uOttawa

L'Université canadienne
Canada's university

Towards the Development of an Automatic Diacritizer for the Persian Orthography based on the Xerox Finite State Transducer

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the PhD degree in Linguistics

Department of Linguistics
Faculty of Arts
University of Ottawa

Declaration

I hereby declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Professor Paul Hirschbühler and my co-supervisor Professor Diana Inkpen for their continuous support during my graduate studies at the linguistics department, for their patience, motivation, enthusiasm, and immense knowledge.

Besides my supervisors, I would like to thank the rest of my thesis committee: Prof. Eric Mathew, Prof. Jeff Mielke, Prof. John Goldsmith, and Prof. Paribakht for their encouragement, insightful comments, and hard questions.

My sincere thanks also go to Dr. Ken Beesley, Dr. Lauri Karttunen, Mahmoud Abu Nasser, and Tim Mahrt for their insightful comments and help with the Python programming language.

Last but not the least, I would like to thank my family: my parents Fatemeh Amin and Mehdi Nojoumian, for their love and support throughout my life, and my wife, Leila Ahmadpanah, first for her contributions in the evaluation of the thesis results and graphic designs and second for her sincere love and support throughout our life.

I would like to present this thesis to my lovely kids: Parham and Roya.

Author's Statement

I would like to hereby acknowledge that parts of chapter one "Background" is an enhanced, modified and updated version of my paper titled "*Morphological Priming of Multi-morphemic Persian Words in Mental Lexicon*" (Nojoumian et al., 2006).

Parts of chapter five "Resolving Morphological Ambiguity" is an adaptation of my second comprehensive exam paper titled "*Processing Phonological and Semantic Ambiguity, Disambiguation of Heterophonic & Homophonic Homographs in Persian Using Semantic Priming* (Nojoumian, 2008).

Abstract

Developing an Automatic Diacritizer for the Persian Orthography
based on the Xerox Finite State Transducer Technology

Due to the lack of short vowels or diacritics in Persian orthography, many Natural Language Processing applications for this language, including information retrieval, machine translation, text-to-speech, and automatic speech recognition systems need to disambiguate the input first, in order to be able to do further processing. In machine translation, for example, the whole text should be correctly diacritized first so that the correct words, parts of speech and meanings are matched and retrieved from the lexicon. This is primarily because of Persian's ambiguous orthography. In fact, the core engine of any Persian language processor should utilize a diacritizer and a lexical disambiguator. This dissertation describes the design and implementation of an automatic diacritizer for Persian based on the state-of-the-art Finite State Transducer technology developed at Xerox by Beesley & Karttunen (2003). The result of morphological analysis and generation on a test corpus is shown, including the insertion of diacritics.

This study will also look at issues that are raised by phonological and semantic ambiguities as a result of short vowels in Persian being absent in the writing system. It suggests a hybrid model (rule-based & inductive) that is inspired by psycholinguistic experiments on the human mental lexicon for the disambiguation of heterophonic homographs in Persian using frequency and collocation information. A syntactic parser can be developed based on the proposed model to discover Ezafe (the linking short vowel /e/ within a noun phrase) or disambiguate homographs, but its implementation is left for future work.

Table of contents

List of Figures and Tables

1 Introduction

Recent breakthroughs in information technology have been made possible in part by computational linguistics research. Text-to-speech (TTS), machine translation (MT) and automatic speech recognition (ASR) applications are widely used by individuals and institutions. For example, modern companies and government organizations try to use efficient automatic dialogue systems to handle high volumes of their client phone calls. TTS systems can be used by the blind and by reader gadgets for millions of readers.

While the core technologies behind these innovative applications are language independent, each language demands and requires its own localizations and adaptations. This is because of differences in language orthographies and structures. For example, the Unicode standard was created, years after the introduction of ASCII code, in order to handle the demands of the word processors requiring different alphabetic characters.

The Persian language is fairly new to the field of computational linguistics. It is mainly because of lack of resources and expertise. However, it is progressing in the field very fast and catching up with other languages. In order to make this progress possible, the main work has already been started on the development of linguistic resources.

An ultimate goal of this study is to develop tools and applications that can be used in the development of linguistic resources. Developing more demanding TTS and ASR applications for the Persian language requires group projects and national resources. Nevertheless, the basic requirement for the development of more modern language applications is viable linguistic resources. This research has aimed to fulfill part of this basic requirement.

In the next section of this introduction, I will present the background information on the Persian language and orthography and state the main focus of my study. Then, I will explain the computational framework that I chose to use in this present work motivated by psycholinguistic research and give a brief summary of recent studies on the mental lexicon, including my past work on morphological priming of multi-morphemic Persian words (Nojoumian et al., 2006).

1.1 Background

Modern Persian[1] (or Farsi[2]) is the major language of Iran and a prominent language in Afghanistan[3] and Tajikistan[4]; it has the official-language status in all three countries. Persian is an agglutinative language of Indo-European origin that has adopted the Arabic orthographic system. Consequently, as with Arabic, the three main Persian short vowels are typically omitted in the written language. When they do appear, these three short vowels are /æ/,/e/, and /o/ represented in the orthography respectively by diacritics (ً), (ِ), and (ُ). While diacritics exist to represent these vowels in the written language, native-speakers are trained to read texts without them after their primary education. The Persian diacritics represent three absent short vowels within a word and one short vowel, known as Ezafe /e/, between certain elements of a noun phrase (NP) in the written language. In the following example (1a), the short vowels are absent in the orthography, while in (1b) they are present within individual words and between the two elements of the NP as Ezafe.

[1] This study will focus on modern standard Persian rather than colloquial or slang versions.
[2] The name of language in English is "Persian"; however, "Farsi" which is the Arabic name of Persian is also used locally to refer to this language. There is no [p] sound in standard Arabic and the substitute phone is [f].
[3] Locally called Dari or Eastern Persian
[4] Locally called Tajiki

1)
(a)

پدرم استاد فیزیک است.

/pdr-m astad fyzyk ast/
father-my-NSg professor-NSg physics-N-Sg is-Verb

(b)

پِدَرَم اُستادِ فیزیک اَست .

/pedær-æm aostad e fyzyk aæst/
father-my-NSg professor-NSg of-Ezafe physics-NSg is-Verb
(my father is professor of physics)

Ezafe links different elements within a noun phrase and sometimes it is written as a single morpheme /y/ if it follows the letter /h/, /A/ or /v/. Traditional grammarians like Khanlari (1994) defined Ezafe as a short vowel /e/ linking a noun and its complement (as another noun or pronoun; for example, in 1a "professor of-Ezafe physics") or a noun and its modifier (a noun and an adjective). More recent accounts on Ezafe by Ghomeshi (1997) and Samiian (1983) define Ezafe as an enclitic phoneme with a grammatical linking function. The diacritizer developed in this study is not going to insert Ezafe. However, a simple model is proposed that can be used as a start point for the development of a robust syntactic parser to deal with Ezafe. In order to develop a rule-based parser, the Ezafe phenomenon should be studied more in a comprehensive corpus.

The absence of the short vowels within a word can also create lexical ambiguities in the form of heterophonic homographs. Heterophonic homographs are words with identical written forms with two or more pronunciations, each associated with a different meaning. The following example shows two different pronunciations associated with the two different meanings of a single orthographic form. The frequently used pronunciation "/jæng/" means "war", whereas the infrequently used one "/jong/" means "show":

2) جنگ /jng/

(a) /jæng/ (war-NSg), frequency 684 جَنگ

(b) /jong/ (show-NSg), frequency 1 جُنگ

The main focus of this study is to develop a Natural Language Processing (NLP) application that inserts diacritics within a word and provides enriched part-of-speech information. A robust parser can use this information in the final stages of processing to insert Ezafe and disambiguate heterophonic homographs. A simple disambiguation model is also proposed in this study in order to pave the way for future research.

Different NLP techniques can be used in order to put diacritics or short vowels that are absent in the orthography. Among them are morphological analyzers and generators, statistical language processors and hybrid systems. Each of these systems might have shortfalls and advantages over the other ones. However, an efficient model would probably be the one that is capable of processing language like the human language processor.

One benefit of computational linguistics research is that computational linguistics research and NLP in particular have the potential to test formal language models and provide more insight into the cognitive mechanisms used in the production and perception of language, such as the mental lexicon. An expert intelligent system capable of understanding, recognizing, translating or retrieving information from natural languages requires an efficient morphological analyzer. Morphological analyzers are algorithms designed to analyze and retrieve information about word stems, morphemes and parts of speech. The output of a morphological analyzer may be fed into a parser or disambiguator for further processing and syntactic-semantic analysis.

Morphological analysis dates back to the 1950s. It was first used in machine translation systems. Early applications included spell-checkers, text-to-speech synthesizers, and information retrieval systems. Those applications used hard-coded C programs encoding spelling and phonological rules along with wordlists of stems and affixes, in order to shrink the lexicon as much as possible (Roark & Sproat, 2007).

Finite State Machines (FSM), as a formal mathematical model, revolutionized computational morphology. "FSM is claimed to be one of the most efficient methodologies able to analyze morphological processes" (Roark & Sproat, 2007:100). Ron Kaplan and Martin Kay developed FSMs into Finite State Transducers (FST) in the 1970s; however, FSTs were first applied within morphology by Koskenniemi in 1983 (Koskenniemi, 1983).

FSTs are sophisticated mathematical algorithms that can be used to encode language models by incorporating morphosyntactic feature-unification[5] methods. A typical FST consists of a root lexicon, a sub-lexicon of affixes and stem-affix combinations constrained by feature-unification rules. FSTs can be utilized in a range of computational applications, from morphological analyzers to statistical part-of-speech taggers (Roark & Sproat, 2007; Jurafsky & Martin, 2000).

Studying the human morphological processor will help to better understand the way that the human brain, and particularly the mental lexicon, works. It will also help bridge the gap between psycholinguistic and computational linguistic research and enables us to design robust NLP models that are closer to the efficient human morphological processor. It is also important for NLP theories to discover real-time mechanisms used by the human neurological processor to reflect grammatical knowledge (Phillips, 2005).

[5] Feature-unification methods use variables known as "flag diacritics" to empower Finite State Transducers with deterministic features.

One of the basic questions in psycholinguistic studies of morphology is how the human brain represents multi-morphemic words in the mental lexicon. Are these words represented as full forms or as roots and affixes? For example, are there multiple entries in the mental lexicon for "plays", "played", and "playing" or one main entry for "play" and several entries for inflectional morphemes "s", "ed" and "ing"?

