the table by rows oriented; the rows are labelled by state-graph which has many ways of getting from one state to another. The table is filled as a matrix by computing the value of each cell from the three cells around it. Furthermore, the forward algorithm computes the sum of probabilities of all possible paths that could generate the observation sequence.

Each cell of the forward algorithm matrix, $forward[t, j]$ represents the probability of being in state j after seeing the first t observations, given the automaton A . Formally, each cell expresses the following probability:

$$forward [t, j] = P(o_1, o_2 \dots o_t, q_t = j | A) P(w)$$

The following pseudo code describes the forward algorithm applied to any word.

forwardAlgorithm (observation, state-graph)

begin

```

ns = numOfStates(state-graph);
no= length(observation);
/* create probability matrix */
forward [ ns+2 , no + 2 ];
forward [0,0] = 1.0;
foreach time step t from 0 to no do
  foreach states from 0 to ns do
    foreach transition s' from s specified by state-graph
      forward [ s' , t + 1 ] = forward [ s , t ] * a[s , s'] * b [s' , ot];
return sum of the probabilities in the final column of forward;

```

end.

Where:

$a [s , s']$ represents transition probability from current state s to next state s'

$b [s' , o_t]$ is the observation likelihood of s' given o_t

$b [s' , o_t]$ is equal 1 if the observation symbol matches the state, and is equal 0 otherwise.

The part of the forward-backward algorithm is the backward probability. This backward algorithm is almost the mirroring of the forward probability [21]. It computes the probability of the observations from $t+1$ to the end. Suppose that we are in state j at time t to given automaton A ; then:

$$\beta_i(o_t) = P(o_{t+1}, o_{t+2}, o_{t+3}, \dots, o_T | q_t = j, A)$$

The backward computation is defined as the following:

1. Initialization:
 $\beta_i(t_1) = a_{iN} \dots \dots \dots 1 < i < N$
2. Iteration:
 $\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta_j(t+1) \dots \dots \dots 1 < j < N, T > t <= 1$
3. Termination:
 $p(o|A) = a_N(T) = \beta_1(T) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) * \beta_j(1)$

Therefore, the transition probability a_{ij} and observation probability $b_i(o_t)$ will be computed from an observation sequence.

4. IMPLEMENTATION AND TESTING

The following developing model is based on components to generate pitch contour for pronunciation analysis and pronunciation adaption.

A. The SpeakCorrect Corpus Architecture

The SpeakCorrect corpus is based on annotated speech; it is will be designed to provide data for the acquisition of acoustic-phonetic knowledge and to support the development and evaluation of automatic speech recognition systems.

1) The SpeakCorrect Structure

Like the Brown Corpus, *SpeakCorrect* includes a balanced selection of dialects, speakers, and materials. It contains three dialect regions, 100 male and female speakers having a range of ages and educational backgrounds each read 150 carefully chosen words. The words were chosen to be phonetically rich and cover all the pronunciation defects of Arabic speakers (Saudi and Egypt regions). Additionally, the design walkouts equilibrium between multiple speakers saying the same word in order to permit comparison across speakers, and having a large range of words covered by the corpus to get maximal coverage of defects. One hundred of the speakers were read by each region, therefore, 100 hundred recorded utterances are stored in the corpus, each file name has internal structure, as shown in Figure 5.

Speaker ID	Gender	Word ₁	Word ₂	...	Word ₁₄₉	Word ₁₅₀
001	Female			...		
002	Male
...
150	Female

Figure (5): Structure of SpeakCorrect Structure

Each item has a phonetic transcription which can be accessed, the corresponding word tokens.

2. The SpeakCorrect Design Features

SpeakCorrect includes features of corpus design. First, such corpus contains two layers of annotation, the phonetic and orthographic levels. At this level there are different labeling schemes. A second property of *SpeakCorrect* is its balance across multiple dimensions of variation, to cover dialect regions and diphones, which facilitate later uses of corpus for purposes, when the corpus was created, such as sociolinguistics.

