



**A HYBRID STATISTICAL AND MORPHOLOGICAL
ARABIC LANGUAGE DIACRITIZING SYSTEM**

نظام تشكيل اللغة العربية الهجين الاحصائي والصرفي

Prepared by

Abdullah Mamoun Hattab

Supervised by

Dr. Abdulameer Khalaf Hussain

**A Thesis Submitted in Partial Fulfillment of the
Requirements for the Master Degree
In Computer Science**

**Department of Computer Science
Faculty of Information Technology
Middle East University**

Amman – Jordan

August, 2012

Middle East University

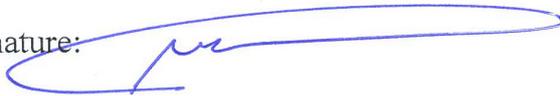
AUTHORIZATION STATEMENT

I, Abdullah Mamoun Hattab, authorize Middle East University to supply hard and electronic copies of my thesis to libraries, establishments, or bodies and institutions concerned with research and scientific studies upon request, according to the university regulations.

Name: Abdullah Mamoun Hattab

Date: 8/8 / 2012

Signature:

A handwritten signature in blue ink, consisting of a large, sweeping loop followed by a few smaller strokes, positioned to the right of the 'Signature:' label.

جامعة الشرق الأوسط

التفويض

أنا عبدالله مأمون الخطاب أفوض جامعة الشرق الأوسط بتزويد نسخ من رسالتي ورقيا
والكترونيا للمكتبات، أو المنظمات، أو الهيئات والمؤسسات المعنية بالأبحاث
والدراسات العلمية عند طلبها.

الاسم: عبدالله مأمون الخطاب

التاريخ: 2012 / 8 / 8

التوقيع:



Middle East University
Examination Committee Decision

This is to certify that the thesis titled “A Hybrid Statistical and Morpho-syntactical Arabic language Diacritizing System” was successfully defended and approved on 8-8-2012.

Examination Committee Member

1- Prof. Reyadh S. Naoum

Chairman



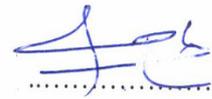
Professor

Dean of Faculty of Information Technology

Middle East University

2- Prof. Azzam Slait

Member



Proffesor

Jordan University

3- Dr. Abdilameer H. Khalaf

Supervisor



Middle East University

DEDICATION

This thesis is dedicated to my parents and my family who have supported me all the way since the beginning of my studies. Also, this thesis is dedicated to my fiancé who has been a great source of motivation and inspiration. Finally, this thesis is dedicated to all those who believe in the richness of learning.

ACKNOWLEDGMENT

All praise is due to Allah, who guided me to this.

I would like to express my sincere gratitude to my supervisor; Dr. Abdulameer Khalaf Hussain. I'm greatly indebted to his assistance, guidance and support and to my father Mr. Mamoun Hattab who inspired me and motivated me in my study.

I am very grateful to my dear parents, beloved fiancé Leyan Saleh, family and my friends whom I consider as my brothers. Thank you all for being always there when I needed you most. Thank you for believing in me and supporting me. I believe that without your support and your prayers, none of this work would have been accomplished. Finally, I hope this thesis be a useful addition to the research activities of Arabic natural language processing.

Table of Contents

A HYBRID STATISTICAL AND MORPHO-SYNTACTICAL ARABIC LANGUAGE DIACRITIZING SYSTEM	I
AUTHORIZATION STATEMENT..ERROR! BOOKMARK NOT DEFINED.	
التفويض..... ERROR! BOOKMARK NOT DEFINED.	III
EXAMINATION COMMITTEE DECISIONERROR! BOOKMARK NOT DEFINED.	
DEDICATION.....	V
ACKNOWLEDGMENT.....	VI
Expectation Maximization Algorithm.....	XI
Translation Model	XI
Random Access Memory	XI
System Model Optimization.....	XI
Application programming interface	XI
Software development kit.....	XI
CHAPTER ONE.....	1
1 INTRODUCTION	1
1.2 RESEARCH MOTIVATION	2
1.3 CONTRIBUTION	3
1.4 CHALLENGES	3
1.5 PROBLEM STATEMENT	4
1.6 METHODOLOGY	5
1.7 THESIS OUTLINE	6
CHAPTER TWO	8
2 LITERATURE SURVEY AND RELATED WORK	8
2.1 DIACRITIZATION SYSTEMS	9
2.2 STATISTICAL MODELS.....	12
CHAPTER THREE	13
3 ARABIC MORPHO-SYNTACTICAL ANALYSIS	13
3.1 INTRODUCTION.....	14
3.2 MORPHOLOGICAL MODELS	14
3.3 FUNCTIONAL APPROXIMATION	17
3.4 REUSED SOFTWARE.....	19
3.4.1 Buckwalter Arabic Morphological Analyzer.....	19

3.4.2	Functional Morphology Library	19
3.4.3	TrEd Tree Editor.....	20
3.5	ELIXIRFM ORIGINAL CONTRIBUTIONS.....	20
3.6	WRITING & READING ARABIC	21
3.6.1	Standard Notation.....	22
CHAPTER FOUR.....		25
4	STATISTICAL MACHINE TRANSLATION	25
4.1	INTRODUCTION.....	26
4.2	LANGUAGE MODEL.....	28
4.3	PHRASE-BASED TRANSLATION MODEL	29
4.4	STATISTICAL WORD ALIGNMENT.....	31
4.5	HEURISTICS FOR SYMMETRIC WORD ALIGNMENT	32
4.6	ALIGNMENT QUALITY AND TRANSLATION QUALITY	33
4.7	GIZA++.....	34
4.8	NGRAM	35
4.9	TRAINING	36
CHAPTER FIVE.....		39
5	PROPOSED MODEL AND METHODOLOGY.....	39
5.1	INTRODUCTION.....	40
5.2	STATISTICAL ARABIC DIACRITIZER MODEL.....	41
5.3	MORPHO-SYNTACTIC ARABIC DIACRITIZER MODEL	45
5.4	46
5.5	RUN TIME PHASE	46
CHAPTER SIX.....		49
6	EXPERIMENTS RESULTS.....	49
6.1	CORPUS SIZE AND SYSTEM BEHAVIOUR	50
6.2	EXPERIMENTAL SETUP	51
6.3	HYBRID SYSTEM RESULTS	52
6.4	ERRORS ANALYSIS	54
CHAPTER SEVEN.....		55
7	CONCLUSION AND FUTURE WORK.....	57
7.1	CONCLUSION.....	58
7.2	FUTURE WORK.....	58
REFERENCES.....		60

TABLE OF FIGURES

Figure 3-1 Arabic letters transliterations	24
Figure 4-1 Interlingua System	26
Figure 4-2 Translation pyramid	27
Figure 4-3 Training and data optimization	37
Figure 5-1Hybrid Arabic diacritizer system’s architecture	40
Figure 5-2 The results of running 3-gram on four words sentence.....	43
Figure 5-3 System Model Optimization	44
Figure 5-4 Morphological structure for “taskuniyna”.....	46
Figure 5-5 Phrase segmentation	47
6-1 Corpus size versus the OOV	51
6-2 output of the hybrid system	53

TABLES

Table 1-1	4
Table 2-1 Error Rate at Word and Character Levels With and Without the Diacritization of the Word-Final Letters.....	9
Table 2-2 Final results of Case Ending	10
Table 4-1phrase-based machine translation system.. Error! Bookmark not defined.	
Table 5-1Arabic words structure	46
Table 5-2 Words analyzing	47
Table 6-1 Morpho-syntactic and Statistical diacritization accuracy rate versus the hybrid diacritizer	53

ABBREVIATIONS

NLP	Natural Language Processing
AI	Artificial Intelligence
SLM	Statistical Language Model
HMMT	Hidden Markove Model Toolkit
KDATD	KACST Diacritized Arabic Text Data
KACST	King Abdulaziz City for Science and Technology
SVM	Statistical Vector Machine
POS	Part-Of-Speech
MADA	Morphological Analysis and Disambiguation for Arabic
BAMA	Buckwalter Arabic Morphological Analyzer
REGEX	Regular Expressions
GNU	General Public License
MT	Machine Translation
SMT	Statistical Machine Translation
OOV	Out-Of-Vocabulary
PB	Phrase-Based
PB-SMT	Phrase-Based Statistical Machine Translation
AER	Acoustic Event Recognition
HMD	Hidden Markove Models
EMA	Expectation Maximization Algorithm
TM	Translation Model
RAM	Random Access Memory
SMO	System Model Optimization
API	Application programming interface
SDK	Software development kit

ABSTRACT

This thesis represents a hybrid Arabic diacritizing system. The main objective of this thesis is to build a system to diacritize Arabic text automatically using statistical model and morph-syntactical model. The first part of this system determines the most likely diacritics by choosing the full-form Arabic sub-sentence diacritization with the highest weight and probability estimation. The second part of the system factorizes and tokenizes each Arabic word into its possible morpho-syntactical constituent pattern, prefix, suffix, stem and root. After factorizing, the morpho-syntactical part selects the most likely diacritization sequence from different factorizations of the word. Most of the previous works on diacritization depend on tools such as Hidden Markov Model Toolkit (HTK) and/or higher linguistic knowledge such as morphology and syntax only, while this system uses statistical machine translation algorithm and ELXIRFM morphological analyzer. The accuracy rate of this hybrid system is higher than the rates of traditional studies with larger domain of Arabic words.

الخلاصة

تمثل هذه الرسالة نظاماً هجيناً للضبط والتشكيل في اللغة العربية. الهدف من هذه الرسالة هو بناء نظام لتشكيل النصوص بشكل تلقائي باستخدام النموذج الإحصائي والنموذج النحوي والصرفي. يحدد الجزء الأول من النظام أرجح الاحتمالات للتشكيل عن طريق اختيار أقرب تقريب للشكل العام لتشكيل أجزاء الجملة في اللغة العربية. أما الجزء الثاني من النظام فيحلل و يقطع كل كلمة عربية لأقرب نمط نحوي صرفي، السوابق، اللواحق، الجذع و الجذر. بعد ذلك يختار هذا الجزء نمط التشكيل الأكثر قرباً بناءً على تحليلات مختلفة للكلمة. معظم الأعمال و الدراسات السابقة حول الضبط و التشكيل كانت فقط تعتمد على مجموعة أدوات مثل "نموذج هيدن ماركوف" و/أو قواعد معرفة لغوية اعلى دقة صرفيا ولغويًا. ، بينما يستخدم هذا النظام آلة إحصائية لترجمة الخوارزميات و محلل ELXIRFM الصرفي ونسبة الصحة لهذا النظام الهجين أعلى منها في الدراسات التقليدية مع نطاق أوسع من الكلمات العربية.

CHAPTER ONE

1 INTRODUCTION

1.1 Research Motivation

Natural Language Processing (NLP) is the processing of the spoken languages by people for communication which encompasses computer needs to understand natural language and also generate the natural language. NLP is a subfield of Artificial Intelligence and linguistic to let computer understand written, spoken languages and the process of computer analysis of input which represents human languages and then convert this input into a useful representation form (Elain Rich and Kevin, 2006).

Arabic language processing as any natural language is a system that consists of a set of symbols (alphabets and diacritics) and set of rules (grammar). The combination of these symbols conveys new information. Rules are used to govern and manage the manipulation of symbols. Language processing applied at five major levels. These levels are morphological and lexical analysis level, syntactical analysis, semantic analysis, discourse integration and pragmatic analysis level (Beesley, 1998).

The Arabic language, which is the mother tongue of more than 300 million people, presents significant challenges to many natural language processing (NLP) applications. Arabic language is complicated for natural language processing because of two main language characteristics. First characteristic is the agglutinative nature of the language and the second characteristic is the aspect of diacritics which causes problems of ambiguity at different levels. The diacritics problem does not only affect sentences' words but also the grammatical base of these words. Language modeling is used in many natural language processing applications such as speech recognition, machine translation, part-of-speech tagging, parsing and information retrieval (Farghaly and Shaalan, 2009).

1.2 Contribution

The contribution of this research can be identified as following:

- 1- Proposing hybrid system containing two layers: Statistical Machine translation and Morpho-syntactical analysis.
- 2- Improving Arabic language processing by introducing a new technique that merges between unfactorized statistical phrase-based approach and factorized linguistics approach for manipulating Arabic language to achieve the advantages of the two approaches, speed and accuracy from the statistical part and the coverage of many language's domains from the linguistic one.
- 3- Applying this hybrid technique on other problem's domain as content classification for Arabic text, this led to an impressive result.