The first hypothesis termed "full listing", referring to the representation of words in the mental lexicon as full forms, seems not to be feasible, especially for agglutinating languages like Turkish in which there are potentially 200 billion words (Hankamer, 1989). The "minimum redundancy" hypothesis, however, suggests that morphemes are the only representation of the words in the mental lexicon. As Jurafsky and Martin (2000) suggested, neither of the hypotheses are truly correct. Instead, some sort of morphological relationships are represented in the mental lexicon (Jurafsky & Martin, 2000:84).

Stanners et al. (1979) suggested that the inflectional forms of words are represented by morphemes whereas words composed of derivational affixes are represented only as a whole unit. Therefore, different inflections of the word "play" are represented as a root and three inflectional morphemes "s", "ed", "ing", but words like "useful" and "useless" are different entries in our mental lexicon (Stanners et al., 1979).

Several studies have conducted experimental research on morphology through the use of priming (e.g. Warslen-Wilson, 1994; Longtin et al., 2003). "In cognitive psychology, priming is the benefit to processing one stimulus as a result of already having encountered that stimulus or one similar ... Priming has often been used subliminally; where people can be given cues that they are not consciously aware of that nonetheless affect future

performance." (Davy, 2006)[6]. In the 'masked priming' technique, for example, the prime as a word, picture or audio stimulus[7] is presented to a human subject for as little as 16 milliseconds[8] and then masked by a second stimulus or the target word which appears for a longer duration, for example for 2 seconds.

In a lexical decision task experiment that uses masked priming, the human subject will be presented with two different scenarios. Either the prime or the target stimuli will be semantically related (cat and dog) or they will not be (cat and finance). Either of the stimuli could be nonsense words. The subject then has to decide whether the target stimulus is a valid word or not. The subject's reaction time and accuracy rates as dependent variables are used to measure the effectiveness of the (positive) prime in the activation of the target stimulus in the mental lexicon (Nojoumian et al., 2003).

The presentation of prime stimulus in such a short SOA is believed to be perceived by the brain unconsciously and informs our understanding about early activation of a lexical representation. Thus, the second stimulus requires less activation time in terms of making a conscious decision about it. This shortened activation is reflected in a faster reaction time by the subject, compared to the reaction time to an unrelated pair of stimuli. However, if the prime does not have a positive effect on activation, it might inhibit the activation process by slowing the reaction time and the stimulus might be ignored by the brain (Buchner et al., 2003).

Marslen-Wilson et al. (1994) used an auditory-visual cross-modal priming technique in which subjects were presented with a written target that was primed in a spoken sentence.

[6] Available online under the section of "Cognitive Psychology, Priming":
http://www.credoreference.com.proxy2.library.illinois.edu/entry/hodderdpsyc/priming
[7] Prime and target stimuli can be presented in different modalities such as visual and auditory.
[8] This period is called Stimulus Onset Asynchrony or SOA.

They showed that only semantically-transparent and morphologically-related words would prime their base in English. For example, the word '*government'* primes its morphologically-related base '*govern'* but *'department'* would not prime *'depart'*. They concluded that semantic transparency builds the organization of mental lexicon in a way that transparent words are stored in the mental lexicon as morphemic units, since they are morphologically close to the family of their bases, while opaque words are stored as a whole unit, since there is no semantic link within their morphemic units. The notion of semantic transparency refers to how clearly the meaning of a complex word is related to the meaning of the base it is derived from (Longtin et al., 2003). For example, in Persian, "the word سرانگشت*[**sar**angosht]* (fingertip) is semantically related to its base انگشت *[angosht]* (finger) and is considered semantically transparent. If the two morphemes of a bi-morphemic word are not semantically related, they might only be etymologically related. The opaque word سرکش ***[sar**kesh]* (bandit) is not semantically related to its base کش *[kesh]* (elastic), but has separate morphemes" (Nojoumian et al., 2006:31).

Experiments involving masked priming in Hebrew by Frost et al. (1997) and cross-modal priming in Arabic by Boudelaa & Marslen-Wilson (2000, 2001) both showed a significant priming effect for morphologically-related pairs, regardless of semantic transparency, while priming was stronger for transparent pairs. Therefore, Semitic languages like Hebrew and Arabic, with non-concatenative morphology, prime morphologically related words, regardless of their semantic transparency. This contrasts with English, a concatenative language, which only primes morphologically related words that are semantically transparent. Logtin et al.'s (2003) experiment on French revealed a significant facilitation in a masked priming paradigm for morphologically related words, but no role for

semantic transparency. However, in an auditory-visual cross-modal priming paradigm, only semantically transparent pairs showed a significant priming effect for French. Therefore, when the prime is presented consciously (auditory), the transparency factor plays a significant priming role, contrary to the situation in which the prime is presented unconsciously and is masked (Nojoumian et al., 2006).

The first experiment run by Longtin et al. (2003) showed that the priming effect was the result of a decomposition process in the mental lexicon. Longtin et al. (2003) compared this finding to a connectionist model that was consistent with the results, and suggested that all words that are formally complex undergo the process of morphological decomposition. If the prime is transparent, then the morphological decomposition is done successfully and the prime stem activates the target; if not, the morphological decomposition fails and the word is considered to be a single unit, which fails to activate the target word. Thus, the success of the process is shown by the priming of the related target. This was shown in the second experiment by Longtin et al. (2003) in which only semantically-transparent words primed their related targets.

Following Longtin et al's. (2003) experiments on French, Nojoumian et al. (2006) ran a lexical decision task experiment using a masked priming paradigm on Persian multi-morphemic words[9]. Nojoumian et al. (2006) obtained significant facilitation only for pairs that were morphologically and semantically related, i.e., a simple word was recognized faster if it was preceded by a semantically transparent related word. Subjects' mean reaction times (RTs) with two main factors of relatedness and transparency were submitted to a 2 x 3 design (related vs. unrelated) x (transparent vs. opaque vs. orthographic) within-subjects analysis of variance (ANOVA). It showed that the main effect of transparency (semantically transparent, opaque,

[9] Analysis of variance is an extension of Nojoumian et al. (2006).

or orthographic) was significant (F(1,15)=11.362, *p=0.004*, MSe=1658) [10]. The main effect of Relatedness (related, unrelated) was also significant (F(1,15)=6.372, *p=0.023*, MSe=772) which further confirmed that responses were faster to related semantically transparent targets than to unrelated non-transparent ones. However, the interaction between transparency and relatedness was not significant (F(1,15)=0.47, *p=0.5*, MSe=2398). To further investigate the significant interaction, separate ANOVA's were performed at each level of transparency (transparent, opaque, and orthographic). It was found that reaction times (RT) to targets related to the transparent meaning of the prime were faster than responses to unrelated targets (F(1,15)=8.48 *p=0.01*, MSe=543). However, relatedness was not significant for opaque (F(1,15)=0.8, *p=0.38*, MSe=826) and for orthographic pairs (F(1,15)=0.427, *p=0.52*, MSe=1800). Therefore, a simple target word was not primed if the prime was not transparently-related to it.

The incompatibility of some of the results of this research with previous research on French and English, and in particular the lack of interaction between the transparency and relatedness factors, might be due to artifacts of unrelated stimuli or other experimental factors. Nojoumian et al. (2006) suggested that using data from more subjects[11] and more accurate frequency data for word pairs may increase the reliability of the results.

In a recent similar study that was done with 180 participants, Najafian (2008) took frequency into account and concluded that highly frequent words are somehow lexicalized in the mental lexicon, meaning that their access is word-based rather than root-based, but words with low frequency are decomposed and accessed through their root morpheme. Najafian did

[10] MSe= Mean Standard-deviation error
[11] Our experiment was run on only 16 participants, whereas Longtin et al. (2003) ran their experiment on 43 participants.

not find any significant effect of transparency in accessing multi-morphemic words in the mental lexicon (Najafian, 2008).

The aforementioned studies on different languages suggest that a kind of morphological decomposition occurs in recognizing multi-morphemic words in the human mental lexicon. This can be thought of intuitively as like when we encounter an unknown word in the target language and try to decompose it to guess its meaning.

However, further studies need to be done in different languages before clear conclusions can be reached about the decomposition of multi-morphemic words in the mental lexicon. Nevertheless, studying the mental lexicon may benefit NLP research. Further, the human lexical processor, as an efficient system, can be modeled in order to develop language engineering applications. NLP research, in return, is able to shed light on the understanding of lexical access in the human mental lexicon by examining simulation models of the human lexical processor. Now the questions are: "can we examine or simulate the same technique shown to be used by the human mental lexicon in a computer application? Can we enable a computer application to read and analyze words and sentences?". The answers are "yes".

NLP research has suggested different models and techniques for dealing with problems like diacritization, namely: rule-based, statistical and hybrid morphological analyzers and generators. If morphological analysis or generation is one of the techniques human mental processor benefits from in dealing with words, then a simulation of this technique might help computers do the same job as efficiently as humans do. Storing a limited inventory of roots and constraint rules would be productive and efficient in analyzing and generating diacritization for new words if they result in sufficiently accurate output. That

is why, for languages like Persian in which large diacritized and tagged corpora do not exist yet[12], choosing a morphological analyzer and generator for the diacritization problem make sense. For doing the diacritization and tagging task, a two-level morphological analyzer or generator seemed to be a reasonable choice because of its dual task nature -- it analyzes the words to find morphemes and part-of-speech tags and it generates the missing vowels simultaneously. The diacritizer, which is developed using the Xerox FST (XFST) platform, is going to be explained and the results are going to be discussed in this thesis. This study will also present a model for resolving ambiguities raised by the Persian script complexities.

1.2 The Persian Automatic Diacritizer[13]

Since the diacritics or short vowels are missing in the Persian orthography, NLP applications like information retrieval and machine translation need to disambiguate the input words first, in order to retrieve the required information. In machine translation, the whole text should be correctly diacritized, meaning that missing short vowels are marked, so that the correct parts of speech (POS) and meanings are retrieved from the lexicon. In fact, a diacritizer should be included in the core engine of any Persian language processing application.

This thesis describes the design and implementation of an automatic morphological analyzer and generator for the Persian language based on the two-level morphological[14]

[12] There has been no large diacritized and tagged corpus in Persian at the time of writing this thesis. Otherwise, statistical methods could have been chosen as an alternative method for diacritization.
[13] The three absent short vowels in the orthography are called /harkat/ or /e'rAb/ ("diacritics") in Persian because they are placed on graphemes or letters.
[14] Analyzing the words to find morphemes and part-of-speech tags and generating the missing vowels simultaneously.

approach, using the Xerox Finite State Transducer tools developed by Beesley & Karttunen, 2003 & 2008[15], at Xerox Research.