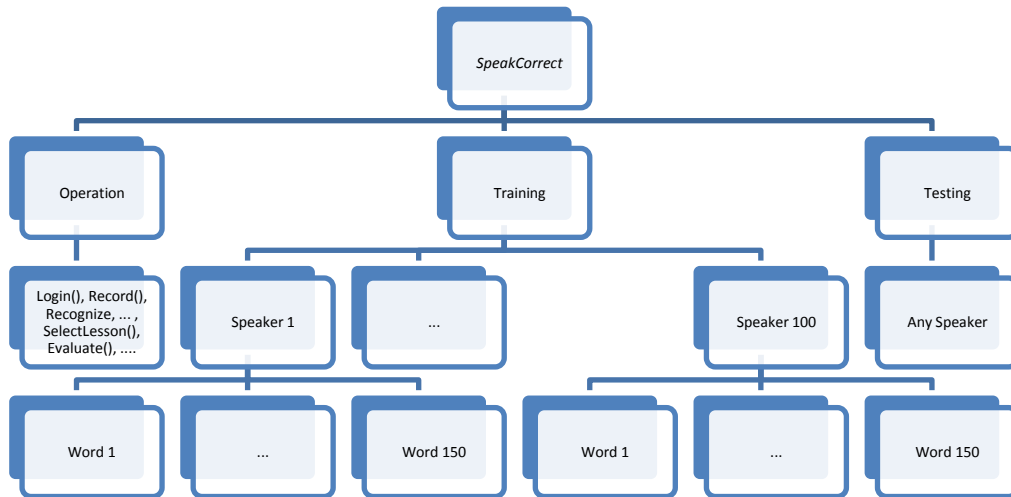


Figure (6): Structure of the Implemented Speak Correct Corpus

3. The SpeakCorrect Data Acquisition

The web is one source of rich repository of data for many natural language processing purposes. However, in our case large quantity of data samples are needed to obtain. Consequently, one of such approach is to obtain a published data from the web. The advantage of using such well-defined web data is that they are documented, stable and reproducible experimentation.

B. The SpeakCorrect Diagrams

The Visual Studio is used to draw a *component diagram* to show the structure the *SpeakCorrect* system. Therefore, UML component diagrams are created to represent the **architecture** of the *SpeakCorrect* system.

1) SpeakCorrect Use Case Diagram

The *use case diagram* is used to summarize who uses the *SpeakCorrect* system. Such diagram is used to illustrate (see figure 7):

- The scenarios in which the *SpeakCorrect* system interacts with peoples, organizations, or external systems.
- The goals that it helps those actors achieve.
- The scope of the *SpeakCorrect* system.

Figure (7): The Proposed Use Case Diagram of the *SpeakCorrect* System

2) SpeakCorrect Class Diagram

The *UML class diagram* describes data types and their relationships separately from their implementation. The diagram is used to focus on the logical aspects of the classes, instead of their implementation. There are three standard kinds of classifier available on the toolbox of the UML tools. These are referred to as *types*: classes, interfaces, and an enumeration. The Classes is used to represent data or object types for most purposes. The Interfaces in a context is employed to differentiate between pure interfaces and concrete classes that have internal implementations. This difference is useful when the purpose of the diagram is to describe a software implementation. And, the Enumeration is used to represent a type that has a limited number of literal values. Figure 8 shows the proposed class diagram of the *SpeakCorrect* that includes 4 classes and one interface.

Figure (8): The Proposed Class Diagram of the *SpeakCorrect* System

3) *SpeakCorrect* Sequence Diagram

The *sequence diagram* is drawn to display an interaction. An interaction is a sequence of messages between typical instances of classes, components, subsystems, or actors. There are two kinds of sequence diagrams: UML sequence diagrams that are part of UML modeling projects, and Code-based sequence diagrams that can be generated from .NET program code. Figures (9-a, and 9-b) illustrate sequence diagrams of the *SpeakCorrect* system.