1.3 Challenges

Due to the following challenges it is difficult to build a reliable Arabic diacritizer:

- 1- Arabic text is typically written without any diacritics (Attia, 2000), (Fatehy, N., 2005).
- 2- Arabic text is commonly written with many common spelling mistakes (ٱ-أ), (ه-هـ), (ى-ي) (Attia, M., Sept. 2005).
- 3- Due to the highly derivative and inflective nature of Arabic, it is very difficult to build a complete vocabulary that covers all (or even most of) the Arabic generable words (Attia, M., Sept. 2005).
- 4- While the virtue of morphological analyzer to solve the problem of coverage instead of using dictionary, its only drawback is that its

relatively sluggish attenuation of the disambiguation error margin with increasing the annotated training corpora which are expensive and time consuming to build and validate (Yaseen, M. et al., 2006).

About two thirds of Arabic text words have a syntactically dependent case-ending which invokes the need to a syntax analyzer which is a hard problem (Yaseen, M. et al., 2006).

1.4 Problem Statement

There are many problems related to the Arabic language processing. The absence of the diacritics from Arabic text represents a major obstacle in processing and manipulating Arabic language text leading to inaccurate results. Table 1.1 below shows the complete set of Arabic diacritics.

Table 1-1

Diacritic's type	Diacritic	Symbol
Short vowel	فتحة	َ
	ضمة	ُ
	كسرة	ِ
Doubled case ending	تنوين فتح	ً
	تنوين ضم	ٌ
	تنوين كسر	ٍ
Syllabification marks	شدة	ّ
	سكون	◌ْ

Native Arabic readers can identify the proper diacritics of the text, but when it comes to computer processing, the computer still needs to be provided with more

specialized algorithms and systems to perform the human ability to identify the proper diacritization of the text. In this context there are two aspects that should be considered in order to build Arabic Auto Diacritical system; these aspects are the Statistical Language Model and Arabic Morpho-syntactic analysis. According to these two approaches, the following problems have been identified:

1. Identifying the Arabic language morpho-syntactic aspects.
2. Many Statistical Language Model (SLM) had been built but each one has its characteristics so there is a need to categorize them.
3. Improving the probabilistic Wording Algorithms.

1.5 Methodology

There are many methodologies that can be used to solve Arabic Language Diacritics problems and these methods vary from one to another. The differences between these methodologies are based on the kind of the used language models.

The system in this thesis consists of two language models. First model is the statistical Arabic diacritizer and the second model is the morpho-syntactical Arabic diacritizer. Statistical Arabic diacritizer analyzes the undiacritized Arabic text as one sentence set and generates subsets of words from the original sentence to find the highest probability in the statistical language model and to diacritize these sub-sentences. These statistically diacritized sentences are sent to the morpho-syntactic diacritizer. Statistical language model also determines the probability of words sequences in the sentence. This hybrid system constructs a general model from translation relations and acquires special rules automatically. These rules are coarse and statistical probabilistic. Morpho-syntactic diacritizer will identify the functional morphemes to merge them into meaning-bearing stems or to remove them from

statistically probabilities. Morphemes functions belong to prefixes and suffixes. These procedures check the statistically diacritized text by applying grammatical rules from the morpho-syntactic language model.

1.6 Thesis Outline

This thesis is organized as follows:

Chapter One: An introduction about natural language processing and the main approaches used in processing Arabic language. This chapter defines the problem of diacritizing Arabic text and the challenges that Arabic language has. At the end of this chapter the used methodology and techniques will be explained shortly and discussed later in chapters 3, 4 and 5.

Chapter Two: This chapter will focus on the related works in the field of Arabic language diacritizing systems that are either based on morphological knowledge or statistical machine translation algorithms. This chapter will also discuss hybrid diacritizing systems from word forms and training algorithms that have been designed by other researches.

Chapter Three: Arabic Morpho-syntactic Analysis. This chapter describes the process of the morphological computational model for Arabic language that has been used in this hybrid system and explained linguistic concepts. These Arabic morphological processes enable the system to derive and inflect words, as well as analyze the structure of word forms and recognize their grammatical functions.

Chapter Four: Statistical Machine Translation. This chapter presents a description of the implementation of the statistical machine translation decoder and a discussion about the major design objective for the decoder, its performance relative to

other SMT decoders, and the taken steps to create a high quality, phrase-based decoder and to reduce the time.

Chapter Five: Proposed Model and Methodology, the hybrid system models and how it is built will be discussed step by step. The description starts with the data preparation, preprocessing and training ending with running the system. The process of data preparation, preprocessing and training will be fully described.

Chapter Six: Experiments Results and Conclusion, the proposed system testing and results analysis. This chapter begins with a comparison between this system's results and recent related work. Then the specifications of the training and testing corpus that are used for our experiments are discussed. After that it gives statistics found during the training phase and a detailed discussion of the experiments and the results of the hybrid system with some comparisons with the factorizing system, at the end of this chapter a complete discussion about error rate analyses of the results.

CHAPTER TWO

2 LITERATURE SURVEY AND RELATED WORK

There are many researchers who have been concentrating on the field of Arabic Language Processing and Diacritization.

2.1 Diacritization Systems

In a character based statistical model (Alghamdi, Muzaffar & Alhakami 2010) presented a system for diacritizing Arabic text which is innovative and showed high performance. They applied a new methodology that is not tied to other tool-kits such as Hidden Markov Model Toolkit. However, the accuracy rate can be improved further by adding the other possible quad-grams that are not included in KDATD. Moreover, linguistic information can be fed into the system to add morphological and syntactical rules that can enhance the accuracy rate. Error's rates are listed in table 2-1.

Table 2-1 Error Rate at Word and Character Levels With and Without the Diacritization of the Word-Final Letters (Alghamdi, Muzaffar & Alhakami 2010)

Error Rate at Word Level			Error Rate at Character Level		
Including Word Final	Excluding Word Final	Error Reduction	Including Word Final	Excluding Word Final	Error Reduction
46.83	26.03	44.42	13.83	9.25	33.12

In an approach of statistical model with machine learning technique (Shaalán, Abo Bakr & Ziedan 2008) described how they proposed a statistical approach for diacritizing case-ending of an Arabic word using SVM machine learning technique. SVM gives best results for many of NLP tasks, such as POS tagging, base NP chunking. This approach was practical and fully automated. The results were promising and can be useful in many applications that need real time

diacritization. The method was appealing as compared to hand-crafted rule based approaches. The results of evaluating the system performance showed that the technique was highly accurate with 95.3% accuracy and 82% F-measure (the percentage of recall multiplied by the percentage of precision multiplied by two all divided by the percentage of recall plus the percentage of precision). Their overall results are in table 2-2.

Table 2-2 Final results of Case Ending (Shaalán, Abo Bakr & Ziedan 2008)

Measurement	Overall Results
Accuracy	0.9953485
Precision	0.809145
Recall	0.837309
F-Measure	0.822986

Also in a statistical model, machine learning technique and Viterbi Algorithm (Alghamdi, Khursheed& Elshafei 2006), in their paper they studied the diacritization system in Arabic text and hence built a system that would be able to diacritize Arabic text automatically. The team investigated different approaches for diacritizing Arabic text and they built three systems:

- I. Automatic Diacritizer of Arabic Text Using Hidden Markov Model
- II. Automatic Diacritizer of Arabic Text Using Viterbi algorithm
- III. An Independent Diacritizer of Arabic Text.

In addition, the project was left open for further research and experiments to improve the efficiency by adding morpho-syntactic knowledge to increase its accuracy in diacritizing, which is what we are applying in this thesis.

In the same field (Elshafei, Al-Muhtaseb & Alghamdi 2006) presented a Hidden Markov Model based method to solve the problem of generating the diacritical marks of the Arabic text. The basic form of the algorithm achieved a word error rate of about 5.5%. The use of higher order grams for frequent words with multiple diacritic versions could lead to a substantial improvement in the performance. Their algorithm needed as well as preprocessing stage to synthesize diacritized forms for the unlisted words, and a post processing stage to generate the end cases.

In a little different approach (Zitouni , Sorensen & Sarikaya 2006) presented a statistical model for Arabic diacritic restoration and some linguistic categories of words. They proposed a Maximum entropy framework, which gives the system the ability to integrate different sources of knowledge. Their model had the advantage of successfully combining diverse sources of information ranging from lexical, segment-based and part of speech (POS) features. Both POS and segment-based features are generated by separate statistical systems in order to simulate real world applications. The segment-based features were extracted from a statistical morphological analysis system using WFST approach and the POS features are generated by a parsing model that also uses Maximum entropy framework.

2.2 Statistical Models

By applying n-gram (statistical method) (Meftouh, Smaili & Laskri 2008) used n-grams to modulate Arabic language several experiments had been carried out on a small corpus extracted from a daily newspaper. The sparseness data conducts us to investigate other solutions without increasing the size of the corpus. They think that even with a large corpus, segmentation is necessary. In fact, a lot of words in Arabic are constructed from patterns which are used as generative rules. Each pattern indicates not only how to construct a word but gives the syntactic role of the generated word.

Habash and Sadat (2006) described the effects of different word-level preprocessing decisions for Arabic on Statistical Machine Translation (SMT) quality. Across different schemes, English performs the best under scarce-resource condition and diacritization performs best under large-resource condition. Across techniques and under scarce-resource conditions, Morphological Analysis and Disambiguation for Arabic's (MADA) are better than Buckwalter Arabic Morphological Analyzer (BAMA) which in turn is better than REGEX. Under large resource conditions, this difference between techniques is statistically insignificant, though it's generally sustained across schemes. Further their analysis showed that combination of output from all schemes had a large potential improvement over all of the different systems, suggesting a high degree of complementarity.

CHAPTER THREE

3 ARABIC MORPHO-SYNTACTICAL ANALYSIS

3.1 Introduction

The diacritization of an Arabic word consists of two components; morphology-dependent and syntax-dependent. While the morphological diacritization distinguishes different words with the same spelling from one another; e.g. (عَلْم) which means “science” and (عَلَم) which means “flag”, the syntactic case of the word within a given sentence; i.e. its role in the parsing tree of that sentence, determines the syntax-dependent diacritic of the word. For example; (دَرَسْتُ عِلْمَ الرِّبَاضِيَّاتِ) implies the syntactic diacritic of the target word - which is an “مفعول به” in the parsing tree - is “فتحة”, while (يُفِيدُ عِلْمَ الرِّبَاضِيَّاتِ جَمِيعَ الْعُلُومِ) implies the syntactic diacritic of the target word – which is a “فاعل” in the parsing tree - is “ضمة” (Al Badrashiny 2009).

This chapter introduced the factorizing part of our presented system EKIXIR and the problems of diacritizing the morphology-dependent parts of the word and syntax-dependent ones according to its point of view are then discussed.

3.2 Morphological Models

In this section, the factorizing part used in the presented system ElixirFM is introduced and the problems of diacritizing the morphology-dependent parts of the word and syntax-dependent ones according to its point of view are then discussed.

One can observe several different streams both in the computational and the purely linguistic modeling of morphology. Some are motivated by the need to analyze word forms as to their compositional structure; others consider word inflection as being driven by the underlying system of the language and the formal requirements of its grammar. How do the current morphological analyzers of Arabic interpret, for instance, the number and gender of the masculine plurals ʾgudud (جُدُد) ‘new ones’ or qudāh (قُدَاة)

‘judges’, or the case of mustawān (مُسْتَوَى) ‘level’ ? Do they identify the values of these features that the syntax actually operates with, or is the resolution hindered by some too generic assumptions about the relation between meaning and form? What is the internal structure of the words? What lexemes or other word classes do they belong to?

There are substantial discrepancies between the grammatical descriptions of Arabic represented e.g. by (Fischer, 2001) or (Holes, 2004), and the information that the available morphological computational systems provide. One of the reasons for this is that there is never a complete consensus on what the grammatical description should be. The other source of the incompatibility lies in the observation that many implementations overlook the following general linguistic fact, restated in various contexts as the principal difference between the function and the form of a linguistic symbol:

The morpho-syntactic properties associated with an inflected word’s individual inflectional markings may underdetermine the properties associated with the word as a whole. (Stump, 2001, p. 7)

According to Stump (2001), morphological theories can be classified into two scales. One of them deals with the question of inferability of meaning, and theories divide into:

Incremental words acquire morphosyntactic properties only in connection with acquiring the inflectional exponents of those properties.