Previous research in Persian NLP has mostly focused on the analysis rather than the generation aspect of Persian morphology. Past research in Persian morphological parsing, include: Azimzadeh & Arab (2007), Amtrup et al. (2000), Megerdoomian (2004), Perslex (Riazati, 1997), and Dehdari & Lonsdale (2002). Azimzadeh & Arab (2007) implemented an FST morphological parser utilizing feature structures and a POS tagger in order to solve ambiguities arising from homographs. The Shiraz NLP morphological analyzer is also a unification-based FST. The research done by Riazati (1997) and Dehdari & Lonsdale (2002) are both based on Koskenniemi's two-level morphological approach. Dehdari & Lonsdale used PC-Kimmo, and Riyazati used the Persian version of Englex. Although a number of these systems achieved an almost reliable coverage on existing language corpora, it seems that there is still a long way to go in order to achieve a comprehensive lexicon with morphotactic rules. I will closely study these works in this dissertation and compare my work with them when it is relevant.

The focus of this study is mainly on diacritization, but the issues that are raised by phonological and semantic ambiguities will also be studied and a few solutions will be proposed. These ambiguities are mostly caused by the Persian writing system in which short vowels do not have any representation in the orthography[16]. Persian morphology, particularly diacritization, introduces interesting challenges to FST, and raises linguistic

[15] The latest version of the XFST is Unicode compatible and better suited for Arabic scripting languages.

[16] Native speakers are able to read Persian sentences using their acquired reading skills and learn to read non-diacritized texts after four years of reading diacritized ones at primary schools, yet they may face problems in reading new words. In other words, native speakers are able to pronounce an unknown word correctly only when it is diacritized, so a computer should not be expected to do better in such cases.

issues that require further exploration. For instance, compound verbs with long-distance morphological dependencies, allomorphs and irregularities in writing conventions (space between morphemes in a word) are among the difficulties that a Persian FST would face.

This research will also utilize and examine the findings of psycholinguistic experiments on the human mental lexicon and build upon the author's comprehensive examination paper (Nojoumian, 2008) on the disambiguation of heterophonic and homophonic homographs in Persian. This design is anticipated to improve the analysis rate by incorporating disambiguation models in its post-processing stage.

1.3 Summary of Contributions

There are several contributions made by this thesis. The first contribution is an application that diacritizes as many Persian words as possible and provides enriched part-of-speech tags for the insertion of Ezafe and frequency information for the disambiguation of heterophonic homographs in a parser. This is done through a series of FST modules, such as tokenization and morphological analysis/generation, which are considered "prerequisite modules inside almost any system that performs syntactic parsing, and parsing itself is a required module inside almost any system for natural language understanding, question answering, discourse analysis, machine translation, etc." (Beesley & Karttunen, 2003:455).

The second contribution is the findings of two relevant psycholinguistics studies on the human mental lexicon. The goals of these studies were twofold: first, to bridge the gap between the relevant research in psycholinguistics and computational linguistics, and second, to increase our understanding of how the human lexical processor deals with morphological analysis and the resolution of word sense ambiguities.

The third contribution is a proposed disambiguation model and demonstration of an example in Persian based on the ordered meaning frequencies[17] of heterophonic homographs and collocations associated with them.

Finally, the last contribution is the development of an enriched lexical database from a text corpus containing valuable information such as diacritization, frequency information, semantic and syntactic features, parts of speech, and English equivalents for the Persian language. This database is easily expandable and portable to all XML compatible formats and can be used both for research and for developing commercial applications. Furthermore, the tagged and diacritized versions of the training and test corpus used in this study can be used to train statistical-based NLP algorithms.

1.4 Organization of the thesis

The second chapter elaborates the design of a typical morphological analyzer and explains the architecture of the system design based on the training corpus data in three main stages of morphological processing. The third chapter starts with a brief look at Persian morphology and the process of developing the corpus for this study. The third chapter will explain the extraction of different lexicons from the training corpus, morphotactic rules, and the finite state framework to be used. Chapter four explains the implementation and application of the diacritizer on the test corpus using the extracted lexicons and rules and also presents the evaluation of the results. Chapter five discusses psycholinguistic findings on the disambiguation of homographs in the mental lexicon, as well as computational approaches to dealing with the issue of word sense disambiguation. In this study I will examine the feasibility of a hybrid (inductive and rule-based) model to deal with the

[17] Frequent and infrequent meanings of a heterophonic homograph.

ambiguities associated with binary homographs in Persian and a rule-based model to deal with the insertion of Ezafe (the short vowel /e/ which links elements of an NP) within the constituents of a noun phrase in a Persian sentence. Furthermore, the results of these models will be discussed in a few examples. The final chapter presents conclusions and directions of future work. A sample of the source code for the transliteration function and the Persian phonetic table are presented in appendices at the end of the thesis[18].

[18] Source code, tools, lexical database, the test corpus and the results developed in this thesis are all available for readers' scrutiny.

2 System Design

The Persian diacritizer consists of several components and modules piped together in order to generate a fully voweled output text. In the XFST system, finite-state transducers that are arranged in a compositional cascade perform the preprocessing stages of normalization, tokenization and morphological analysis/diacritization. These transducers are non-deterministic and can produce multiple outputs depending on the ambiguities. However, by using feature-unification constraining rules known as "flags" and "ambiguity reduction strategies"[19], some of the potential ambiguities can be resolved in the lexicon, before doing any syntactic parsing.

2.1 The Core Analyzer and Generator

The main goal here is to develop the core lexical diacritizer first, and, if successful, develop other modules and combine them towards a robust full-text diacritizer.

The lexical diacritizer contains two main Lexical Transducer (LT) modules. The first LT will analyze the input word and strip the stem from its affixes, based on the morphotactic rules, and the second LT will generate the vowels and compile the phonological alternation rules. The output will be one fully voweled lexical item for non-ambiguous tokens and two or more items for lexically ambiguous tokens in this stage. The following diagram shows the lexical diacritizer design:

[19] The unique strategies introduced in this paper for the first time will dramatically reduce ambiguities associated with the languages whose orthography lack non-trivial vowels.

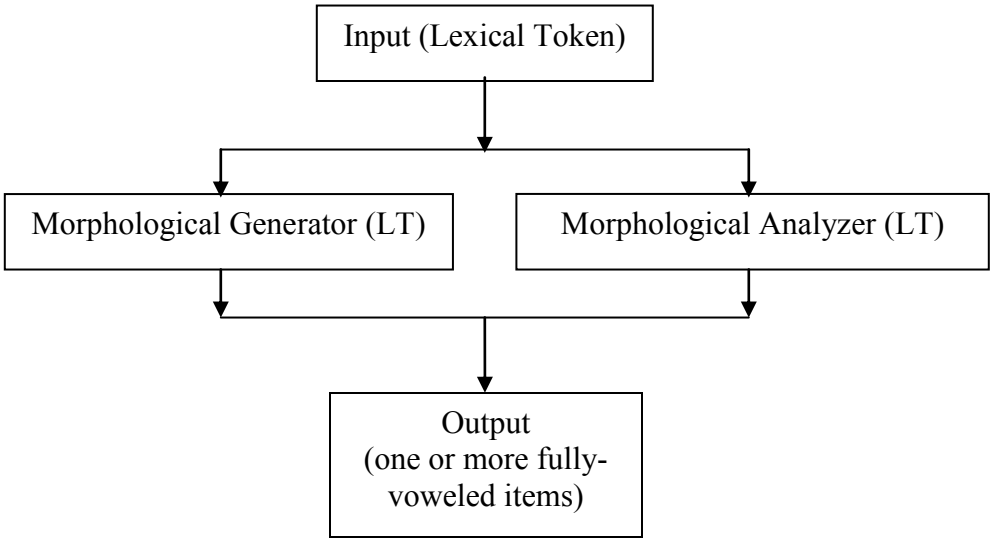


Figure 1: The unvoweled lexical token is voweled in a parallel processing method.

Examples 3a and 3b illustrate the unvoweled lexical input and the diacritized output at the first stage of the process:

3)
(a) پسران psrAn[20] (boys) → p..s..r-An+Noun+Pl → pe**s**æ**r**An+Noun+Pl

(b)بانوان bAnvAn (ladies) → bAn**v**-An+Noun+Pl → bAn**v**An+Noun+Pl

Following Nojoumian's (2003) text corpus, a comprehensive lexical database was developed containing syntactic and semantic information. It is mainly used to build the roots lexicon. The lexicon has to be exported to a text-like format in order to be encoded in the Xerox LEXC module. LEXC or **LEX**ical **C**ompiler is defined as "a high-level declarative language and associated compiler for defining finite-state automata and transducers" (Beesley & Karttunen, 2003:203).

[20] See the Persian phonetic & phonological table in Appendix I.

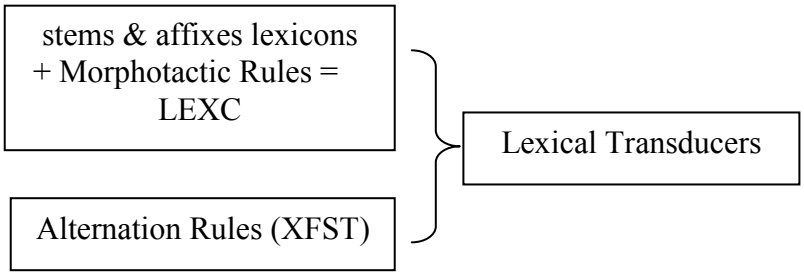


Figure 2: A Xerox Lexical Transducer composed of LEXC files & XFST Alternation Rules

By combining LTs and XFSTs with a tokenizer/normalizer and using an optimized lookup strategy, a running text can produce an analyzed and diacritized output. The tokenizer first disassembles the running text into tokens. Tokens include numbers (dates, times, zip codes, phone numbers), abbreviations, single words and eventually multi-words. Multi-word tokens represent compound nouns and verbs that are considered desirable tokens because they increase the rate of precision of POS taggers and disambiguators. A pre-processing module that is incorporated into the tokenizer makes it possible for the tokenizer to deal with critical multi-word entries. Correctly tokenized entries need to be normalized by substituting their spaces with a character and be sent to the core analyzer and generator by the lookup strategy and eventually get diacritized.

The lookup strategy uses modules in a serial order, to avoid overgeneralization, an issue raised by the non-deterministic nature of the FSTs and the large amount of morphological overlaps in Persian orthography[21]. Therefore, homographs, exceptions and function words[22] are looked up first and eliminated from the FST's input; then nouns, verbs and compounds are looked up sequentially. Using a callback strategy, tokens that are not

[21] The overlaps are the result of missing vowels. For example, the plural suffix /An/ in Persian can also be part of a root in /zehdAn/ (uterus-NSg). If roots are not constrained correctly, then /zehdAn/ (uterus-NSg) will be analyzed incorrectly as /zohd-An/ (abstinences-NPl) because /zohd/ is a singular noun.