■

Figure (9-a): Sequence Diagram of the *SpeakCorrect* System

Figure (9-b): Login Sequence Diagram of the *SpeakCorrect* System

D. The SpeakCorrect User Interface

The user interface in *SpeakCorrect*, as shown in Figure 10, is divided into three tiers. The top part contains the presentation tier; the middle part of the user interface includes the logical or business tier; which starts with registration where the login takes place login, microphone setting and adaptation, language and speech lessons and finally evaluation. The third tier is the internal one, which hosts all the properties, databases, files, etc.

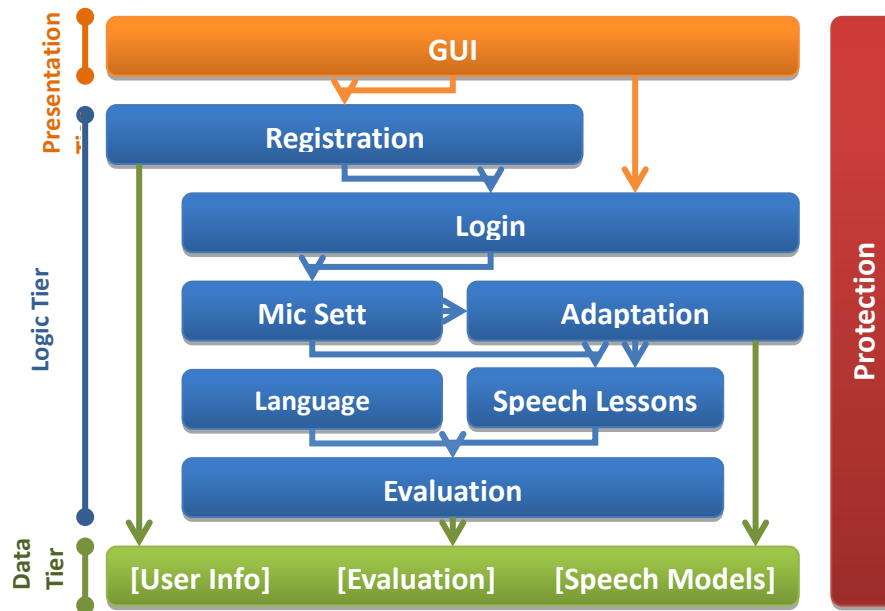


Figure (10): The Different Tiers of the *SpeakCorrect* System

E. The *SpeakCorrect* Testing

Due to previous difficulties found in Arabian accents for English pronunciation, the following testing model is based on component to evaluate and guide students for pronunciation analysis and pronunciation adaption. Figure 11 illustrates the login student information.



Figure (11): The Login Interface of the *SpeakCorrect* System

Consequently, the user interface is designed in Silverlight technology. Such user interface includes different visual properties to perform basic functions: Moving to previous and next demos playing sample (predefined example), user voice and recording user's voice. Figure 12 illustrates device setting and microphone adjustment.



Figure (12): The Device Setting and Microphone adjustment of the *SpeakCorrect* System

The pitch contour is the fluctuation in frequency associated in human voices. The development of the proposed project includes “open source .Net Code” which included the pitch contour calculation, in added to HTK code that contains mathematical algorithms. The implementation code contains collaboration module between C# code (.Net Client/Server) and the HTK component code. The second component is trying to compare the input voice against the predefined trained voices and therefore providing a mistake-if any, as shown in figure 13.