Realizational association of a set of properties with a word licenses the introduction of the exponents into the word’s morphology.

The other scale concerns the core or the process of inflection:

Lexical theories associate word's morpho-syntactic properties with affixes

Inferential theories consider inflection as a result of operations on lexemes; morpho-syntactic properties are expressed by the rules that relate the form in a given paradigm to the lexeme.

Many of the computational models of Arabic morphology, including in particular (Beesley, 2001), (Ramsay and Mansur, 2001) or (Buckwalter, 2002), are lexical in nature, i.e. they tend to treat inflectional affixes just like full-fledged lexical words. As they are not designed in connection with any syntax–morphology interface, their interpretations are destined to be incremental. That means that the only clue for discovering a word's morpho-syntactic properties is through the explicit affixes and their prototypical functions. Some signs of a lexical–realizational system can be found in (Habash, 2005).

The computational models in (Cavalli-Sforza et al., 2000) and (Habash et al., 2005) attempted the inferential realizational direction. Unfortunately, they implemented only sections of the Arabic morphological system. The Arabic resource grammar in the grammatical framework (El Dada and Ranta, 2006) is perhaps the most complete inferential–realizational implementation to date. Its style is compatible with the linguistic description in e.g. (Fischer, 2001) or (Badawi et al., 2004), but the lexicon is now very limited and some other extensions for data-oriented computational applications are still needed.

The implementation of the ElixirFM system was inspired by the methodology in (Forsberg and Ranta, 2004) and by functional programming, just like the Arabic GF is

(El Dada and Ranta, 2006). Nonetheless, ElixirFM reuses the Buckwalter lexicon (Buckwalter, 2002) and the annotations in the Prague Arabic Dependency Treebank (Hajič et al., 2004b), and implements a yet more refined linguistic model.

3.3 Functional Approximation

The theoretical model of Functional Arabic Morphology, belonging to the inferential–realizational family, compares to the style of the Buckwalter Arabic Morphological Analyzer, classified as lexical–incremental. ElixirFM converts Buckwalter’s information into the format of ElixirFm’s model. The result of this conversion is called the functional approximation.

Buckwalter Arabic Morphological Analyzer (Buckwalter, 2002, 2004a) consists of a lexicon and a Perl program implementing an original algorithm for recognizing inflected Arabic words. It is the most widely used tool in analyzing Arabic language morphologically. The coverage of the lexicon is excellent (Buckwalter, 2004b, Maamouri and Bies, 2004) and the runtime performance of the program is very reasonable. Importantly enough, the first version of the Buckwalter analyzer was published as open-source software. The analyzer consumed an ordinary Arabic text, resolved its contiguous orthographic strings, and produced morphological analyses characterizing each of them as a whole. The morphs group implicitly into the prefix, stem and suffix segments and the lemma identifies the semantically dominant morph, usually the stem, if there is one. Morphs are labeled with tags, giving them the feel that they must be morphemes, which is the source of the disagreement between incremental and realizational interpretations, as noted earlier. Let us illustrate these terms on a common example. Buckwalter’s morphology on the string `wbjAnbhA` (وجانبها) meaning ‘and close to her’ would yield with the segments now indicated explicitly.

(wabijAnibihA)	[jAnib_1]
wa/CONJ + bi/PREP +	prefix(es)
jAnib/NOUN +	stem
i/CASE_DEF_GEN + ha/POSS_PRON_3FS	suffix(es)

The underlying lexical words or the syntactic tokens, as we denote them, are however still implicit. They read wa – (وَ) ‘and’, bi – (بِ) ‘at’, jAnib+I (جَانِبِ) ‘side-of’ and Ha (هَا) ‘her’. Note the morph i, which is a mere affix and not a run-on token, unlike the other three clitics.

There is no enough functional information provided in this kind of analyses, which we claimed in chapter 3. Yet, in the past experience with Buckwalter morphology (cf. Hajič et al., 2004b, 2005), we tried to approximate the functional views as closely as possible, and developed our tokenization and tag conversion algorithms (Smrč and Pajas, 2004).

When morphs are regrouped into tokens, their original tags form sequences (central column below) which map into a vector of values of grammatical categories. The tokens of our example will receive these converted, quasi-functional, positional4 tags (left column):

C-----	wa	CONJ	(وَ)	wa-
P-----	bi	PREP	(بِ)	bi-
N----2R	jAnib+i	NOUN+CASE_DEF_GEN	(جَانِبِ)	˘g ⁻ anib-i
S--3FS2-	Ha	POSS_PRON_3FS	(هَا)	h ⁻ a

The positional notation starts with the major and minor part-of-speech and proceeds through mood and voice up to person (position six), gender, number, case, and state. The values of the categories are unset, i.e. rendered with -, either if they are

irrelevant for the particular part-of-speech and its refinement (positions one and two), or if there are no explicit data present in the original analysis, like no information on gender and number in jAnib+i. On the contrary, categories may be implied in parallel, cf. suffixed possessive pronouns being treated as regular pronouns, but in functional genitive (position nine), some values can only be set, based on other knowledge, which is the case of formal reduced definiteness, referred to also as state (position ten).

3.4 Reused Software

The ElixirFM implementations of functional Arabic morphology have developed and implemented from many open-source software projects that were used during building work. ElixirFM and its lexicons are licensed under GNU (General Public License) and are available on <http://sourceforge.net/projects/elixir-fm/>, along with other accompanying software (MorphoTrees, Encode Arabic) and the source code of this thesis (ArabTEX extensions, TreeX).

3.4.1 Buckwalter Arabic Morphological Analyzer

The bulk of lexical entries in ElixirFM is extracted from the data in the Buckwalter lexicon (Buckwalter, 2002). Habash (2004) commented on the lexicon's internal format. ElixirFM devised an algorithm in Perl using the morphophonemic patterns of ElixirFM that finds the roots and templates of the lexical items, as they are available only partially in the original, and produces the ElixirFM lexicon in customizable formats for Haskell and for Perl.

3.4.2 Functional Morphology Library

Functional Morphology (Forsberg and Ranta, 2004) is both a methodology for modeling morphology in a paradigmatic manner, and a library of purposely language

dependent but customizable modules and functions for Haskell. It is partly built on the Zen computational toolkit for Sanskrit (Huet, 2002). Functional Morphology is also related to the Grammatical Framework, cf. (El Dada and Ranta, 2006) and <http://www.cs.chalmers.se/~markus/FM/> (Humayoun, 2006).

3.4.3 TrEd Tree Editor

TrEd <http://ufal.mff.cuni.cz/~pajas/tred/> is a general-purpose graphical editor for trees and tree-like graphs written by Petr Pajas. It is implemented in Perl and designed to enable powerful customization and macro programming. ElixirFM extended TrEd with the annotation mode for MorphoTrees.

3.5 ElixirFM Original Contributions

The most important and original contributions of ELIXIRFM are:

- a- Recognition of functional versus illusory morphological categories and definition of a minimal but complete system of inflectional parameters in Arabic.
- b- Morphophonemic patterns and their significance for the simplification of the model of morphological alternations.
- c- Inflectional invariant and its consequence for the efficiency of morphological recognition in Arabic.
- d- Intuitive notation for the structural components of words.
- e- Conversion of the Buckwalter lexicon into a functional format resembling printed dictionaries.
- f- ElixirFM as a general-purpose model of morphological inflection and derivation in Arabic, implemented with high-level declarative programming.

- g- Abstraction from particular orthography affecting clarity of the model and extending its applicability to other written representations of the language.
- h- MorphoTrees as a hierarchization of the process of morphological disambiguation.
- i- Expandable morphological positional tags, restrictions on features, their inheritance.
- j- Open-source implementations of ElixirFM, Encode Arabic, MorphoTrees, and extensions for ArabTEX

3.6 Writing & Reading Arabic

The ArabTEX typesetting system (Lagally, 2004) defined its own Arabic script meta-encoding that covers both contemporary and historical orthography to an exceptional extent. The notation is human-readable and very natural to write with. Its design is inspired by the standard phonetic transcription of Arabic, which it mimics, yet some distinctions are introduced to make the conversion to the original script or the transcription unambiguous.

Unlike other transliteration concepts based on the strict one-to-one substitution of graphemes, ArabTEX interprets the input characters in context in order to get their proper meaning. Deciding the glyphs of letters (initial, medial, final and isolated) and their ligatures is not the issue of encoding, but of visualizing of the script.

ArabTEX's implementation was documented in but the parsing algorithm for the notation had not been published except in the form of the source code. The TEX code is organized into deterministic-parsing macros, yet the complexity of the whole system makes consistent modifications or extensions by other users quite difficult.

Figure 3.1 shows the letters of the Arabic orthography and their corresponding Buckwalter transliteration, Arabic TEX notation and phonetic transcription.

3.6.1 Standard Notation

Explanation will take the perspective of how sounds are represented in the writing, rather than of a calligrapher trying to encrypt the individual graphemes into notation.

The notation for consonants is listed in Figure 3.1. Short vowels are coded as a, i and u, the long ones A, I and U. We concatenate consonants and vowels in their natural order.

Long vowels produce a combination of a diacritic and a letter in the script. Doubling of a consonant is indicated with the “تشدّة” (ّ) ~ diacritic, while no vowel after a consonant results in the “سكون” (◌ْ) o. These rules interoperate, so U an I can often, even though not always, behave in the orthographic representation like uw and iy would. The consonant T and the long vowel Y can only appear as final letters, otherwise the former changes to t and the latter to A or ay. Determination of the orthographic carrier for hamza is subject to complex rules, but phonologically, there is just one apostrophe (◌ْ) consonant.

Articles, the definite article “الـ” is connected by a hyphen with the word it modifies. If assimilation is to take place, either the word’s initial consonant is doubled, or the l of the article is replaced with that consonant directly.

The indefinite article N must be distinguished by capitalization. Whether or not the orthography requires an additional silent -alif, need not be indicated explicitly. It is, however, possible to enforce a silent prolonging letter after an indefinite article (cf. Lagally, 2004). Most notably, it is used for the phonologically motivated ending aNY.

Extras, the silent -alif also appears at the end of some verbal forms. It is coded UA if representing \bar{u} , and aWA or just aW if standing for aw. The phonological auxiliary vowels that are prefixed are preserved in the notation; yet, they can be elided in speech

or turned into was. la (l̄) “{“ in the script. The auxiliary vowels that are suffixed can be marked as such by a hyphen, if one prefers so.

A word, due to the minute form of certain lexical words, the Arabic grammar has developed a convention to join them to the ones that follow or precede, thus making the whole concatenation a single orthographic word. Although by any criteria separate words, wa ‘and’, fa ‘so’, bi ‘in, by, with’ and li ‘to, for’ are written as if they were part of the word that follows them. Functionally similar words that are “heavier” monosyllables or bisyllabic, for example, -aw ‘or’, fī ‘in’, ,alā ‘on’, are not so written. (Holes, 2004, p. 92)

Lunar consonants				Solar consonants				
<i>hamza</i>	◌ْ ◌ُ ◌ِ	'	'	ء	<i>t</i>	<i>t</i>	<i>t</i>	ت
	<i>b</i>	<i>b</i>	<i>b</i>	ب	<i>ṭ</i>	<i>_t</i>	<i>v</i>	ث
	<i>ǧ</i>	<i>^g</i>	<i>j</i>	ج	<i>d</i>	<i>d</i>	<i>d</i>	د
	<i>ḥ</i>	<i>.h</i>	<i>H</i>	ح	<i>ḍ</i>	<i>_d</i>	<i>*</i>	ذ
	<i>ḥ̣</i>	<i>_h</i>	<i>x</i>	خ	<i>r</i>	<i>r</i>	<i>r</i>	ر
<i>ʿayn</i>	◌َ ◌ِ ◌ُ	'	<i>E</i>	ع	<i>z</i>	<i>z</i>	<i>z</i>	ز
	<i>ǧ̣</i>	<i>.g</i>	<i>g</i>	غ	<i>s</i>	<i>s</i>	<i>s</i>	س
	<i>f</i>	<i>f</i>	<i>f</i>	ف	<i>š</i>	<i>^s</i>	<i>š</i>	ش
	<i>q</i>	<i>q</i>	<i>q</i>	ق	<i>ṣ</i>	<i>.s</i>	<i>S</i>	ص
	<i>k</i>	<i>k</i>	<i>k</i>	ك	<i>ḍ</i>	<i>.d</i>	<i>D</i>	ض
	<i>m</i>	<i>m</i>	<i>m</i>	م	<i>ṭ</i>	<i>.t</i>	<i>T</i>	ط
	<i>h</i>	<i>h</i>	<i>h</i>	ه	<i>ẓ</i>	<i>.z</i>	<i>Z</i>	ظ
	<i>w</i>	<i>w</i>	<i>w</i>	و	<i>l</i>	<i>l</i>	<i>l</i>	ل
	<i>y</i>	<i>y</i>	<i>y</i>	ي	<i>n</i>	<i>n</i>	<i>n</i>	ن