[22] Function words may contain abbreviations, prepositions, and punctuation marks.

found, are sent to a guesser module so that their diacritics or POS tags are guessed. The guesser can utilize a recursive analysis to increase the diacritizer's chance in finding unknown tokens.

Naturally, ambiguous homographs (homophones and heterophones) will produce several outputs[23] with different POS tags or phonetic representations. A POS or semantic disambiguator should be used to further analyze or parse the generated output and choose the correct candidate while eliminating the others. Eventually, each token should have a unique fully-diacritized output. The last stage of the diacritization is the insertion of Ezafe. Ezafe is a single short vowel "/e/" which links and connects different elements within a noun phrase. A simple parser-like model that can predict Ezafe by just looking at the sequence of the POS tags within a sentence will be examine. The result, if correctly diacritized, Ezafe-inserted and tagged, would be a highly-desirable input for other natural language applications like a text-to-speech synthesizer, spell and grammar checker, machine translator, etc.

Figure 3, shows an ideal model for a full-text diacritizer in which the input from a running text was tokenized and fully-voweled and tagged in the output. This model is supposed to generate one output per input if there is no ambiguity associated with the input tokens.

[23] If rules are not carefully designed, even none-ambiguous words may generate several outputs and be overanalyzed.

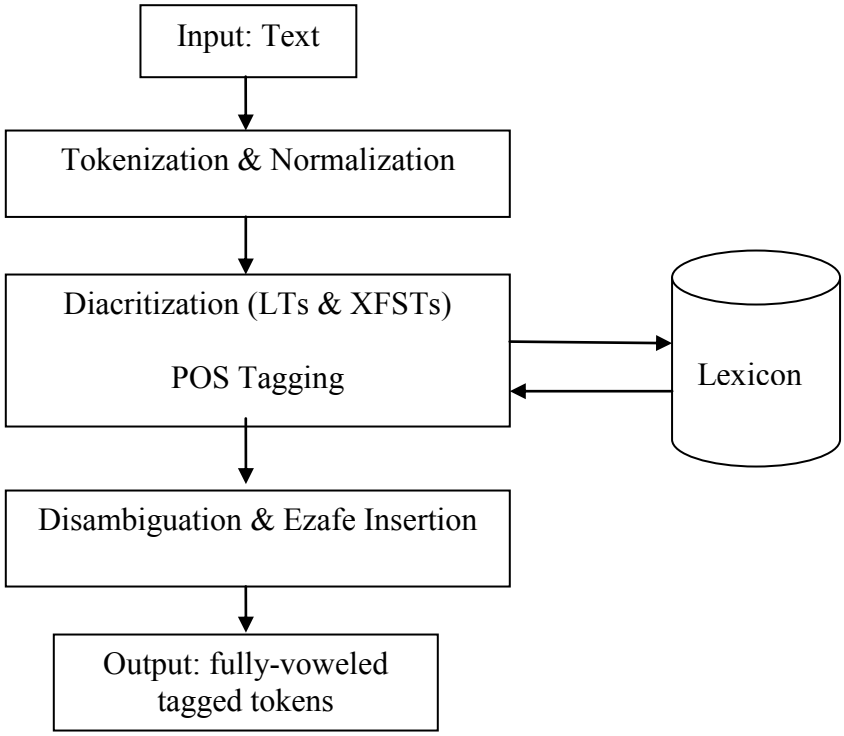


Figure 3: an ideal model for a full-text diacritizer for script ambiguous languages[24]

If the input token is an ambiguous homograph, then it will have multiple results, corresponding to different diacritic markings. Several strategies is explored to disambiguate multiple results by further looking at the words in neighborhood (N-grams) and by the syntactic-semantic features retrieved from the lexicon. It is worth noting that, if the ambiguity is not resolved in an early stage in a computational system, it may propagate itself into the other layers of the analysis and generate a higher complexity in later stages of analysis. For instance, in a machine translation system, the morphological analysis should deal with the lexical and morphological ambiguities; otherwise, in later stages of parsing and generation, the system encounters an exponential amount of possible analyses. To avoid this

[24] Arabic, Persian and Hebrew to name a few. Other diacritics like Sokun (mute) and Tashdid (gemination) are not usually distinctive in Persian.

problem, it would be more efficient to generate only a single disambiguated output from the morphological analyzer before feeding it to the later stages of the application.

In order to make a more precise analysis, a full-form lexicon is developed from a text corpus equipped with ambiguity resolution components, such as homograph N-grams (local collocations) and frequency numbers for the ambiguous entries. In order to eliminate several ambiguous outputs and choose the correct one, a disambiguation model is introduced, by incorporating the frequency and neighborhood features of Persian homographs. In short, the infrequent homograph candidate will be looked up first and if it is not found in pre-defined contexts or if there was no context at all, it is eliminated and the frequent candidate will be chosen as the winner. The reason why checking the context for the infrequent homographs is proposed is obvious; because the phonological representation of the infrequent meaning of a homograph contains less contextual information than its frequent counterpart. For example, in Farsi the frequent meaning of the homograph /jng/ pronounced as /jæng/ (war) has a frequency of 684, whereas its infrequent meaning representation pronounced as /jong/ (show) has a frequency of 1 in a one-million word text corpus. If the infrequent candidate is looked up, the right or left context is just checked once but for the frequent candidate more contexts needs to be checked to find the correct pronunciation. For an efficient coverage of homographs in different contexts, the proposed model should be trained on a large text corpus.

This model has the advantage of lesser context for the infrequent homograph candidate. This will be possible if the lexicon has accurate frequency counts for all possible binary homographs[25] and the text corpus has a sufficient amount of contextual collocations for the infrequent homographs. This model will only resolve lexical binary ambiguities and not

[25] Heterophonic homographs with only two alternative pronunciations

syntactic ones. A robust syntactic parser will still be required to resolve syntactic ambiguities and accurate insertion of Ezafe.

2.2 Processing Stages and Strategies

Automatic processing of a running Persian text presents different challenges that can be dealt with in different processing stages. The first stage should prepare the text for proper tokens. The main processing stage is where the tokens are analyzed, tagged and eventually diacritized. Final tuning will be dealt with in the post-processing stage. The three stages are discussed in more details in the next chapters.

2.2.1 Pre-processing

Pre-processing is the stage in which a running text, as the main input to an NLP application, is converted into tokens. It is vitally important to define what a token is, in order to prevent unexpected ambiguities and over-generations in later stages of the analysis.

A token in Persian is defined as a single or **spaced** multi-word, a punctuation mark, an abbreviation or a number that is separated by a single space. The token boundary as a physical space between the tokens has a contradictory function in this definition. In order to solve this problem, the tokenizer should be able to treat the space within a multi-word differently from the token boundaries. This can be done by incorporating a multi-word recognizer into the tokenizer. However, other inconsistencies may still exist in typed texts extracted from text corpora. The main problem originates from the Persian script and the lack of a widely accepted typing or writing convention. For instance, there is no accepted convention for typing electronic texts with correct spacing within words and sentences. In

Persian orthography, most of the letters connect in a typed or written word[26]. However, there are seven disjoined letters {ا، ر، ز، ژ، د، ذ، و} {/A/, /r/, /z/, /J/, /d/, /D/, /v/} that do not connect to their following counterparts. Persian letters change their shapes based on the four positions they appear in, to accommodate smooth connection. These positions are initial, middle, final-connected and final-standalone. The following example illustrates the letter /G/ in four different positions:

4) /G/={غ، ـغ، غ،ـغ}

/G/ in initial position: غـار /GAr/ (cave)

/G/ in middle position: مغـرب /maGreb/ (west)

/G/ in final-connected position: بالـغ /bAleG/ (mature)

/G/ in final standalone position: باغ /bAG/ (garden)

Example 4: Illustrated change of a Persian letter shape in different positions

Suppose that we have two words in a context, named A and B. In order to maintain a correct word boundary between them, a single space is required between the final letter of the word A and the initial letter of the word B. Now imagine that if word "A" ends in a disjoined letter like /v/ at its final position, word "B" which is next to it, comes without any space. Since words "A" and "B" do not connect in this case, typists will not notice that and sometimes leave it as it is. This is shown in the following example. This case is a typo that creates many errors in a typical text corpus.

5) **Word-A**[Space]**Word-B** --> و گفت /va goft/ (and said)

Word-AWord-B --> وگفت /vagoft/ (andsaid)

The other issue is due to the fact that some inflectional morphemes or words have the option not to connect to a stem. In handwriting, it is considered as an exception to the

[26] Persian is written from right to left.

writing convention, by considering a smaller space than the usual word boundary within the word of a sentence. This convention is supported by the Persian Academy of Language and Literature. Unicode standard introduced a grapheme with a pseudo-space functionality called Zero-Wide Non Joiner (ZWNJ) to address this requirement. This character does not have any representation in orthography, but assumes a dummy space between the two letters it mediates. In Microsoft Word a good typist should hold three keys “Ctrl”, “Shift”, and “2” together to put a ZWNJ invisible character within a word. Unfortunately, typists are reluctant to do this task, to save time. The following example illustrates this issue:

6)

a)	کتابها	/ketAbhA/	(books) - [Standard - no space]
b)	کتاب‌ها	/ketAb-hA/	(books) - [Standard - used ZWNJ]
c)	**کتاب ها**	/ketAb hA/	(book s) - [**Non-Standard** –spaced]

In this example, the plural morpheme /hA/ in 6a lacks the pseudo-spacing convention, but is still standard; in 6b, correct pseudo-spacing is illustrated and in 6c which is the non-standard sample, the morpheme is spaced wrongly. The last item (6c) creates problems for a text tokenizer, since it has the usual word boundary (space) in the wrong position. A good tokenizer should be able to solve these inconsistencies and produce a clean and correct token (without spelling error) for the analyzer. A possible solution to the first problem would be a pre-tokenizer with a spell-checking functionality and a solution to the second problem would be a multi-word tokenizer. Ghayoomi et al. (2010) enumerate several of the issues that they faced while developing a Persian text corpus. For the first issue, a frequency decision algorithm is suggested for the preferred correct spelling. They suggest other semi-automatic solutions to the script-related problems (Ghayoomi et al., 2010).