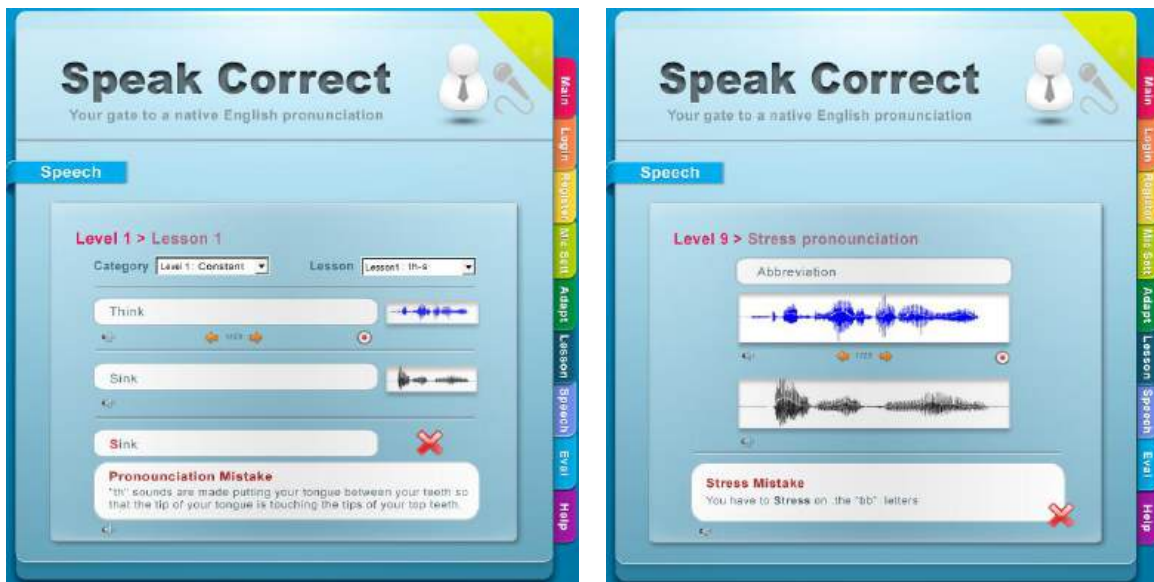


Figure (13): The Levels and Associated Lessons Testing of the *SpeakCorrect* System

Consequently, the user interface of the *SpeakCorrect* is designed in Silverlight technology. Such user interface includes tabs to perform basic functions: Moving to previous and next demos playing sample (predefined example), user voice and recording user’s voice, as shown in different previous figures. Consequently, the Model-View-ViewModel (MVVM) pattern is used to facilitate the interconnection “Click Event” for the tabs. Such visual elements properties are bounded in the underlying ViewModel class.

Below in figure 14 describes pronunciation guide with visually feedback. The feedback contains hints about things the user might try to say if the conversation has stalled. The things-tab holds a picture of all the items the user has managed to acquire. Finally, the evaluation offers lesson summarization to the students at different levels.



Figure (14): The Levels and Associated Lessons Testing of the *SpeakCorrect* System

ACKNOWLEDGEMENT

The teamwork of the *SpeakCorrect* project was funded as part of the strategic technology project (10-INF-1406-03) held at the King Abdulaziz University (KAU). Also; authors of the paper thankful to King Abdulaziz City for Science and Technology (KACST) through their grand's number 10-INF-1406-03. Their financial and support during the period this research took place is greatly acknowledged.

REFERENCES

- [1] K. Macherey, O. Bender, H. Ney, (2009). Applications of Statistical Machine Translation Approaches to Spoken Language Understanding, Audio, Speech, and Language Processing, IEEE Transactions on Volume: 17 , Issue: 4.
- [2] R. De Mori, F. Bechet. D. Hakkani-Tur, M. McTear, G. Riccardi, G. Tur, (2008). Spoken language understanding, Signal Processing Magazine, IEEE Volume: 25 , Issue: 3.
- [3] M. Dinarell, E. A. Stepanov, S. Varges, G. Riccardi, (2010). The LUNA Spoken Dialogue System: Beyond utterance classification, Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference.
- [4] N. Camelin, F. Bechet, G. Damnati, R. De Mori, (2010). Detection and Interpretation of Opinion Expressions in Spoken Surveys, Audio, Speech, and Language Processing, IEEE Transactions on Volume: 18 , Issue: 2.
- [5] K. Laskowski; E. Shriberg; (2010). Comparing the contributions of context and prosody in text-independent dialog act recognition. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference.
- [6] R. Cantrell, M. Scheutz; P. Schermerhorn, Wu Xuan, (2010). Robust spoken instruction understanding for HRI. Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference.
- [7] S. Young-In, W. Ye-Yi, J. Yun-Cheng, M. Seltzer, I. Tashev, A. Acero, (2009). Voice search of structured media data, Acoustics, Speech and Signal Processing, 2009. ICASSP, IEEE International Conference.
- [8] P. Heracleous, N. Aboutabit, D. Beautemps, (2009). HMM-based vowel and consonant automatic recognition in Cued Speech for French, Virtual Environments, Human-Computer Interfaces and Measurements Systems. VECIMS '09. IEEE International Conference.
- [9] A. Kain, J. Van Santen, (2009). Using speech transformation to increase speech intelligibility for the hearing- and speaking-impaired, Acoustics, Speech and Signal Processing. ICASSP 2009. IEEE International Conference.
- [10] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, and M. Turk, (2011). TranslatAR: A mobile augmented reality translator. Applications of Computer Vision (WACV), 2011 IEEE Workshop on 5-7 Jan. 2011.