Variants of <i>ʿalif</i>				Suffix-only letters					
<i>ʿalif</i>	(<i>ā</i>)	<i>A</i>	<i>A</i>	ا	<i>ʿalif maqṣūra</i>	(<i>ā</i>)	<i>Y</i>	<i>Y</i>	ى
<i>waṣla</i>	'	"	{	آ	<i>tā> marbūṭa</i>	(<i>t/h</i>)	<i>T</i>	<i>p</i>	ة

Variants of <i>hamza</i>				Non-Arabic consonants				
<i>madda</i>	<i>ā</i>	' <i>A</i>		آ	<i>p</i>	<i>p</i>	<i>P</i>	پ
	◌ْ	' <i>a</i>	o	أ	<i>č</i>	<i>^c</i>	<i>J</i>	چ
	◌ِ	' <i>i</i>	<i>I</i>	إ	<i>ž</i>	<i>^z</i>	<i>R</i>	ژ
	◌ُ	' <i>w</i>	<i>W</i>	ؤ	<i>v</i>	<i>v</i>	<i>V</i>	ف
	◌ِ	' <i>y</i>	}	ئ	<i>g</i>	<i>g</i>	<i>G</i>	گ

Figure 3-1 Arabic letters transliterations

CHAPTER FOUR

4 STATISTICAL MACHINE TRANSLATION

4.1 Introduction

After roughly 40 years of research in the area of machine translation (MT) the state of the art is still not well-defined (Wilks, 2008). As mentioned in (Winiwarter, 2007) the quality of Statistical Machine Translation (SMT) systems did not improve significantly over the last 10 years. While some statistical machine translation systems, for example, translate word-for-word and decide the form of the sentence afterwards (Brown et al., 1990), which works well for semantically similar languages, corpus-based machine translation systems build their transfer rules automatically from bilingual corpora (Winiwarter, 2006). Radically different languages, in particular, present a huge challenge for either approach and the hybrid system in this thesis presents a reasonable combination of those two approaches to improve the results of MT while offering several candidates for translation.

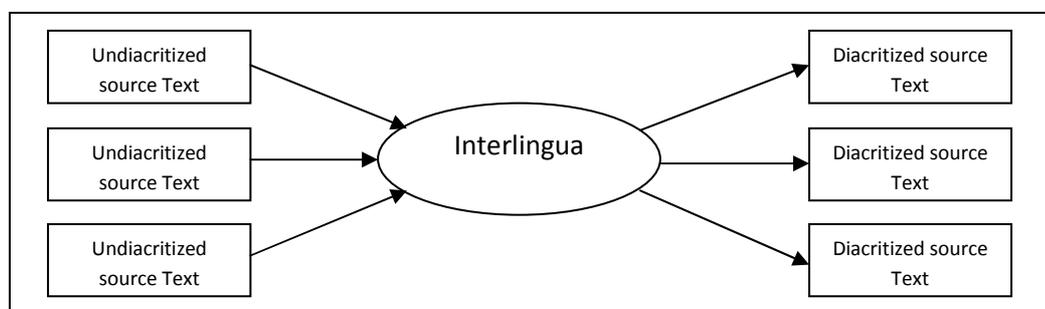


Figure 4-1 Interlingua System

Historically the first method was the Interlingua approach (Hutchins and Somers, 1992). The objective of this method is to find a language independent representation, which mediates between two or even several languages. The idea is depicted in Figure 4.1 as undiacritized text as source translated to unique Interlingua form and after it is translated into a diacritized text. This semantic driven approach is also referred to as knowledge-based translation (Leavitt et al., 1994). Figure 4.2

illustrated the path from the source text to the target text, through the intermediate language Interlingua. Undigitized text as source first analyzed to find its interlingua form, each source text has only one interlingua form that generates only one target text. Translating from source to target directly without using interlingua forms is called direct translation. The paths of other approaches of MT are located somewhere closer to the bottom, of the pyramid, which means that the path of the translation is closer to a direct translation as opposed to a detour through an intermediate step.

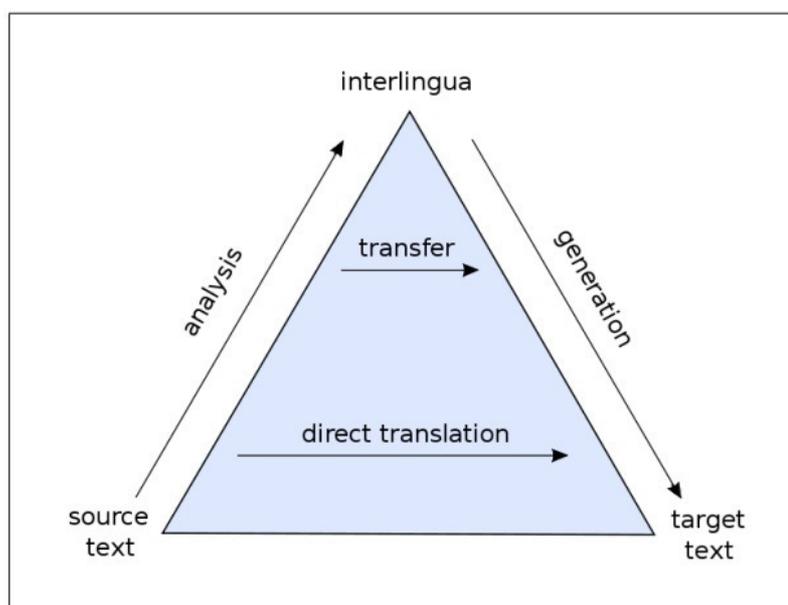


Figure 4-2 Translation pyramid

The training procedures consist of aligning each sentence pair and assigning probability scores to each match, depending on the frequency of those matches. Additionally a chance of reordering is preserved, which means that a certain probability is given to phrases which occur in a different order, than in the training set.

Currently the most popular SMT system is Moses (Hoang et al., 2007). This open-source tool offers setups for translation between various European languages. Currently, the only Asian language supported in the default configuration is Chinese. Some of the

Moses problems are faced when translating between languages with different characteristics, the words order differed from source language and target language. The word reordering, which works quite well for structurally similar languages, becomes computationally difficult with languages that have completely different word order. Even with the highest possible reordering parameters, many of the translations are not good enough to convey the meaning of the source sentence, while in this thesis, translation from Arabic undiacritized text to a well diacritized one demands same words order to guarantee accuracy. Another problem is out of vocabulary (OOV) that the corpus is the only source of knowledge for the translation. It means that the word would not be translated if a word is encountered that does not occurred in the corpus.

There are two main characteristics that any good translation should possess: a translation should be true to the meaning of the original, and a translation should be formulated in fluent natural language. These two factors are often referred to as faithfulness and fluency. Both factors must be balanced to produce an optimal translation.

4.2 Language Model

The Language Model is not just something we see in statistical machine translation, but also in many other fields of Natural Language Processing as a whole, such as in speech recognition, and spelling checkers (Maarten van Gompel, 2009). The Phrase-based Translation method proposed in this thesis will also make extensive use of a language model; therefore it is worth providing the necessary theoretical background in certain detail. The purpose of a language model is to predict the likelihood of a particular sentence. However, one of the useful implications of a language model is that

malformed sentences are in a certain sense less likely to be produced in a language than a well-formed sentence.

Given a word string, s_1, s_2, \dots, s_n , we can, without loss of generality, write:

$$\Pr(s_1, s_2 \dots s_n) = \Pr(s_1) \Pr(s_2 | s_1) \dots \Pr(s_n | s_1 s_2 \dots s_{n-1}).$$

Thus, we can recast the language modelling problem as one of computing the probability of a single word given all of the words that precede it in a sentence. At any point in the sentence, we must know the probability of an object word, s_j , given a history, $s_1, s_2 \dots s_{j-1}$. Because there are so many histories, we cannot simply treat each of these probabilities as a separate parameter. One way to reduce the number of parameters is to place each of the histories into an clustering class in some way and then to allow the probability of an object word to depend on the history only through the equivalence class into which that history falls. In an n-gram model, two histories are equivalent if they agree in their final n-1 words. Thus, in a bigram model, two histories are equivalent if they end in the same word and at a trigram model, two histories are equivalent if they end at the same two words (Peter F. Brown et al. 1990).

4.3 Phrase-based Translation Model

The basic idea of phrase-based translation is to segment the given source sentence into phrases, then translate each phrase and finally compose the target sentence from these phrase translations (Zens, Och and Ney, 2004) Given a sentence pair and a corresponding word alignment, phrases are extracted following the criterion in (Och and Ney, 2004) and the modification in phrase length in (Costa-jussa and Fonollosa, 2005). A phrase (or bilingual phrase) is any pair of many source words and more than one target words that satisfies two basic constraints:

1. Words are consecutive along both sides of the bilingual phrase,
2. Word on either side of the phrase is aligned to a word out of the phrase.

It is infeasible to build a dictionary with all the phrases (recent papers showed related work to tackle this problem, (Costa-jussa and Fonollosa, 2005). That is why we limit the maximum size of any given phrase. Also, the huge increase in computational and storage cost of including longer phrases does not provide a significant improve in quality (Koehn, Och, and Marcu, 2003) as the probability of re-appearance of larger phrases decreases.

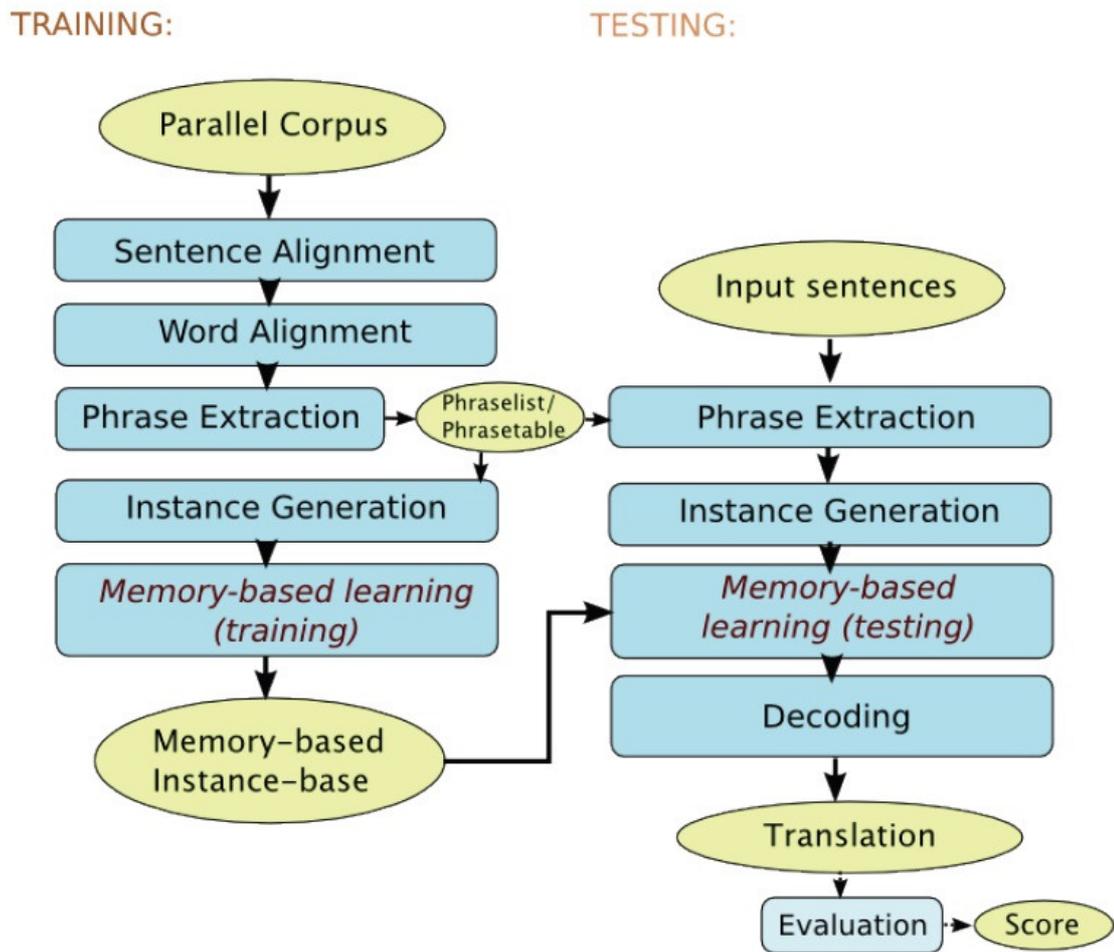
The hybrid system in this thesis has considered two length limits. First, the system extracts all the phrases of length X or less (usually X equal to 3). Then, the system adds phrases up to length Y (Y greater than X) if they cannot be generated by smaller phrases. Basically, the system selects additional phrases with source words that otherwise would be missed because of cross or long alignments (Costa-jussa and Fonollosa, 2005).