The second problem was dealt with by developing a multi-word recognizer for the tokenizer module. Sharifi-Atashgah and Bijankhan (2009) define multi-words as static Multi Token Units (MTU) belonging to a close set and dynamic MTUs belonging to an open set generative in nature. Therefore, static MTUs have space in certain constant positions within a single token while dynamic MTUs are long-dependencies and may include idioms and phrasal verbs. They defined templates for dynamic MTUs and claimed that these templates would correctly tokenize the dynamic MTUs in a running Persian text. The static MTUs, like separated plural, imperfective, superlative, nominal and Ezafe[27] morphemes and some non-complex compound phrases, however, can be hardcoded and tokenized correctly. Other linguists, like Riazati (1997) suggest dealing with MUTs in later stages of the process. However, wrongly tokenized entries will result is wrongly tagged items and will create problems in later stages of the processing, especially in word sense disambiguation. Megerdoomian (2004, 2006) considered static MTUs and claimed that a post-tokenization module was able to tokenize them correctly in her system. The system's tokenizer in this study is capable of dealing with static MTUs and some dynamic MTUs. It does not recognize long-distance dependencies like phrasal verbs or idioms, but it is capable of incorporating templates of this kind. Tokenizing long-distance dependencies as MTUs might contribute to better predictions in insertion of Ezafe and syntactic disambiguation of homographs. However, the accuracy rate in this system may show possible improvements to the results.

The other pre-processing task is normalization. This is the stage in which punctuation marks, letters, and formatted phrases, such as abbreviations, get tokenized and tagged

[27] Ezafe as represented by a single letter /y/ [ye] at the end of words with final long vowels /A/, /u/ and the silent /h/.

accordingly. Punctuation marks, for example, are vital to determine where to put Ezafe or eliminate wrong insertions. Among many POS taggers and morphological analyzers developed for Persian, a few mention the normalization stage. Megerdoomian's (2004, 2006) system recognizes date, numbers and Internet expressions. Shamsfard et al. (2009) claim that their system recognizes digits, numbers, dates, proper nouns, abbreviations and multi-part verbs. A good normalizer should not only recognize digits and different meaningful formats, but it should also tag them properly, in order to be useful in later stages of the processing. A normalizer is designed to recognize digits, single letters, proper nouns, web and email addresses, abbreviations and punctuation marks. The core processor can later tag digits as integers or decimals, dates, zip codes, bank accounts, credit or bankcard numbers, social security numbers, license plate numbers, currencies, times, phone numbers, and IP addresses. Proper nouns are also tagged as Persian, Arabic or Foreign proper nouns. Other signs, like parentheses and mathematical signs are all tagged as punctuations later on.

2.2.2 Core Processing

Once the text is tokenized and normalized, it is ready to be sent to the main processing stage, which is handled by the "Lookup" module. Untagged tokens should be diacritized in a sequential order, because FSTs are non-deterministic and prone to over-generation and over-analysis. Considering the missing vowels in Persian, there are many unknown overlaps between the roots and affixes. Recursive or circular rules are also an issue in over-generation of output results. Thus, the diacritizer sets aside the processed tokens in each sequence and sends the rest to the next step. The sequence or the lookup strategy is designed based on the syntactic and semantic categories of the tokens. Therefore, homographs are first to be analyzed and the function words, abbreviations, exceptions, and

numbers are the next. Then nouns, adjectives and adverbs are analyzed. Verbs and compounds are next and eventually punctuation and unknown tokens are analyzed and diacritized. This modular design has another advantage: it gives detailed information as to the amount of each analyzed category in the resulting statistics. The lexicons developed for the diacritizer have been separately encoded based on their grammatical categories in order to facilitate their maintenance as well as to avoid overlap. For example, homographs are analyzed before other categories since they contain enriched tags (frequency of meaning information) and should be excluded once identified in the text to avoid incorrect analysis and overlap. If they are tagged and analyzed incorrectly, the ambiguity propagates quickly in the upper level of analysis, often resulting in an exponential amount of analyzes by a parser.

2.2.2.1 Heterophonic Homographs

Persian has six vowels: three long and three short vowels. Short vowels or diacritics are /a/, /e/, and /o/ and are not usually written in Persian texts. While short vowels may play a significant role in the phonological realization of words, the grammatical functions determined by the syntax could also affect their phonological realization. Nevertheless, the nature of Persian phonotactical constraints emphasizes the role of the vowels within the Persian word syllable: C**V**(C)(C).

Two main problems are created because of short vowel omission in the orthography. The first problem has to do with the correct transcription of words out of context, whereas the second problem emanates from the Persian functional enclitic Ezafe, which is the unstressed short vowel /e/ within certain elements of a noun phrase constituent, mostly between two or more nouns and adjectives. Ezafe will be dealt with in the post-processing stage, because POS tags are required to disambiguate Ezafe.

The following example illustrates the lack of the Persian short vowels in an unstressed compound heterophonic homograph:

7) ببر /**bbr**/

Possible transcriptions:

/be.bar/	V-Imperative	(take)
/be.bor/	V-Imperative	(cut)
/babr/	N-Sg	(tiger)

The above example also illustrates the different readings of a heterophonic homograph in Persian. I will further discuss disambiguation strategies for this issue in later chapters and offer possible solutions to deal with it in a diacritizer. Nevertheless, homographs need to be tagged correctly in order to be disambiguated later. That is why simple unstressed homographs are first in the queue to get diacritized and tagged. The lexicon contains around 650 homographs with their relative frequency information. Since most of these homographs are binary (have only two readings), they are tagged as high- or low-frequency homographs (HH, HL). This invaluable information can be used to disambiguate homographs in later stages of the processing. Because of the binary nature of homographs, the output of this step is two or more results (candidates) for each single token that is going to be disambiguated in the post-processing stage. A model has been suggested to disambiguate homographs by choosing the best candidate based on their frequency and likelihood information or a weighted probability that factors in both pieces of information (see section 5.3).

2.2.2.2 Exceptions

Next, words belonging to the "exceptions" set are diacritized, tagged, and set aside from the rest of the main processing stage. This set contains function and stop words, proper

nouns, abbreviations and word numbers. Two-letter and overlapping roots that cause processing over-generation have also been included. This design will eliminate many lexical ambiguities. For instance, the two-letter noun root /Oz/ (greed) does not allow a possessive pronoun suffix and if it is included in the main lexicon, it will be wrongly analyzed as the non-existing word /Oz-mvn/ [Oz-emvn] (our greed) which is overlapped with another root /Ozmvn/ [Ozmvn] (test).

8)

آزمون /Oz-mvn/ (greed) --> آز- مون /Oz-emvn/ (our greed)
آزمون /Ozmvn/ (test) --> آزمون /Ozmvn/ (test)

Our design prevents these kinds of over-generation and wrong analysis by checking and isolating overlapped and lexicalized roots. An alternative solution to this issue is "flag diacritics" in XFST. Flag diacritics are user defined constraints which give a deterministic functionality to the non-deterministic FSTs. However, defining a huge set of flags makes the algorithm very complicated and prone to programming errors. Using the most productive word processes and morphotactics, several constraints were defined as "flag diacritics". Less productive word processes causing overlapping analysis were identified while training the diacritizer by the training corpus. These stems and morphemes were isolated from the main analyzer and the exception handler was created. This module contains roughly 10000 function words, numbers, proper nouns, abbreviations, exceptions and affixes.

2.2.2.3 Nouns, Adjectives, and Adverbs

After analyzing and diacritizing the ambiguous tokens, non-ambiguous items are analyzed. The main lexicon contains about 16000 nouns, adjectives and adverbial roots and

affixes. The morphotactic grammar analyzes and diacritizes words belonging to these categories as well as compound nouns, adjectives and adverbs. Recognized words that are diacritized and tagged in this stage are set aside and the rest of the unrecognized tokens are sent to the next step.

2.2.2.4 Verbs & Verb's Derivative Categories

Verbs and grammatical categories derived from verbs are analyzed here. Predicates are also analyzed in this module. To cover more compounds, two verb roots have been allowed in the rules. It should be mentioned that more than two verb roots might result in over-analysis and over-generation. However, since this is done within the last steps, over-analysis and over-generation are rare.

2.2.2.5 Punctuation

Punctuation marks and numbers are tagged in this module. Punctuation marks are very important to determine the correct place of the Ezafe. A coma inserted correctly will eliminate a wrongly inserted Ezafe because it is tagged as a punctuation mark and is considered a disjoiner within an NP. For a more accurate tag set, this module is flexible and can tag punctuations according to parsing requirements.

2.2.2.6 Unknown tokens

Unknown tokens may be words that are not covered in the main lexicon and may belong to proper noun or other categories. However, a callback strategy is developed to diacritize these tokens as well. The strategy is to strip and diacritize the affixes and tags them based on the striped affixes. However, the tag will carry a code that shows the root was not

recognized. Other unknown tokens are tagged as unknown. Depending on the number of unknown tokens, other statistical or rule-based strategies like syllabus analysis or pattern recognition for Arabic loanwords might be developed, subject to future works.

2.2.3 Post-processing

Diacritized and tagged tokens that are ambiguous need to be disambiguated. Furthermore, the enclitic Ezafe “e” needs to be inserted within elements of a noun phrase in a sentence. Because of the scope of this research and time limitations, a disambiguator or parser was not developed. However, the problem is discussed in detail and a simple model is developed to tackle both the ambiguity and the Ezafe issues. A sample of the test corpus has also been manually disambiguated using the developed model and the results will be presented.

3 Persian Morphology & Application Platforms

In this chapter, Persian morphology and word-formation processes will be briefly discussed, before the existing platform and technology for the implementation of a morphological analyzer and generator is explained.

3.1 A Brief Look at Persian Morphology

Persian morphology is concatenative in nature. In Persian, a simple word may consist of one stem plus one to five affixes (prefixes and suffixes). Persian affixation is a productive process in word-formation; therefore, morphological processes form many words. There are certain loanwords that should be treated as exceptions (about 15% of the lexicon).

3.1.1 Stems

A stem can be defined as a root plus derivational morpheme(s). Derivational morphemes are usually close to the root of the word. A simple word, however, can be a bare root or consist of an optional derivational and/or inflectional morpheme.

Word = ROOT + [derivational morphemes] + [inflectional morphemes]

3.1.2 Affixes

The Persian word-formation process allows both prefixes and suffixes. Prefixes are most productive in verbs. I was able to identify about 20 prefixes and 100 suffixes (80 derivational + 20 inflectional) in my training text corpus. The main Persian verb-formation process is compounding, because there are not too many simple verbs in Persian. Derivation is one of the productive word–formation processes in Persian, which is mostly based on derived stems of the simple verbs. The Persian clitics that show subject-verb agreements are

treated as inflectional morphemes. The following example illustrates a Persian complex noun with a prefix, verb root and two derivational morphemes:

9) ناتراوایی (impermeability)
/nA-**tarAv**-A-yy/
{Inflectional prefix + [[**verb root** + derivational morph.]+ derivational morph.]}

A word usually consists of zero to a maximum of three prefixes and zero to a maximum of five suffixes.

3.1.3 Word-formation Processes

Simple Words: One-lexical morpheme words like "book" /ketAb/.