- [11] M. Wohlmayr, M. Stark, and F. Pernkopf, (2011). A Probabilistic Interaction Model for Multi-pitch Tracking With Factorial Hidden Markov Models. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19, No. 4, May 2011.
- [12] A. Levy, S. Gannot, E. A. P. Habets, (2011). Multiple-Hypothesis Extended Particle Filter for Acoustic Source Localization in Reverberant Environments. *Audio, Speech, and Language Processing*, IEEE Transactions on Volume: 19 , Issue: 6.
- [13] G. Katsutoshi, K. Masataka, O. Kazunori, O. Tetsuya, G. Hiroshi, (2011). Simultaneous processing of sound source separation and musical instrument identification using Bayesian spectral modeling. Graduate School of Informatics, Kyoto University, Japan, (2011). *Acoustics, Speech and Signal Processing (ICASSP)*, 2011 IEEE International Conference on 22-27 May 2011.
- [14] M. Huijbregts, D. L. Mitchell, (2011). Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection. *Acoustics, Speech and Signal Processing (ICASSP)*, 2011 IEEE International Conference on 22-27 May 2011, page(s): 4436 – 4439, Prague, Czech Republic.
- [15] K. P. Sarathy, K. P. Ramakrishnan, (2008). A research bed for unit selection based text to speech synthesis, *Spoken Language Technology Workshop*, 2008. *SLT 2008*. IEEE, Page(s): 229 – 232.
- [16] Y. Han, P. Yuanyuan, W. Hong, Z. Zhengpeng, (2004). An acoustic-phonetic analysis of large vocabulary continuous Mandarin speech recognition for non-native speakers, *Chinese Spoken Language Processing*, 2004 International Symposium on 2004. Page(s): 241 – 244.
- [17] N. A. Abdul-Kadir and R. Sudirman, (2011). Vowel Effects towards Dental Arabic Consonants based on Spectrogram, *IEEE Second International Conference on Intelligent Systems, Modelling and Simulation*, IEEE Computer Society 2011.
- [18] F. Kensaku, Y. Takuto, Y. Kana, M. Mitsuji and M. Masakazu, (2011). A Double Talk Control Method Improving Estimation Speed by Adjusting Required Error Level, *Workshop on Hands-free Speech Communication and Microphone Arrays*, IEEE May 30 - June 1, 2011 .
- [19] K. Juraj and R. Gregor, (2009). *Adding Voicing Features Into Speech Recognition Based on HMM in Slovak*, *IEEE Conference, Systems, Signals and Image Processing, IWSSIP 2009*. 16th International Conference.
- [20] D. Jurafsky and J. H. Martin, University of Colorado, Boulder, (2008). *Speech and Natural Language Processing*. 2nd edition, Prentice Hall.
- [21] Z. Yan and Li Shang, (2012). Speaker Recognition Based on Principal Component Analysis and Probabilistic Neural Network, *Lecture Notes in Computer Science*, 2012, Volume 6839, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, Pages 708-715.
- [22] H. Tobias, G. Franz, and M. Wolfgang, (2012). Self-learning speaker identification for enhanced speech recognition. *Computer Speech & Language*, Volume 26, Issue 3, June 2012, Pages 210–227.
- [23] W. Silke, (2012). Automatic Error Detection in Pronunciation Training: Where we are and where we need to go. *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012 KTH, Stockholm, Sweden , (IS ADEPT, Stockholm, Sweden, June 6-8 2012)*.
- [24] P. Bryan, (2012). Rosetta Stone ReFLEX: Toward Improving English Conversational Fluency in Asia. *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012 KTH, Stockholm, Sweden, (IS ADEPT, Stockholm, Sweden, June 6-8 2012)*.
- [25] B. Smith (2011). *Arabic Speakers: Learner English*, Cambridge Handbooks for Language Teachers, 2nd Edition, Series Editor Scott Thornbury.
- [26] J. ZHU, H. WANG, E. H. HOVY, (2010). Confidence-based Stopping Criteria for Active Learning for Data Annotation. *ACM Transactions on Speech and Language Processing*, Vol. 6, No. 3, Article 3, Publication date: April 2010.
- [27] J. ZHU, H. WANG, and E. H. HOVY, (2008). Multi-Criteria-Based strategy to stop active learning for data annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics*. 1129–1136.
- [28] K. Knight and J. May, (2009). *Handbook of Weighted Automata*, Edited by Manfred Droste, Werner Kuich, Heiko Vogler, Springer. Chapter 14: Applications of Weighted Automata in Natural Language Processing.
- [29] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large Language Models in Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 858–867, Prague, June 2007.