Given the collected phrase pairs, we estimate the phrase translation probability distribution by relative frequency.

$$p(f|e) = \frac{N(f,e)}{N(e)} \text{ (Costa-jussa and Fonollosa, 2005)}$$

Where $N(f,e)$ means the number of times the phrase f is translated by e . If a phrase e has $N>1$ possible translations, then each one contributes as $1/N$ (Zens, Och and Ney, 2004). The whole setup of the phrase-based machine translation system that proposed in this thesis, is illustrated in figure 4.1

Figure 4-3 phrase-based machine translation system



4.4 Statistical Word Alignment

There are several classes of methods for creating lexical correspondences giving a bilingual sentence pair. The first class of research attempts to construct directly the word alignment on bilingual sentence pairs with notable methods such as the IBM Models (Brown et al., 1993). Another class of approaches generates word alignment in the process of aligning structures or tree representations of the bilingual sentences (Ding et al., 2003). The objective of this approach is to produce an alignment between constituents (or sentence substructures), and word alignment which can be viewed as a byproduct of this process. A third group of research is bilingual parsing (Wu, 1997; Alshawi et al., 2000). This approach is regarded as an intermediate form of the above-

mentioned two classes in that it permits the probabilistic trade-off between lexical correspondences and the amount of information present in the monolingual parses.

4.5 Heuristics for Symmetric Word Alignment

Given that the most widely used word alignment models, namely the generative models, are mostly asymmetric, i.e. these models assume that each target word can be aligned to exactly one source word; these models can produce 1-to-1 and 1-to-n links, but not n-to-1 links. Och and Ney (2003) were the first who introduced heuristics for symmetric word alignment by heuristically selecting links from the union of links produced by source-to-target and target-to-source word alignment. A set of heuristics were presented including union, intersection and refined methods, of which refined methods systematically produce better SMT results in their experiments. Given source-to-target alignment $A_{f \rightarrow e}$ and target-to-source alignment $A_{e \rightarrow f}$, the alignment intersection $A \cap$ and union $A \cup$ are defined as follows:

Intersection: $A \cap = A_{f \rightarrow e} \cap A_{e \rightarrow f}$,

Union: $A \cup = A_{f \rightarrow e} \cup A_{e \rightarrow f}$,

Given this definition, intersection links are a subset of union links $A \cap \subseteq A \cup$. Alignment intersection normally has a higher precision and union yields a higher recall. However, neither of them is most suitable for PB-SMT systems. Intersection contains too few links and results in a large number of phrase pairs in the phrase extraction stage because the phrases that are consistent with word alignment increase substantially when a large number of unaligned words exist, which causes the phrase extraction to be not properly constrained. Union normally contains a large number of incorrect links which can prohibit the extraction of useful phrases. (Och and Ney, 2003).

Some other methods for symmetrisation had also been proposed. Matusov et al. (2004) proposed an algorithm which considers the alignment problem as a task of finding the edge cover with minimal costs in a bipartite graph, where the parameters of IBM Models and first-order HMM word-to-word alignment models are used to determine the costs of aligning a specific target word to a source word. Fraser and Marcu (2007a) presented a new generative model allowing the production of m-to-n links; however, this model substantially increases the complexity of the alignment process.

4.6 Alignment Quality and Translation Quality

The intrinsic alignment quality is normally measured against a manually annotated word alignment data. In the context of MT, the impact of word alignment on the final translation quality is considered an important objective. However, the correlation between intrinsic word alignment quality (e.g. precision, recall and F-score) and extrinsic translation quality of PB-SMT systems is quite complicated. Despite current intensive investigations into the impact of word alignment quality on SMT, no conclusive agreement can be reached, given that different studies used different data and systems. However, there is a widespread recognition within the community that an improvement in intrinsic word alignment quality does not necessarily imply an improvement in translation quality (Liang et al., 2006; Ma et al., 2008a), and vice-versa (Vilar et al., 2006). Fraser and Marcu (2007b) and Ma et al. (2009a) also showed that the correlation is weak when the intrinsic quality is measured with F-score. Besides general measures like F-score and AER, various studies have investigated the effect of balancing precision and recall on MT performance. While Ayan and Dorr (2006) and Chen and Federico (2006) observed that higher precision alignments are more useful in

a PB-SMT system, Mariño et al. (2006) observed that a high recall alignment improved the performance of an N-gram-based SMT system.

Fraser and Marcu (2007b) compared the performance of PB-SMT using the word alignment obtained via the intersection, union and refined symmetrisation of IBM Model 4 source-to-target and target-to-source alignments. The word aligner was trained with different amounts of data so that the quality of word alignment varied. Their results on large corpora did not confirm the hypothesis that higher precision alignments are more beneficial to PB-SMT systems than higher recall alignments. From their experiments, increasing the alignment precision (for example, by taking the intersection of source-to-target and target-to-source alignments) improves PBSMT systems only when the training data set is small. With larger corpora, higher recall alignments (like union or refined methods) are better.

4.7 GIZA++

GIZA++ (Och and Ney, 2003) is a program that trains the IBM Models (Brown et al., 1993) as well as a Hidden Markov Model (HMM) (Vogel et al., 1996), and uses these models to compute Viterbi alignments for statistical machine translation. While GIZA++ can be used on its own, it typically serves as the starting point for other machine translation systems, both phrase-based and syntactic. For instance, running GIZA++ is the first step in training the popular phrase-based translation system Moses (Koehn et al., 2007). The hierarchical phrase-based translation system Hiero (Chiang, 2005) also uses GIZA++ to generate word alignments. Galley et al. (2004) use word alignments from GIZA++ to learn rules for syntax-based machine translation. Both the IBM Models and the Hidden Markov Model are trained using the EM algorithm. Because EM chooses parameters which maximize the likelihood of the data, it tends to

over fit the parameters to the data. Johnson (2007) showed that Hidden Markov Models for part-of-speech tagging perform better when the number of hidden states is restricted; when more hidden states are allowed, the forward-backward algorithm, a version of the EM algorithm (Dempster et al., 1977) which trains HMMs, will over fit the model to the data. Johnson experimented with Bayesian techniques, which use a prior on the parameters to discourage them from taking on unreasonable values, and found that using variation Bayes decreased the amount of over fitting that occurred when more hidden states were used. Bayesian techniques in general and variation Bayes in particular have been used to control over fitting in a number of natural language processing applications. In machine translation, Blunsom et al. (2008) and DeNero et al. (2008) use Bayesian techniques to learn bilingual phrase pairs. In this setting, which involves finding a segmentation of the input sentences into phrasal units, it is particularly important to control the tendency of EM to choose longer phrases, which explain the training data well but are unlikely to generalize. This report, in contrast, is concerned with word-level translation models. This is because word-level alignments are widely used as the first step in most current translation systems.

4.8 Ngram

The N-gram approach to SMT is considered to be an alternative to the phrase-based translation, where a given source word sequence is decomposed into monolingual phrases that are then translated one by one (Marcu and Wong, 2002). The N-gram-based approach regards translation as a stochastic process that maximizes the joint probability $p(f, e)$, leading to a decomposition based on bilingual n-grams. The core part of the system constructed in this way is a translation model (TM), which is based on bilingual units, called tuples, that are extracted from a word alignment (performed with GIZA++ tool 4) according to certain constraints. A bilingual TM actually constitutes an n-gram LM of tuples, which approximates the joint probability between the languages under consideration and can be seen here as a LM, where the language is composed of tuples.

4.9 Training

Statistical machine translation relies heavily on the available training data. Typically, the more data is used to estimate the parameters of the translation model, the better it can approximate the true translation probabilities, which will obviously lead to a higher translation performance. However, large corpora are not easily available. The collected corpora are usually from very different areas. Larger amount of training data requires larger computational resources. With the increasing of training data, the improvement of translation quality will become smaller and smaller. Therefore, while keeping collecting more and more parallel corpora, it is also important to seek effective ways of making better use of available parallel training data. Training and training optimization model is presented in figure 4.3

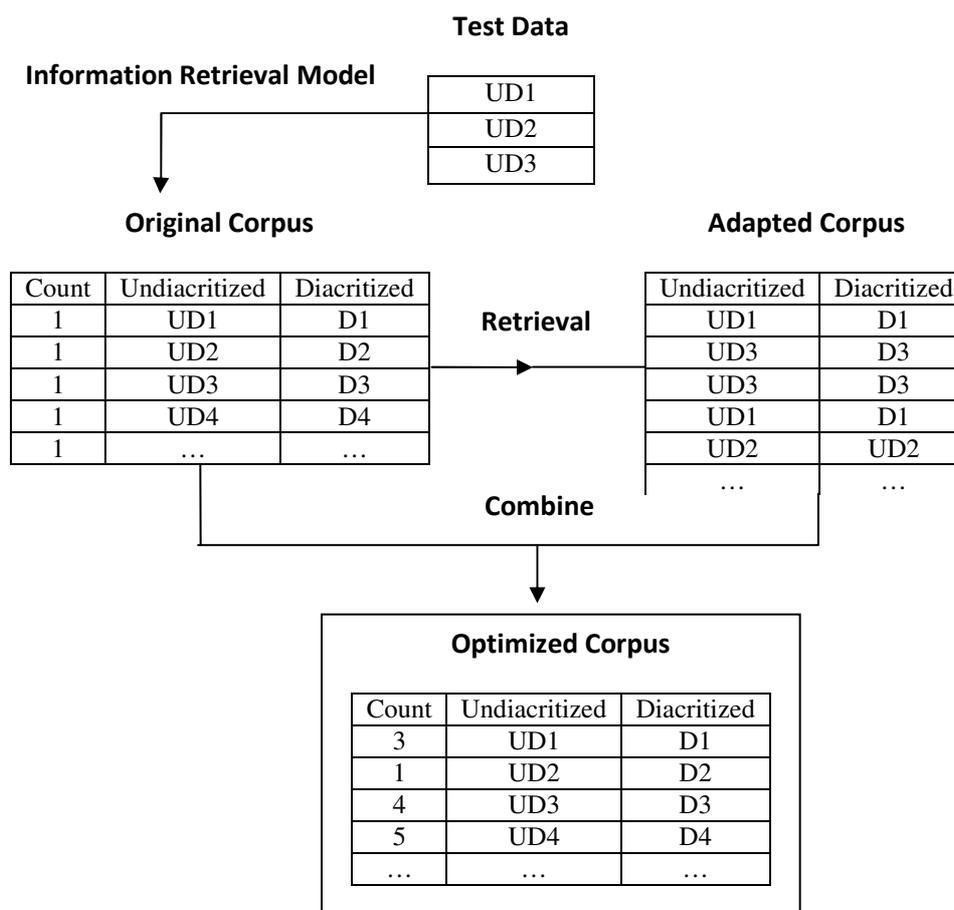


Figure 4-4 Training and data optimization

There are two cases when we train a SMT system. In the first case, when we know the target test set or target test domain, for example, when building a specific domain SMT system or when participating the NIST MT evaluation. In the second case, when we are unaware of any information of the testing data. There are two main methods to exploit full potential of the available parallel corpora in the two cases. For the first case, by optimizing the training data offline to make it match the test data better in domain, topic and style, thus improving the translation performance. For the second case, by first dividing the training data into several domains and training sub models for each domain. Then, in the translation process, we try to optimize the predefined models according to the online input source sentence. Information retrieval model is used for

similar sentences retrieval in both methods. Experiments show that both methods can improve SMT performance without using any additional data.