Loanwords from languages like Arabic, Turkish, French and English are considered simple morphemes. This is a criterion to avoid analysis of loanwords because they are not usually productive. For example, the French loanword in Persian /dekorAsyvn/ (decoration), will not be decomposed to /dekor-Asyvn/ although the word /dekor/ exists in Persian. By convention, loanwords have been stored in the lexicon as single roots/words.

Compound Words: The combination of at least two lexical morphemes makes a compound word. Unfortunately, most of the compounds are written with a space in between them, which make them too hard to identify as a compound. (Three-stem compounds are not very frequent in Persian)

10) /ktAbxAnh/ کتابخانه
/ketAb-xAneh/
book-house → (library)
N+N → N

11) /brf pAk kn/ برف پاک کن
/barf-pAk-kon/
snow-clean-do → (wiper)
N+Adj+V → N

Complex Words: One lexical morpheme in addition to at least one grammatical morpheme.

12) /ketAb-y/ کتابی
book + indef. → (a book)
N+ infl. → N

13) /jangal-bAn-y/جنگلبانی
forest-keep + indef. → (forestation)
N+V+derv. → N

Compo-complex: Minimum two lexical morphemes (a compound) + minimum one grammatical morpheme.

14) /sarmA-xord-egy/ سرماخوردگی
cold-caught +indef. → (catching cold)
N+V+derv. → N

3.1.4 Compound Verbs

There are not too many simple verbs in Persian, some of which are no longer used. The verb compounding process, however, is a very productive process in Persian. Compound verbs are built by two processes, namely combination and incorporation.

Combination: In combination, a so-called light verb (either simple or linking) is combined with a noun, adjective, past participle, prepositional phrase, or adverb. Light verbs may include stative, inchoative and causative moods.

15) /xoShAl bvdan/ خوشحال بودن
happy + to be
Adj + verb → (to be happy)

/xasteh Sodan/ خسته شدن
tired + to become
Adj + verb → (to become tired)

/sedA kardan/ صدا کردن
sound + to do
Noun + verb → (to call)

Incorporation: In incorporation, the direct object loses its object marker /rA/, indefinite specifier /yy/, or pronouns and attaches to the verb as an incidental generic noun (without grammatical case, so it is a non-referential noun, i.e., no other noun in the context can refer

to it; it becomes part of the verb scope). Through incorporation, the verb becomes intransitive.

16) /GaDA xordan/ غذا خوردن
food + to eat (to eat food)
Noun + verb

بچه‌ها غذایشان را خوردند.
/baKe-hA GaDA-yeSAn rA xordand/
children food-their OM eat-3Pl-Past
(The children ate their food.)

بچه‌ها غذا خوردند.
/baKehA GaDA xordand/
Children food eat-3Pl-Past
(The children ate food.)

3.2 Technological Platforms

There are several technologies and platforms to do morphological analysis and generation. Technological advances made it possible for faster processing and larger memory capacity. However, developing a large and robust lexicon is a time intensive and laborious task, which requires professional linguistic knowledge. Therefore, it is still justifiable and more efficient to develop roots and affixes lexicons instead of full-form dictionaries. Nevertheless, developing a sizable text corpus and a lexicon of about 30,000 roots with accurate parts of speech tags, diacritization and semantic information would take several years with limited professional human resources. That is why this kind of research projects are more feasible to do if defined in a team framework.

At the time that this study was started, two-level morphological analysis had already been implemented by several methods and algorithms (see page 13). The latest at the time

was finite state transducers[28]. This method was chosen in this study because of its fast speed and flexibility to develop morphological analyzers and its powerful parallel processing capability, in which morphological analysis and phoneme manipulation can be done simultaneously. Furthermore, FSTs are fully reversible, meaning that developed systems by FSTs can be potentially used for analysis as well as generation.

"Finite State Automata and Transducers are among formal devices for encoding many language models, from morphological grammars to statistical part-of-speech taggers" (Roark & Sproat, 2007:1). These multi-functionality features of the FSTs and the fact that they are able to incorporate morphosyntactic feature-unification encouraged me to use them. Xerox has recently introduced an enhancement utility to their XFST interface, which works on a pattern matching design. At the time that this study was being tested on the test corpus, Xerox had not yet released the latest FST license for educational purposes.

The latest version of the FST system, which is based on a pattern-matching algorithm, would greatly enhance the tokenizer's performance. The older version of this application was not able to handle a large size of multi-units because of its longest-match algorithm. A good tokenizer for Persian should be able to recognize at least ten thousand or more locally dependent multi-units.

In the next section, the corpus development project that resulted in the creation of an enriched lexical database for the Persian language will be described first. Next, the basic structure of the Xerox Finite State Transducer (XFST) and the implementation of the Persian morphotactic and feature-unification rules in the FST platform will be explained in detail.

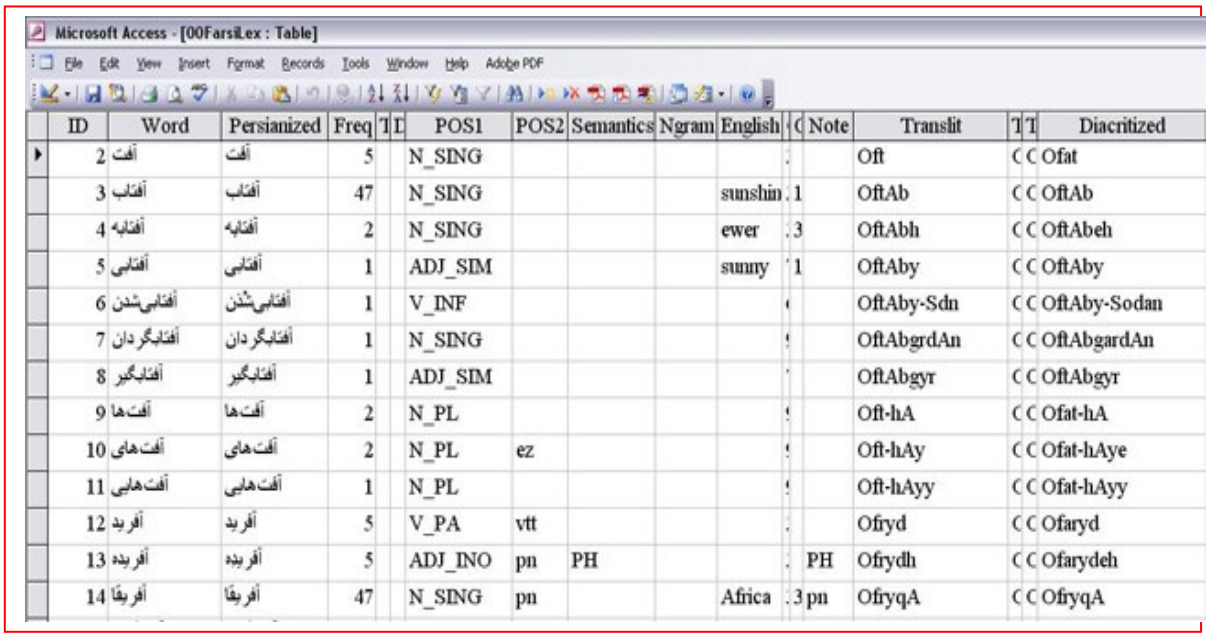
[28] Xerox has implemented a pattern matching approach recently. However, I could not access this new algorithm because of licensing issues.

3.3 Corpus Development for Persian

This task should indeed be considered a full database development project within the thesis. The reason why a corpus and a lexicon should have been developed for this research was the lack of available robust resources for the Persian language. At the time the development of a corpus and lexical database was started for this research (three years prior to writing the thesis), a few text corpora such as Nojoumian (2003) and Bijankhan et al. (2004e) corpora were available. However, because of the special needs of the current research, code incompatibilities and the sizable amount of errors in the previous corpora, I decided to develop a text corpus from scratch. The text corpus for this study was collected mainly from common texts extracted from Iranian daily news web sites[29]. The corpus covers at least ten different domains including, but not limited to, art, literature, culture, politics, sport, science, humanities, social sciences, economy, business, and religion. It was saved in raw text format and cleaned of garbage characters and error codes. The sources were chosen based on a quick survey of the closeness of the content to the standard Persian orthography. This ensured fewer coding and spelling errors. In order to clean the texts from unwanted characters like unnecessary Unicode control characters and double line/space, a few useful tools were developed with AWK, Perl and Python. In order to maintain the text corpus in an XML compatible framework, it was tokenized and imported to an Access database. All punctuation and text marks were preserved. The collected corpus contained about one million words. Two different text corpora were randomly extracted from the original corpus;

[29] Although developing text corpus for the purpose of research does not require any formal copyright clearance in Iran, the Iranian government is not a signatory to any international convention on intellectual properties and copyright treaties. The current Iranian government has not signed any copyright agreement with the United States of America and the government of Canada.
http://en.wikipedia.org/wiki/List_of_parties_to_international_copyright_agreements

one for the training set, consisting of about 90% of the original corpus and one for the testing set consisting of about 10% of the original corpus. A pure full-form wordlist was created from the training corpus containing 25 fields such as word, frequency, POS1, POS2, transliteration, and diacritized word. The following snapshot from Figure 4 shows a part of the main lexicon derived from the training corpus.



ID	Word	Persianized	Freq	POS1	POS2	Semantics	Ngram	English	Note	Translit	Diacritized
2	آفت	آفت	5	N_SING						Oft	Ofat
3	آفتاب	آفتاب	47	N_SING				sunshin	1	OftAb	OftAb
4	آفتابه	آفتابه	2	N_SING				ewer	3	OftAbh	OftAbeh
5	آفتابی	آفتابی	1	ADJ_SIM				sunny	1	OftAby	OftAby
6	آفتابی‌شدن	آفتابی‌شدن	1	V_INF						OftAby-Sdn	OftAby-Sodan
7	آفتابگردان	آفتابگردان	1	N_SING						OftAbgrdAn	OftAbgardAn
8	آفتابگیر	آفتابگیر	1	ADJ_SIM						OftAbgyr	OftAbgyr
9	آفت‌ها	آفت‌ها	2	N_PL						Oft-hA	Ofat-hA
10	آفت‌های	آفت‌های	2	N_PL	ez					Oft-hAy	Ofat-hAye
11	آفت‌هایی	آفت‌هایی	1	N_PL						Oft-hAyy	Ofat-hAyy
12	آفرید	آفرید	5	V_PA	vtt					Ofryd	Ofaryd
13	آفریده	آفریده	5	ADJ_INO	pn	PH			PH	Ofrydh	Ofarydeh
14	آفریقا	آفریقا	47	N_SING	pn			Africa	3 pn	OfryqA	OfryqA

Figure 4: Snapshot of the project's main lexicon in Microsoft Access

This lexicon contains about 40,000 words and provides valuable information such as word frequencies and frequencies of heterophonic homographs. The latter information has been extracted manually by checking and pronouncing each single homograph in context. Part of speech data has been labeled by the author and checked by another trained native speaker. A second POS (POS2) has been labeled if a homophonic homograph had two different parts of speech. The words were transliterated or Romanized by the Access internal programmed modules in order to avoid later incompatibility issues in text editors. Almost 100 different functions were written in Microsoft Visual Basics for Applications (VBA)

during the development of this database. The code for the Transliteration function can be found in Appendix II. The main challenge in developing such a lexicon was filling the diacritization field for each word. Several lexicons were extracted from the main lexicon. Verbs, function words, nouns, adjectives, adverbs, proper nouns and foreign words were distributed in different tables. Roots were extracted out of words and rules were written in LEXC files. Main POS tag names were adapted from Bijankhan (2004) in order to make them consistent with other corpora. The following table shows the 35 POS for the main full-form lexicon and the distribution of grammatical categories.