- [30] M. Galley, M. Hopkins, K. Knight, and D. Marcu, (2004). What’s in a translation rule? In HLT-NAACL Proceedings, 2004.
- [31] D. Gildea, (2003). Loosely tree-based alignment for machine translation. In ACL Proceedings, Sapporo, Japan, 2003.
- [32] J. Graehl and K. Knight, (2004). Training tree transducers. In HLT-NAACL Proceedings, 2004.
- [33] K. Knight and J. Graehl, (2005). An Overview of Probabilistic Tree Transducers for Natural Language Processing. In CICLing Proceedings, 2005.
- [34] S. Kumar and W. Byrne, (2003). A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In HLT-NAACL Proceedings, 2003.
- [35] J. May and K. Knight, (2006). A Better n-best List: Practical determinization of weighted finite tree automata. In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pages 351–358, New York City, USA, June 2006. Association for Computational Linguistics.
- [36] J. May and K. Knight, (2006). Tiburon: A weighted tree automata toolkit. In Oscar H. Ibarra and Hsu-Chun Yen, editors, Proceedings of the 11th International Conference of Implementation and Application of Automata, CIAA 2006, volume 4094 of Lecture Notes in Computer Science, pages 102–113, Taipei, Taiwan, August 2006. Springer.
- [37] I. Dan Melamed. Multi-text grammars and synchronous parsers. In NAACL Proceedings, 2003.
- [39] B. Pang, K. Knight, and D. Marcu, (2003). Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In NAACL-HLT Proceedings, 2003.
- [40] I. A. Sag, T. Wasow, and E. M. Bender, (2003). Syntactic Theory. CSLI Publications, 2nd edition, 2003.
- [41] S. M. Shieber, (2004). Synchronous grammars as tree transducers. In TAG+ Proceedings, 2004.
- [42] S. M. Shieber, (2006). Unifying Synchronous Tree Adjoining Grammars and Tree Transducers via bi-morphisms. In EACL Proceedings, 2006.
- [43] B. Zhou, S. F. Chen, and Y. Gao, (2006). Folsom: A fast and memory efficient phrase-based approach to statistical machine translation. In Proceedings of the IEEE/ACL 2006 Workshop on Spoken Language Technology, pages 226–229, Palm Beach, Aruba, December 10–13 2006.
- [44] S. Abdou, M. Rashwan, H. Al-Barhamtoshy, K. Jambi, and W. Al-Jedaibi, (2012). *Enhancing the Confidence Measure for an Arabic Pronunciation Verification System*. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training June 6 - 8, 2012, KTH, Stockholm, Sweden.