CHAPTER FIVE

5 PROPOSED MODEL AND METHODOLOGY

5.1 Introduction

This thesis develops a hybrid system that combines the statistical machine translation based diacritizer with another diacritizer that is based on morpho-syntactical knowledge based. Each of these approaches has its own advantages and disadvantages. This hybrid system will take the advantages of both approaches to optimize the accuracy of the Arabic diacritizer and to remove a large extent of ambiguities in order to enhance the performance of the diacritizer of Arabic text. Figure 5-1 shows the architecture of this hybrid Arabic diacritizer system.

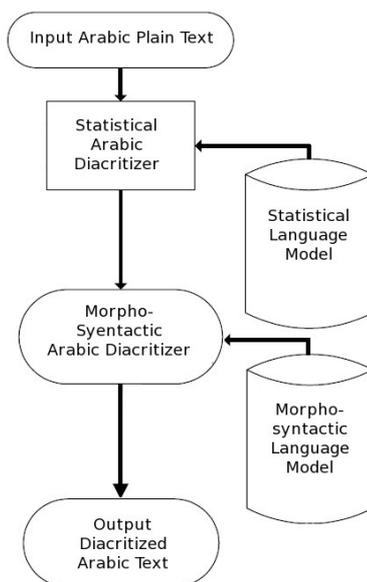


Figure 5-1 Hybrid Arabic diacritizer system's architecture

This hybrid system consists of two models. First model is the statistical Arabic diacritizer and the second model is the morpho-syntactical Arabic diacritizer. Statistical Arabic diacritizer analyzes the undiacritized Arabic text as one sentence set and generates subsets of words from the original sentence to find the highest probability in the statistical language model and to diacritize these sub sentences. These statistically diacritized sentences are sent to the morpho-syntactic diacritizer. Statistical language

model also determines the probability of words sequence in the sentence. The hybrid system constructs a general model from translation relations and acquires special rules automatically. These rules are coarse and statistically probabilistic. Morpho-syntactic diacritizer identifies the functional morphemes to merge them into meaning-bearing stems or to remove them from statistical probabilities. Morphemes functions belong to prefixes and suffixes. These procedures check the statistically diacritized text by applying grammatical rules from the morpho-syntactic language model.

5.2 STATISTICAL ARABIC DIACRITIZER MODEL

The first model of the hybrid system is based on statistical machine translation which heavily relies on the available training data. Typically, the more data is used to estimate the parameters of the diacritization model, the better it can approximate true diacritization probabilities, which will obviously lead to a higher translation performance. The statistical language model consists of three major steps. The first step is to create a list of frequently used Arabic well-diacritized sentences. The second step is concerned with creating a non- diacritized copy to build the training model. In the third step, the list created in step one will be used to diacritize Arabic text. Also, the statistical language model applies training process on a diacritized Arabic text corpus (books, articles and newspapers) that was developed manually by Arabic language experts. The Arabic corpus in the hybrid system consists of 98 text files with an average of 82223 diacritized words in each file that are treated as white space-delimited tokens for building training process. This huge number of text files provides an acceptable domain of training corpus which overcomes the common problem of Out-Of-Vocabulary in traditional studies. Most statistical diacritic systems use transfer rules and a rich translation lexicon, while this system focuses on machine translation as knowledge based systems that apply Interlingua representation as an intermediate step

between input and output. The statistical language model of the system develops a Baye's rule () to reformulate the translation probability for translating a non-diacritic sentence (ND) into diacritized sentence (D) as explained below:

$$\mathbf{argmaxep}(D|ND) = \mathbf{argmaxe} \mathbf{p}(ND|D) \mathbf{p}(D),$$

Where the above structure generates and abstracts the probability of D given ND and this structure can be found by assuming that ND has occurred and applied under that assumption depending on the probability of D occurrence. This concept can be explained in the following example.

$$\mathbf{P}(ND) = \mathbf{P}(\text{درس الطالب مادة الامتحان})$$

$$\mathbf{P}(D) = \mathbf{P}(\text{الامتحان, الامتحان, مَادَةٌ, مَادَةٌ, الطالب, الطالب, دَرَسَ, دَرَسَ, دَرَسَ})$$

$$\mathbf{Argmaxep}(\text{دَرَسَ الطالبُ مَدَّةَ الامْتِحَانِ | درس الطالب مادة الامتحان})$$

The steps of building the statistical language model are:

Step 1 N-gram model: An original motivation for developing n-gram package was to work in word sense disambiguation. The goal of a language model is to determine the probability of a word sequence, conditioning the probability of a word on the identity of the last n-1 words. The n-gram package used in this hybrid system is built using C++. The n-gram uses Ternary search tree instead of hashing table for faster n-gram frequency counting.

Step 2 Alignments: Initially the system sets a pair of strings which are translations of one string form to another form, by enclosing the string in parentheses and separating them using a vertical bar. Thus it writes the translation (ND|D); i.e. “drs AltAlb mAdp

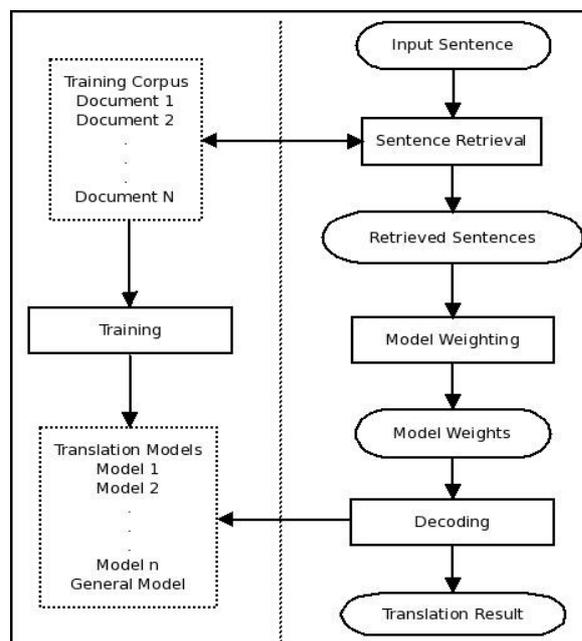
AlAmtHan”, after running applying 3-gram on the previous sentence it will generate combinations of 1-gram, 2-gram and 3-gram as shown in table 5.2.

1-gram	2-gram	3-gram
دَرَسَ	دَرَسَ الطَّالِبُ	دَرَسَ الطَّالِبُ مَادَّةَ
الطَّالِبُ	الطَّالِبُ مَادَّةَ	الطَّالِبُ مَادَّةَ الامْتِحَانِ
مَادَّةَ	مَادَّةَ الامْتِحَانِ	
الامْتِحَانِ		

**Figure 5-2 The results of running 3-gram on four words sentence
Step 3 Data preparation**

The parallel corpus must be converted into a format that is compatible with the GIZA++ toolkit. GIZA++ is a freely available implementation of the IBM Models. Two vocabulary files are generated and the parallel corpus is converted into a digitized format. The vocabulary files contain words, integer word identifiers and word count information. A sentence pair now consists of three lines. The first line is the frequency of the sentence; this number can be used for weighting different parts of the training corpus differently. Next two lines contain word ids of the ND and the D sentence. GIZA++ also divides the sentence's into words word classes. This division is done automatically by running mkcls tool to train word classes by using a maximum-likelihood-criterion. The resulting word classes are especially suited for language and statistical translation models. Word classes are only used for the IBM reordering model in GIZA++.

Step 4 Running GIZA++: In order to run GIZA++, the system establishes word alignments. The word alignments are taken from the intersection of bidirectional runs of GIZA++ plus some additional alignment points from the union of the two previous steps. Running GIZA++ is the most time consuming step in the training process and requires a much memory space (4-8 GB RAM which is common space for large parallel corpora). After some statistical information and the ND sentence, the D sentence is listed word by word with references to aligned ND words. Each ND word may be aligned to multiple D words, but each D word is only aligned to at most one ND word. This one-to-many restriction is reversed in the inverse GIZA++ training run. The training and translation procedure of the system model optimization is illustrated in Figure 5-3.



5-3 System Model Optimization

The system was developed to diacritize undiacritized Arabic text using the knowledge base system. The input to the system is the sequences of words, sentences. The system assumes each sentence as a sequence of undiacritized words. The objective

is to diacritize the given sequences of undiacritized words by decoding it from the trained data and from language model.

5.3 MORPHO-SYNTACTIC ARABIC DIACRITIZER MODEL

The second model of the hybrid system is morpho-syntactical Arabic diacritizer. This model uses a mature functional Arabic morphology analyzer called ElixirFM to develop a computational model of the morph-syntactical analysis. Using ELIXIRFM, the system in this thesis will be able to derive, inflect and analyze the structure of word forms and recognize their grammatical functions using deferent morphological theories. Morphological theories can be classified into two scales. The first scale deals with the question of inferring of meaning. The second scale is concerned with the core or the process of inflection using lexical theories to associate word's morpho-syntactic properties with affixes, and inferential theories to consider inflection as a result of operations on lexemes. The second scale of morphological theories is used in the system to analyze sentences and words. Arabic morpho-syntactical model assumes the canonical structure uniquely to represent any given Arabic word as, w , to be a quadruple of lexemes (or morphemes) so that $w \rightarrow q = (p, r, f, s)$ where, p is prefix, r is root, f is pattern and s is suffix code. Figure 5-5 illustrates the morpho-syntactical analyzer analysis for word "taskuniyna", and table 4 shows the results of applying the morpho-syntactical analyzer on another sentence which is "درس الطالب الامتحان". The total number of lexemes of all these categories in this model is around 7,800. With such a limited set of lexemes, the dynamic coverage exceeds 99.8% measured on large Arabic text corpora excluding transliterated words.

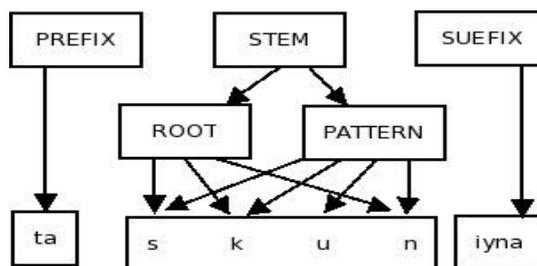


Figure 5-4 Morphological structure for “taskuniyna”.

Table 5-1 Arabic words structure

Word	Prefix	Root	pattern	Suffix
درس	-	د ر س	فَعَلَ	-
الطالب	الـ	ط ل ب	فَاعِل	-
مادة	-	م د د	فَاعِلَة	ة
الامتحان	الـ	م ح ن	اِفْتَعَلَ	-

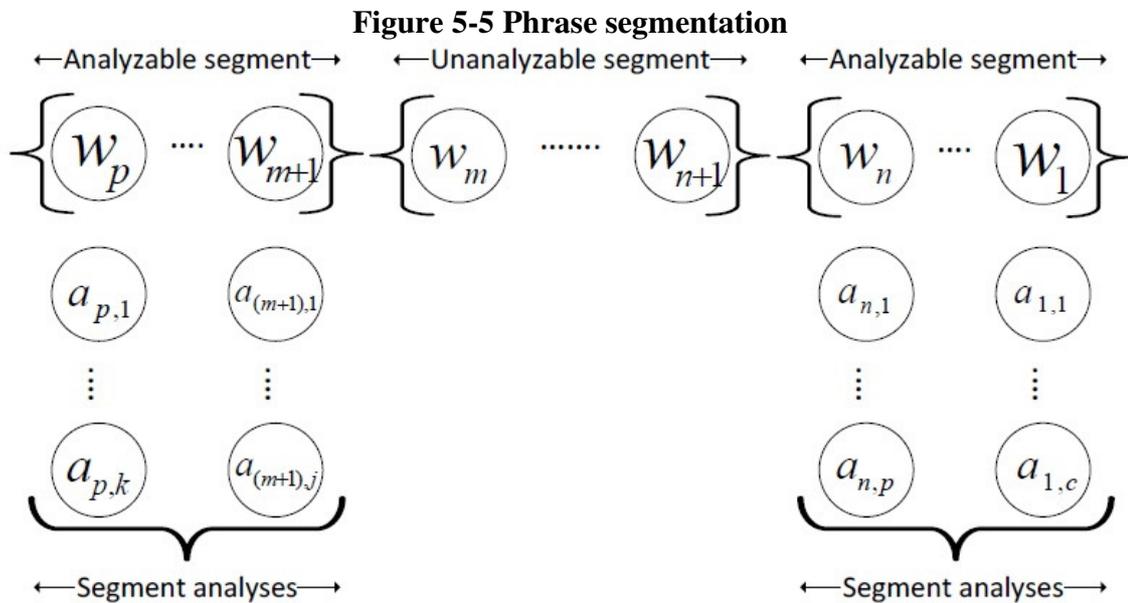
5.4

5.5 Run Time Phase

In this phase, the input text is passed to the hybrid system to search for each word of the input text in the dictionary at the statistical language model. If the word is found, then it will get all its probabilities starting from highest weight, then all these statistical results are sent to the morpho-syntactical analyzer to release ambiguity. Word is called “analyzable” and all its diacritization occurrences are retrieved from the dictionary. A consequent series of analyzable words in the input text is called “analyzable segment” and the remaining words are called “un-analyzable segment”. The next example illustrates this idea:

“يطلق الناس اسم تشامبيانز ليجز على دوري رابطة الابطال لكرة القدم”

All the diacritization occurrences of the words in an analyzable segment constitute a lattice, as shown by figure 5-5 below, that is disambiguated via n-grams probability estimation and A* lattice search to infer the most likely sequence of diacritization.