No.	POS1	Grammatical Category	Frequency	Relative
1	ADJ_CMP	Adjective Comparative	294	0.74
2	ADJ_INO	Adjective Objective	524	1.31
3	ADJ_ORD	Adjective Ordinal	69	0.17
4	ADJ_SIM	Adjective Simple	6293	15.74
5	ADJ_SUP	Adjective Superlative	241	0.60
6	ADV_EXM	Adverb of Example	20	0.05
7	ADV_I	Adverb Interrogative	28	0.70
8	ADV_NEG	Adverb Negative	23	0.05
9	ADV_NI	Adverb Non-interrogative	688	1.72
10	ADV_TIM	Adverb Time	142	0.35
11	CON	Conjunction	91	0.23
12	DET	Determiner	14	0.035
13	IF	IF	15	0.037
14	INT	Interjection	19	0.047
15	LET	Letter	25	0.063
16	N_ABR	Noun Abbreviation	73	0.18
17	N_AR	Noun Arabic	514	1.29
18	N_ENG	Noun English	2148	5.37
19	N_MORP	Noun Morpheme	11	0.027
20	N_NUM	Noun Number	128	0.32
21	N_PL	Noun Plural	6633	16.58
22	N_SING	Noun Singular	16783	41.96
23	OH	Onomatopoeia	34	0.085
24	P	Preposition	71	0.18
25	PP	Prepositional Phrase	160	0.40
26	PRO	Pronoun	51	0.13
27	PRO_REF	Pronoun Reflexive	7	0.017
28	PS	Phrase / Sentence	60	0.15
29	V_AUX	Verb Auxiliary	38	0.09
30	V_IMP	Verb Imperative	107	0.27
31	V_INF	Verb Infinitive	995	2.49
32	V_PA	Verb Past	1476	3.69
33	V_PRE	Verb Predicate	542	1.36
34	V_PRS	Verb Present	882	2.20
35	V_SUB	Verb Subjunctive	790	1.97
Total			**39995**	100

Table 1: Distribution of the POS of the main lexicon.

The POS2 field specifies a second part of speech for homophonic homographs as well as distinctions between proper nouns, location names, dates, numbers, compound nouns, etc. Other fields like Class were designed to specify the last phoneme of the words, especially if they are long vowels, because root words ending in long vowels behave differently during the affixation process. One of the advantages of developing such a valuable database is that it can be used for different NLP applications. It is also an invaluable resource for corpus linguistics research. At the time of writing this thesis, the author is unaware of any Persian lexical database with accurate diacritization of words, frequency information for homographs, and accurate part of speech and semantic information for each record. An English equivalent has even been included for each word to be used as a dictionary in machine translation research projects. Roots and base forms were extracted from the main word list tables in 10 different categories, namely homographs, exceptions, abbreviations, function words, proper nouns, numbers, predicates[30], main roots (nouns, adjective, and adverbs), verbs, and compounds.

To reduce the time-intensive manual work, especially on the diacritization task, an innovative method was used to semi-automatically diacritize Arabic loanwords and some Persian words with special syllabic patterns. These words are root-and-template based because Arabic morphology is mostly non-concatenative. Therefore, if templates are well defined in an algorithm, you might be able to insert some of the missing short vowels correctly. It should be mentioned, however, that algorithms which are used to diacritize the Arabic input cannot be used to diacritize Persian because of the non-concatenative nature of the Arabic language and different roles for the diacritics in the two languages. The short

[30] Predicates are words connected to the stative verb "to be" in written Persian. For example: /OmrykA-st/ [Noun-V_PRE] (it is America)

vowels that are absent in the Arabic orthography have syntactic functions (/o/ shows subject, /æ/ shows object and /e/ indicates possessive). Therefore, an Arabic diacritizer should put more emphasis on a syntactic parser than a morphological analyzer although it needs a morphological analyzer or POS tagger as well. The Arabic language is an inflectional language. It requires algorithms that are able to define inflectional patterns, roots and slots. Furthermore, many Arabic loanwords have a different pronunciation and even a different meaning in Persian. For example, the word /jAme?h/ means "society" in Persian but it means "university" in Arabic or the word /jonub/ (south) in Persian is pronounced as /janub/ in Arabic with the same meaning. Xerox has developed transducers that are non-concatenative and proper for the Arabic language. A number of applications have also been developed by Xerox for the Arabic language. Therefore, in designing templates for Arabic loanwords, pronunciation differences should be considered. For this study, thirty Arabic templates with potential consonants[31] as roots were defined. It is worth mentioning that even if some entries may incorrectly get diacritized, a lot of manual work is eliminated if the process is done step by step. Nevertheless, a final editing would be inevitable in order to ensure data accuracy.

If the algorithms are well designed and run in sequential steps, a significant amount of Arabic loanwords could be correctly diacritized semi-automatically because this allows for easier testing and debugging. An estimated 15% of the lexicon comprised Arabic loanwords. In order to maintain quality control, automatic diacritization was done in stages, and in each stage, diacritized lexical items were inspected thoroughly and incorrect vowel insertions were manually corrected. The following example shows an Arabic loanword in Persian and the algorithm used to automatically diacritize it:

[31] Four Persian consonants /g, K, p, Z/ do not exist in Arabic scripts.

17)

استخدام /AstxdAm/ N_SING (employment)

```
' A&st&CCAC >> Aeste CCAC >> AestexdAm (employment)
Function Arabic1(fldIn, PartOfSpeech As Variant) As String
Dim re As RegExp, strOut As String, Result As Boolean
strIn=Nz(fldIn)
POS=Nz(PartOfSpeech)
Set re=New RegExp
Result=Match(strIn,^Ast[btcjHxdMrzsSCZTDEGfqklmnvhIYU]{2}A[btcjHxdMrzsSCZTDEGfqklmnvhIYU]"
)
  If Result = True Then
     If (POS Like "AD*") Or (POS Like "N_S*") Or (POS Like "N_P*") Then
         strOut = LSubFunction("^Ast", "Aeste", strIn)
         Arabic1 = strOut
     End If
  End If
End Function
```

The above function encodes the Arabic template "A&st&CCAC" with three main consonants as its root (known as radicals) and two long vowels (/A/) and fills the two empty slots "&" with the missing short vowels (/e/). Romanized consonants were defined and matched through a regular expression. It is important to maintain the template as solid seven letters; otherwise, the error rate will be very high because of unknown overlaps with Persian roots. However, the Persian inflections of the loanword are not recognized because the template was limited to seven letters, the result showed above 85% accuracy rate with the same coverage rate. It is possible to tune the coded consonants by removing non-existing consonant clusters from the templates to get even higher accuracy rates. The following table lists some of the Arabic loanword templates in Persian that can be used to automatically diacritize words in the process of lexicon development:

Class[32]	Arabic pattern/ example	Template	short vowel insertion	Loanword example	number of detections	Part of speech	Suggested Regular Expressions [33]
2	اِستِفعال	AstCCAC	A**e**st**e**CCAC[34]	AestexrAj	87	N	^AstC{2}AC
3	مُستَفعِل	mstCCC	m**o**st**a**CC**e**C	mostaxdem	32	N, Adj	^mstC{3}
4	مُتِفاعِل	mtCACC	m**o**st**e**CAC**e**C	moteqAren	42	Adj	^mtCAC{2}
5	مُنفَعِله	mnCCCh	m**o**n**C**a**C**e**C**eh	montaDereh	5	N, Adj	^mnC{3}h
6	مُنفعِل	mnCCC	m**o**n**C**a**C**e**C**	montaDer	85	N, Adj	^mnC{3}
7	مُفاعِلَت	mCACCt	m**o**CAC**e**C**a**t	moSArekat	75	N	^mCAC{2}t
8	مُفاعِله	mCACCh	m**o**CAC**e**Ceh	moqAbeleh	120	N	^mCAC{2}h
9	اِفتِعال	ACtCAC	A**e**Ct**e**CAC	AeqteCAd	335	N	^ACtCAC
10	مُتَفَعِّل	mtCCWC	m**o**t**a**C**a**CW**e**C	motaSakWer	107	N, Adj	^mtC{2}WC
11	تَفَعُّل	tCCWC	t**a**C**a**CW**o**C	taEajWob	250	N	^tC{2}WC
12	تَفاعُل	tCACC	t**a**CAC**o**C	tafAhom	181	N	^tCAC{2}
13	تَفعیل	tCCyC	t**a**CCyC	taSxyC	478	N	^tC{2}yC
14	اِنفِعال	AnCCAC	A**e**n**C**e**C**AC	AenfejAr	73	N	^AnC{2}AC
15	فِعالَت	CCACt	C**e**CAC**a**t	SebAhat	270	N	^C{2}ACt
16	مُستَفِع	mstCC	m**o**st**a**C**e**C	mostabed	30	Adj	^mstC{2}
17	فَواعیل	CvACyC	C**a**vACyC	navAmys	6	N_Pl	^CvACyC
18	مَفاعیل	mCACyC	m**a**CACyC	maCAdyq	9	N_Pl	^mCACyC
19	فَواعِل	CvACC	C**a**vAC**e**C	mavAred	60	N_Pl	^CvAC{2}
20	فَعایِل	CCAyC	C**a**CAy**e**C	jazAyer	107	N_Pl	^C{2}AyC
21	فُعول	CCvC	C**o**CvC	qolvb	204	N_Pl	^C{2}vC
22	فُعَلا	CCCA	C**o**C**a**CA	SoEarA	12	N_Pl	^C{3}A
23	مَفاعِل	mCACC	maCACeC	maTAleb	168	N_Pl	^mCAC{2}
24	اَفعال	ACCAC	A**a**CCAC	AaTrAf	102	N_Pl	^AC{2}AC
25	مُفاعِل	mCACC	m**o**CAC**e**C	mosAfer	210	N, Adj	^mCAC{2}
26	مُفَعَّل	mCCWC	m**o**C**a**CW**a**C	monaDWam	315	Adj	^mC{2}WC
27	فَعّال	CCWAC	C**a**CWAC	najWAr	225	N	^C{2}WAC
28	فَعّ	CCW	C**a**CW	rabW	62	N	C{2}W
29	مَفعول	mCCvC	m**a**CCvC	maSrvT	415	N, Adj	mC{2}vC
30	فاعِل	CACC	CAC**e**C	kAseb	790	N, Adj	CAC{2}
Total					4855		

Table 2: Arabic templates and roots.