The diacritized full-form words of the disambiguated analyzable segments are concatenated to the input words in the un-analyzable segments to form a less ambiguous sequence of Arabic text words. The latter sequence is then handled by the Morpho-syntactical model that is illustrated in chapter 3.

Then by passing this text to the system; the output will be as follows: (note: “A” means Analyzable Segment and “U” means Un-Analyzable Segment) as illustrated in table 5-2

Table 5-2 Words analyzing

A	A	A	A	A	A	U	U	A	A	A
القدم	لكرة	الابطال	رابطة	دوري	على	ليجز	تشامبيانز	اسم	الناس	يطلق
القدم	لكرة	الابطال	رابطة	دوري	على	ليجز	تشامبيانز	اسم	الناس	يطلق
القدم	لكرة	الإبطال	رابطة	دوري				اسم	الناس	يطلق
.	.	.	رَبطة	.				.	الناس	.
.

.	
---	--	---	--	---	--	--	--	---	--	---

By applying that on the above example the output text becomes as follows:

‘يُطَلَقُ النَّاسُ اسْمَ تَشَامِييَانِزَ لِيَجْزَ عَلَى دَوْرِي رَابِطَةَ الْأُبْطَالِ لِكُرَةِ الْقَدَمِ’

CHAPTER SIX

6 EXPERIMENTS RESULTS

This chapter shows a fully comprehensive study for the comparison between the statistical, morpho-syntactical diacritizers and the hybrid diacritizing system.

6.1 Corpus Size and System behaviour

- I. The increase of training corpus leads to increase of the unique words list in the dictionary for the statistical language model. While the growing of only one domain made the system's data biased to that domain, so the training corpus from all domains must be balanced in order to reach more accuracy in result's kind. Increasing corpus size for a domain doesn't solve OOV problem but at certain size of data with balanced domains OOV get stable measures as shown in figure 6-1.
- II. The effect of the training corpus size increases the possible analyses for words (e.g. the word is "علم" and the analyses are "عَلِمَ - عَلِمَ - عَلِمَ ...etc"), it is found that by increasing the corpus size the possible analyses per words increase too.
- III. Morphological rules for prefixes and suffixes sometimes take off more characteristics from original word which blurs the meaning of that word (e.g. the word is "العلميون" and after stemming rules "علمي - علمي - علميون - العلم - العلمي... etc") each word from the analyzed stems may have more than one pattern for diacritizing. In this system all results are retrieved for testing and analyzing.

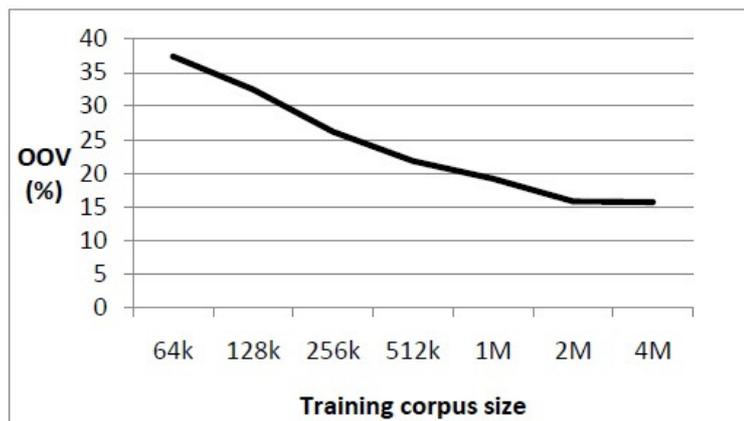


Figure 6-1 Corpus size versus the OOV

6.2 Experimental Setup

The annotated corpus data used to test the proposed Arabic diacritizer consist of the following packages:

I- A standard Arabic text corpus with size of 50,000 words collected randomly.

This package is called TST_CORP_1. This text corpus is morphologically analyzed, manually revised and validated from training corpus data domain.

II- A standard Arabic text corpus with size of 50,000 words that are out of the training corpus with complex morphological and syntactical structures. This corpus is extracted mainly from old classical Arabic literature and Islamic domains. This package is called TST_CORP_2.

III- The test data consists of 10,000 words that cover diverse domains out of training corpus, TST_CORP_1 and TST_CORP_2. This test package is called TST_CORP_3.

The error counting conventions in the following experiments are as follows:

1. The word is counted as “OOV error” if the word is not diacritized by the statistical diacritizer wrong.
2. The word is counted as “Syntax error” if the diacritics (including case-ending) of the word is wrong.
3. The word counted as “Morphological error” if the word is not analyzed or wrongly analyzed by the morpho-syntactical diacritizer.

Words might be counted in one error’s category or more, i.e. if word is OOV and not analyzable it counts for as both errors.

The experiments have been conducted to evaluate the performance of Arabic text diacritization via both the two architectures presented in this thesis.

6.3 Hybrid System Results

The above experimental packages are used to evaluate the performance of the hybrid Arabic diacritizer via the two models presented in this thesis separately to compare the diacritization accuracy of the statistical language model and the morpho-syntactical model. The change in diacritization accuracy of both models with the gradual increase of training corpus size is sensed. All these measure are registered in table 6-1.

These results show that the hybrid Arabic diacritizer accuracy is higher than the morpho-syntactical and statistical language diacritizer. The difference between the morpho-syntactical diacritizer error rates is clearly wide, while the difference between the statistical diacritizer error rates is much closer and is vanishing with the increase of training data.

Table 6-1 Morpho-syntactic and Statistical diacritization accuracy rate versus the hybrid diacritizer

Corpus data	Morpho-syntactical accuracy	Statistical Language model accuracy	Hybrid Arabic diacritizer accuracy
TST_CORP_1	94.5%	99.8%	99.4
TST_CORP_2	91.2%	97%	94.8%
TST_CORP_3	86.2%	81.7%	85.9%

The initial evaluation of the system is encouraging. However, the accuracy rate can be improved further by adding other possible quad-grams that are not included in the n-gram model. Moreover, linguistic information can be fed into the system to add morphological and morpho-syntactical rules that can enhance the accuracy rate. Figure 3 illustrates the output of the hybrid system.

```

echo " درس الطالب مادة الامتحان " | tools/translater/dist/bin/translate -f work/model/config.ini
Loading lexical distortion models...
have 1 models
Creating lexical reordering...
weights: 0.300 0.300 0.300 0.300 0.300 0.300
Loading table into memory...done.
Created lexical orientation reordering
Start loading LanguageModel /home/abdullah/demo/work/lm/news-commentary.lm : [0.007] seconds
/home/abdullah/demo/work/lm/news-commentary.lm: line 1476: warning: non-zero probability for <unk> in closed-
vocabulary LM
Finished loading LanguageModels : [0.049] seconds
Start loading PhraseTable /amd/nethome/abdullah/demo/work/model/phrase-table.0-0.gz : [0.032] seconds
Finished loading phrase tables : [0.005] seconds
IO from STDOUT/STDIN
دَرَسَ الطَّالِبُ مَادَّةَ الْاِمْتِحَانِ

```

Figure 6-2 output of the hybrid system

6.4 Errors Analysis

The presented hybrid system produced low error margins of 3.9% for the morphological errors and 10.6% for the syntactical errors. The following table 6.2 shows the change of diacritizing error margins for both models with the changing of testing corpus types.

Table 6-2 Morphological and Syntactical Error margins

Corpus data	Morphological Errors	Syntactical Errors	OOV
TST_CORP_1	5.4%	0%	0%
TST_CORP_2	7.8%	3.7%	5.5%
TST_CORP_3	12.1%	15.8%	19.3%
AVG=	8.43%	6.5%	8.26%

We had studied the system performance, memory and processing time needed for each model as shown in table 6-3.

Table 6-3 System size relating to Training corpus size

Training corpus	Language Model Size		
	Morpho-syntactic	Statistical	Hybrid
64K	108.6M	2.9M	111.5M
128K	108.6M	4.3M	112.9M
512K	108.6M	9.6M	118.2M
1.0M	108.6M	27.0M	135.6M
5.0M	108.6M	48.1M	156.7M

6.5 Hybrid diacritizing system vs. RDI's diacritizer

RDI diacritizer is a commercial product, it was built by native Arabic speaker based on factorizing Arabic words into roots and derivatives and find out the most possible combination. Although RDI's system uses disambiguation technique based on n-gram to take in consideration the statistical correlation among words in sentences.

Using RDI's demo website <http://www.rdi-eg.com/technologies/Diac.aspx> we applied a set of sentences on both systems all of these sentences are less than 10 words because of the limitation on RDI's demo website and record results of both systems in tables 6-4 to 6-7.

Table 6-4 Hybrid vs. RDI sentence 1

Source	يطلق الناس اسم تشامبينز ليجز على دوري رابطة الابطال
Hybrid	يُطَلِّقُ النَّاسُ اسْمَ تَشَامْبِيَانِزْ لِيَجِزْ عَلَى دَوْرِي رَابِطَةِ الْأَبْطَالِ
RDI	يُطَلِّقُ النَّاسُ اسْمَ تَشَامْبِيَانِزْ لِيُوجِزْ عَلَى دَوْرِي رَابِطَةِ الْأَبْطَالِ

Table 6-5 Hybrid vs. RDI sentence 2

Source	درس الطالب مادة الامتحان
Hybrid	دَرَسَ الطَّالِبُ مَادَّةَ الْإِمْتِحَانِ
RDI	دَرَسَ الطَّالِبُ مَادَّةَ الْإِمْتِحَانِ

Table 6-6 Hybrid vs. RDI sentence 3

Source	لا اله الا الله
Hybrid	لَا إِلَهَ إِلَّا اللَّهُ
RDI	لا اله الا الله

Table 6-7 Hybrid vs. RDI sentence 4

Source	ادخل الاسم والباسورد
Hybrid	ادْخُلِ الاسْمَ والباسورد
RDI	ادْخُلِ اَلِاسْمَ وَالْبَاسُوْرِدْ

According to the previous tables 6-4 to 6-7 it is noticeable that RDI's does not diacritize words' last character, the drawback of factorization can be found clearly in table 6-4 in word "ليجز", the Hybrid system found this word at the training corpus and diacritized it as "لِيَجْزُ" while RDI's system couldn't find its diacritization so it used factorization which affected word's structure and meaning "لِيُوَجِّزْ".

In table 6-7 the word "والباسورد" did not exist at training corpus for the Hybrid system and it couldn't find morphological analysis for it so it gave OOV error and returned it as it is without any modification. From analyzing RDI's result we can find out that word "باسورد" existed in RDI's training corpus and it tried to find its derivatives by applying Arabic morphological rules so it became "وَالْبَاسُوْرِدْ" which is not the optimal diacritization.

CHAPTER SEVEN

7 CONCLUSION AND FUTURE WORK

7.1 Conclusion

The hybrid system in this thesis studied the problem of Arabic diacritization and the techniques used to build full automated hybrid Arabic diacritizer using statistical machine translation method and morpho-syntactical analysis. The statistical method used in the system works on full form words is faster than the morpho-syntactical analyzer in terms of learning new words and diacritizing Arabic words. One of the advantages of this hybrid system is that it solved the Out-Of-Vocabulary problem in statistical model by applying morpho-syntactical analyzer. This system can be more complemented by adding morphological generator to extend training corpus. Another advantage of this hybrid system is that it shows competent error margins compared with other recent systems work on the Arabic diacritization problem. The accuracy rate of this system is 99.4% which is higher than rates listed in tables 2-1 and 2-2. These advantages are the results of implementing morpho-syntactical diacritization as a post-check on the statistical diacritizer's results. So this system is considered as a new technique in Arabic language processing. The effective results of this system are obtained from using a larger domain of Arabic corpus than the reported in other systems under realistic conditions.

7.2 Future Work

This system shows encouraging results and was built as independent modules with API and SDK that can be reused separately or as one whole unit. Our future work can be summarized in the following points:

- 1- Increase the size of the training data to affect this increase on the statistical model.+6

- 2- Add syntactical linguistic rules and add semantic layer to increase the accuracy of the morpho-syntactical diacritizer.
- 3- Add syntactical tags to the statistical model in order to merge grammatical rules with statistical probabilities.
- 4- Categorize training corpus to defined domains and add Weight to each category. This weight will be used at the statistical model to define which category text belongs to.
- 5- Apply this hybrid technique in other problem domains such as part of speech tagging, speech recognition content classification and many other domains.

REFERENCES

1. Al Badrashiny M. A., (2009), AUTOMATIC DIACRITIZER FOR ARABIC TEXTS Master thesis, Faculty of Engineering, Cairo University.
2. Alghamdi , M. , Khursheed , M. , Elshafei , M. , Alhargan , F. , Alkanhal , M. , Alshamsan , A. , ... , & Almuhtasib , H., (2006), Automatic Arabic Text Diacritizer , Technical report, available: <http://www.mghamdi.com/ATD.pdf>.
3. Alghamdi , M. , Muzaffar , Z. & Alhakami , H., (2010), Automatic Restoration of Arabic Diacritics: A Simple, Purely Statistical Approach , The Arabian Journal for Science and Engineering, Vol.35,No.2C
4. Ali Farghaly and Khaled Shaalan, (2009), Arabic Natural Language Processing: Challenges and Solutions, ACM Transactions on Asian Language Information Processing (TALIP).
5. Alshawi, H., Douglas, S., and Bangalore, S., (2000), Learning dependency translation models as collections of finite-state head transducers Computational Linguistics, 26(1):45-60.
6. Attia, M., (2004), Arabic Orthography vs. Arabic OCR, Multilingual Computing & Technology magazine www.Multilingual.com, Dec. 2004.
7. Ayan, N. F. and Dorr, B. J. (2006). Going beyond AER: An extensive analysis of word alignments and their impact on MT. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 9-16, Sydney, Australia
8. Badawi, Elsaid, Mike G. Carter, and Adrian Gully. 2004. Modern Written Arabic: A Comprehensive Grammar. Routledge.

9. Beesley, K. R., 1998, Arabic Morphological Analysis on the Internet, Proceedings of the International Conference on Multi-Lingual Computing, Cambridge, England.
10. Beesley, Kenneth R. and Lauri Karttunen. 2003. Finite State Morphology. CSLI Studies in Computational Linguistics. Stanford, California: CSLI Publications.
11. Blunsom, Phil, Trevor Cohn, and Miles Osborne. 2008. Bayesian synchronous grammar induction. In Neural Information Processing Systems (NIPS).
12. Brown, P. F., Della-Pietra, S. A., Della-Pietra, V. J., and Mercer, R. L. (1993). The mathematics of Statistical Machine Translation: Parameter estimation. Computational Linguistics, 19(2):263 -311.
13. Buckwalter, Tim. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. LDC catalog number LDC2002L49, ISBN 1-58563-257-0
14. Cavalli-Sforza, Violetta, Abdelhadi Soudi, and Teruko Mitamura. 2000. Arabic Morphology Generation Using a Concatenative Strategy. In Proceedings of NAACL 2000, pages 86 -93, Seattle.
15. Chen, B. and Federico, M. (2006). Improving Phrase-Based statistical translation through combination of word alignment. In FinTAL - 5th International Conference on Natural Language Processing, pages 356 -367, Turku, Finland
16. Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In Proceedings of ACL-05, pages 263 -270. Ann Arbor, MI.
17. Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39(1):1 -21.

18. DeNero, John, Alexandre Bouchard-Cote, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 314 -323. Honolulu, Hawaii.
19. Ding, Y., Gildea, D., and Palmer, M. (2003). An algorithm for word-level alignment of parallel dependency trees. In Machine Translation Summit IX, pages 95 -101, New Orleans, LA.
20. El Dada, Ali and Aarne Ranta. 2006. Open Source Arabic Grammars in Grammatical Framework. In Proceedings of the Arabic Language Processing Conference (JETALA), Rabat, Morocco, June 2006. IERA
21. Elaine Rich and Kevin Knight. Artificial Intelligence. McGraw Hill companies Inc. (2006).
22. Elsharifi , M. , AlMuhtasib , H. , & Alghandi , M., 2006, Machine Generation of Arabic Diacritical Marks , The International Conference on Machine Learning; Models, Technologies & Applications (MLMTA 06)
23. F. Och and H. Ney, The alignment template approach to statistical machine translation, Computational Linguistics, vol. 30, no. 4, pp. 417–449, December 2004
24. Fatehy, N., 2005, An Integrated Morphological and Syntactic Arabic Language Processor Based on a Novel Lexicon Search Technique , master thesis, Faculty of Engineering, Cairo University, 1995.
25. Fischer, Wolf Dietrich. 2001, A Grammar of Classical Arabic. Yale Language Series , Yale University Press, third revised edition. Translated by Jonathan Rodgers

26. Fischer, Wolfdietrich. 2001. A Grammar of Classical Arabic. Yale Language Series. Yale University Press, third revised edition. Translated by Jonathan Rodgers.
27. Forsberg, Markus and Aarne Ranta. 2004. Functional Morphology. In Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming, ICFP 2004, pages 213 -223. ACM Press.
28. Fraser, A. and Marcu, D. (2007b). Measuring word alignment quality for Statistical Machine Translation. Computational Linguistics, 33(3):293 -303.
29. Galley, Michel, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What is in a translation rule In Proceedings of NAACL-04, pages 273 -280. -
30. -Habash , N. , & Sadat , F. . 2006 -Arabic Preprocessing Schemes for Statistical Machine Translation , Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 49?52 , New York. -
31. -Habash, Nizar, Owen Rambow, and George Kiraz. 2005. Morphological Analysis and Generation for Arabic Dialects. In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 17?24, Ann Arbor, Michigan. Association for Computational Linguistics -
32. -Haji -, Jan, Otakar Smr -, Petr Zem nek, Jan -nidauf, and Emanuel Be -ka. 2004b. Prague Arabic Dependency Treebank: Development in Data and Tools. In NEMLAR International Conference on Arabic Language Resources and Tools, pages 110 -117. ELDA.
33. Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer,

- Brooke Cowan, Wade Shen, Christine Moran, and Ondrej Bojar. Moses: Open source toolkit for statistical machine translation. pages 177-180, 2007.
34. Holes, Clive, 2004, *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics, Georgetown University Press.
 35. Holes, Clive. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.
 36. Huet, Gérard. 2002. *the Zen Computational Linguistics Toolkit*. ESSLLI Course Notes, FoLLI, the Association of Logic, Language and Information
 37. J.R.R. Leavitt, D.W. Lonsdale, and A.M. Franz. A reasoned interlingua for knowledge-based machine translation. In *Proceedings of the 10th Canadian Conference on Artificial Intelligence*, 1994.
 38. Johnson, Mark. 2007. Why doesn't EM and good HMM POS-tagger? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296-305. Association for Computational Linguistics, Prague, Czech Republic.
 39. Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, Demonstration Session*, pages 177-180.

40. Lagally, Klaus. 2004. ArabTEX: Typesetting Arabic and Hebrew, User Manual Version 4.00. Technical Report 2004/03, Fakultät Informatik, Universität Stuttgart, March 11.
41. Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 104-111, New York, NY.
42. M. Attia, 2000, A Large-Scale Computational Processor of the Arabic Morphology, and Applications, M.Sc. thesis, Dept. of Computer Engineering, Faculty of Engineering, Cairo University, 2000.
43. M. Attia, 2005, Theory and Implementation of a Large-Scale Arabic Phonetic Transcriber, and Applications, PhD thesis, Dept. of Electronics and Electrical Communications, Faculty of Engineering, Cairo University, Sept. 2005.
44. M. R. Costa-jussa and J. Fonollosa, Improving the phrase-based statistical translation by modifying phrase extraction and including new features, Proceedings of the ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond
45. M. Yaseen, et al., 2006, Building Annotated Written and Spoken Arabic LRs in NEMLAR Project, LREC2006 conference <http://www.lreconf.org/lrec2006>, Genoa-Italy, May 2006.
46. Ma, Y. and Way, A. (2009a). Bilingually motivated domain-adapted word segmentation for Statistical Machine Translation. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), pages 549-557, Athens, Greece.

47. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W., the Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. Proceedings of the Arabic Language Technologies and Resources Intl. Conference; NEMLAR, Cairo
48. Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP.
49. Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313-330.
50. Marino, J. B., Banchs, R. E., Crego, J. M., de Gispert, A., Lambert, P., Fonollosa, J. A. R., and Costa-juss`a, M. R. (2006). N-gram-based Machine Translation. *Computational Linguistics*, 32(4):527-549
51. Mario F. Triola, 2010, *Elementary Statistics*, Addison Wesley, Longman.
52. Matusov, E., Zens, R., and Ney, H. (2004). Symmetric word alignments for Statistical Machine Translation. In Proceedings of the 20th international conference on Computational Linguistics (COLING 2004), pages 219-225, Geneva, Switzerland.
53. Meftouh , K. , Smaili , K. , & Laskri , M., 2008, Arabic statistical language modeling 2nd International Conference on Arabic Language Resources and Tools , Cairo, Egypt.
54. Och, F. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19-51.
55. P. Koehn, F. Och, and D. Marcu, Statistical phrase-based translation, Proc. of the Human Language Tech-nology Conference, HLT-NAACL'2003, May 2003.

56. Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lerty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguist.*, 16(2): 79-85, 1990
57. R. Zens, F. Och, and H. Ney, Improvements in phrase-based statistical machine translation, *Proc. Of the Human Language Technology Conference, HLT-NAACL'2004*, pp. 257–264, May 2004
58. Ramsay, Allan and Hanady Mansur. 2001. Arabic morphology: a categorial approach. In *EACL 2001 Workshop Proceedings on Arabic Language Processing: Status and Prospects*, pages 17-22. Toulouse, France.
59. Shaalan , K. , Abo Bakr , H. M. , & Ziedan , I. 2008 , A Statistical Method for Adding Case Ending Diacritics for Arabic Text, *Language Engineering Conference*, pp. 225-234 . Cairo, Egypt
60. Smr, Otakar and Petr Pajas. 2004. MorphoTrees of Arabic and their Annotation in the TrEd Environment. In *NEMLAR International Conference on Arabic Language Resources and Tools*, pages 38-41. ELDA.
61. Smr, Otakar. 2007. *Functional Arabic Morphology. Formal System and Implementation*. Ph.D. thesis, Charles University in Prague
62. Stump, Gregory T. 2001. *Intentional Morphology. A Theory of Paradigm Structure*. Cambridge Studies in Linguistics. Cambridge University Press
63. Vilar, D., Popovic, M., and Ney, H. (2006). AER: Do we need to improve our alignments In *Proceedings of the International Workshop on Spoken Language Translation*, pages 205-212, Kyoto, Japan
64. Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *COLING-96*, pages 836-841.

65. W. John Hutchins and Harold L. Somers. An Introduction to machine translation. Academic Press, 1992.
66. Werner Winiwarter. Machine translation using corpus-based acquisition of transfer rules. In Proceedings of the Second International Conference on Digital Information Management, pages 345-350, Lyon, France, 2007a. IEEE Engineering Management Society.
67. Werner Winiwarter. WETCAT – Web-enabled translation using corpus-based acquisition of transfer rules. In Proceedings of the Third IEEE International Conference on Innovations in Information Technology, Dubai, United Arab Emirates, 2006.
68. Wu, D. (1997). Stochastic Inversion Transduction Grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377-403.
69. Yorick Wilks. Machine Translation: Its Scope and Limits. Springer Publishing Company, Incorporated, 2008. ISBN 0387727736.
70. Zitouni, I. , Sorensen , J. S. , & Sarikaya , R., 2006, Maximum Entropy Based Restoration of Arabic Diacritics, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, page 577-584, Sydney , Australia.