The number of occurrences in the above table reflects the number of words found in the final lexicon, not in the corpus, and it does not necessarily reflect correct detections. However, together the fully and partially diacritized items account for almost 15% of the

[32] Class shows the order of RE. The lexicon contains the class information for further investigations.
[33] Note that regular expressions can be refined to reflect a better and flawless match. For example, consonants can be specified to restrict overlaps with possible Persian matches.
[34] Arabic Consonants {b,t,c,j,H,x,d,M,r,z,s,S,C,Z,T,D,E,G,f,q,k,l,m,n,h,I,Y,U}
Persian Consonants {b,t,c,j,H,x,d,M,r,z,s,S,C,Z,T,D,E,G,f,q,k,l,m,n,h,I,Y,U,p,g,K,J}
Persian Short Vowels {a,e,o} – Persian long vowels {A,v,y} – Gemmination sign {W}

corpus. When incorporating the regular expressions in the algorithm, part of speech constraints should be accounted for; otherwise, there would be many errors because of overlaps between certain patterns. The automatic diacritization of individual words could save a lot time and labor in the process of lexicon development and although manual checking is an inevitable part of the process, it is still worth doing it.

At the time of writing this thesis, I found a study by Yoosofan et al. (2010) which attempted to do a similar work. The authors developed a pattern-matching algorithm to identify Arabic loanwords and used a 372,249 word-segment taken from the Hamshahri newspaper as a corpus. Their system identified 27% of the words in the above corpus as Arabic words. Yoosofan et al. (2010) did not develop any comprehensive system, arguing that their "findings need to be corroborated by future studies" (Yoosofan et al., 2010:1084).

With regard to Persian syllabic patterns, Samareh (1999) has done extensive research on Persian syllable restrictions. According to Samareh, Persian basic syllables, i.e., CV, CVC, and CVCC follow some syntagmatic restrictions. Only a few unique patterns but many overlaps in consonant clusters were found in this study. Furthermore, Persian semi-consonants /y, v/ are not definable as mere consonants or (long) vowels. Nevertheless, those few unique patterns listed in Samareh (1999) were used to automatically diacritize relevant words. The accuracy rate resulted from this process ranged from 50% to 75% and sometimes 75% of vowel insertions were correct within a single word. Although manual work was essential to correct wrong vowel insertions, this method saved the project a lot of time. The following example shows some Persian one-syllable words and an algorithm used to automatically diacritize them:

18)

```
'CCC >> CVCC >> CVEC >> examples ZaEf (weakness) - feEl (action) - roEb (fear)
Function farsi6(fldIn, PartOfSpeech As Variant) As String
Dim re As RegExp, Result1 As Boolean, Result2 As Boolean, Result3 As Boolean
Dim strOut As String, subStr As String
strIn = Nz(fldIn)
POS = Nz(PartOfSpeech)
Set re = New RegExp
      Result3 = Match(strIn, "^[r]E[b]")
      Result2 = Match(strIn, "^[fS]E[lr]")
      Result1 = Match(strIn, "^[ZrybvnCbsTSljq][EI][lSfsZbdmny]")

      If Result3 = True Then
            If (POS Like "N*") Then
               strOut = Left(strIn, 1) & "o" & Mid(strIn, 2, Len(strIn))
               farsi6 = strOut
            End If
      ElseIf Result2 = True Then
            If (POS Like "N*") Then
               strOut = Left(strIn, 1) & "e" & Mid(strIn, 2, Len(strIn))
               farsi6 = strOut
            End If
      ElseIf Result1 = True Then
            If (POS Like "N*") Then
               strOut = Left(strIn, 1) & "a" & Mid(strIn, 2, Len(strIn))
               farsi6 = strOut
            End If
      End If
End Function
```

The above example shows a basic Persian CVCC syllable in which the second consonant is a constant letter /E/. This syllable makes three different patterns with short vowels. These syllable structure restrictions can be encoded as rule-based automatic diacritization modules, to help in the development of the lexicon. However, manual checking is necessary for controlling the quality of the generated data. Twelve syllabic patterns were developed for the automatic diacritization of Persian words. Almost half of the preliminary diacritization task was done automatically, with the least amount of manual checking by coding phonotactical rules adapted from the above-mentioned Arabic root-and-templates and Persian syllabic patterns. This is a significant reduction of time and labor in developing large amounts of language resources.

Most of the algorithms in the above tools were developed in Microsoft Visual Basics for Applications (VBA). Relational databases like Access are Unicode friendly and fully compatible with XML. All developed lexicons can be exported easily to XML format and the latest linguistically-recommended standards such as Relax NG compact syntax. This standard has many advantages including validation (which flashes out many typing errors), flexibility (our single core dictionary could serve multiple projects), and dependability (long-term archiving). XML-based lexicons can easily be down-translated into a LEXC module by an XSLT[35] parser (Beesley & Karttunen, 2003 & 2008).

Finally, by developing root and affix lexicons, the morphological analyzer was enabled to recognize and generate millions of words. Imagine that just a verb root in Persian can be conjugated to at least 25 different forms. The proof of this claim will be shown when a 90% coverage of the morphological analyzer on the test corpus and on a completely unknown corpus (Bijankhan) is presented.

In the next sections, extraction of morphotatic rules from a full-form wordlist is explained and then the methodology employed to develop the morphological analyzer and generator will be discussed.

3.4 Persian Morphotactic Rules

The most important factor in designing morphotactic rules is how to define the root or stem in a language. In this research, as explained above, root lexicons were extracted from the training text corpus through semi-automatic processes. Using an embedded regular

[35] **XSLT,** which stands for e**X**tensible **S**tylesheet **L**anguage **T**ransformation, is a scripting language that can be used to translate or down-translate an XML file into another structured (XML or HTML) or non-structured text format file like LEXC source code (http://www.w3schools.com/xsl/).

expressions' library[36] in VBA[37] under Microsoft Access, a few modules were programmed to strip frequent inflectional morphemes like /hA/ from lexical items. However, striping most of the derivational and inflectional morphemes is only possible manually. Even using semi-automatic processes requires professional quality control before publishing the list of base forms or roots.

A list of morphotactic rules and affixes gradually were created by manually checking each root (see figures 10 to 12). The main full-form lexicon contains a list of 40,000 words extracted out of the 890,000 words of the training corpus. A randomly chosen test corpus with almost 110,000 lexical items was created for the final evaluation.

Not all the potential affixes and associated rules can be found in a small text corpus. Therefore, a list of additional affixes consistent with the roots and rules were obtained from the literature. A total of 25,000 roots and 150 affixes were extracted for all part of speech categories. The results show a perfect coverage on the test corpus and very good coverage on the sizable Bijankhan text corpus[38]. The coverage results will be presented in the last chapter.

In order to get an accurate part of speech tagger, while it was not the main purpose of this research, the grammars were designed in a modular way. Therefore, each category has its own list of affixes and rules. For nouns, adjectives and adverbs, the derivational morphemes contain 80 suffixes with adjective, adverb and noun functionalities. Inflectional morphemes form about 20 suffixes with different combinations. There are about 50 more prefixes and suffixes covering other categories such as verbs, numbers, etc. About 35 constraints were defined as "flag diacritics" for main categories in order to stop over-

[36] Microsoft VBScript Regular Expressions, Version 5.5
[37] Visual Basics for Applications, Version 6.5, Microsoft Corporation
[38] Bijankhan corpus contains about 2,500,000 words.

generation and over-analysis. Most of the restrictions are coded for derivational morphemes, because inflectional morphemes are more systematic and productive (Manning & Schutze, 2001:83). This feature unification functionality has created a deterministic dynamic for the non-deterministic finite state network. The following example illustrates over-generation and an incorrect analysis in example 19. Example 20 shows the result if flag restrictions and the correct analysis and generation are not used:

19)

Token: dysk (disk)
Existing roots:
a) /dys/ N-Sg (a plate)
b) /dysk/ N-Sg (disk)
Rule c):
N-Sg + /k/ [ak]

Analysis/generation:
1) /dys-ak/* Noun-Sg
2) /dysk/ Noun-Sg

20)

Token: dysk (disk)
Enhanced roots:
a) /dys/**@U.AK.Absent@** N-Sg (a plate)
b) /dysk/ N-Sg (disk)
Rule c):
N-Sg + @U.AK.Present@/k/[ak]

Analysis/generation:
/dysk/ Noun-Sg

The token in 19 has two analyses with the rule in 19c, because there are two roots in the lexicon, 19a and 19b. However, 19a does not allow the /k/ suffix; so, the second analysis

can be excluded by defining this constraint for the lexical root in 19a and enhancing the rule in 20c to allow the suffix /k/ when the flag "AK" is set to "Present". XFST allows as many unification features as possible. In practice, no more than seven flag restrictions were used for a single suffix. In the above example, feature unifications are used in roots and suffixes, but they might be used in prefixes and suffixes to constrain the affixation process.

3.5 Finite State Automata, Networks and Tools

Finite State Automata (FSA) are mathematical tools that play a significant role in computational linguistics. "Variations of automata such as finite state transducers, Hidden Markov Models and N-gram grammars are important components of the speech recognition and synthesis, spell-checking, and information extraction applications" (Jurafsky & Martin, 2000:22).

Finite state computing implements regular expressions, uses a series of networks or algorithmic tools, and combines them into a fully featured language processing system. These tools may include a tokenizer/normalizer, morphological analyzer/generator, semantic disambiguator, etc. (Beesley & Karttunen, 2003).

FSA can simply be defined as a formalism that is able to show a regular language in a "finite" number of "states" or nodes connected by transitional links or arcs. Arcs represent the alphabet of a regular language. Each FSA has a start and a final or accepting state in which a regular string can be validated as "accepted" or "rejected". Transitions can represent infinite number of iterations in a circular way.

The following figure illustrates simple regular languages {“ab”} and {“a”, “ab”, “abb”,...} with their encoded FSA networks adapted from Beesley & Karttunen, 2003:

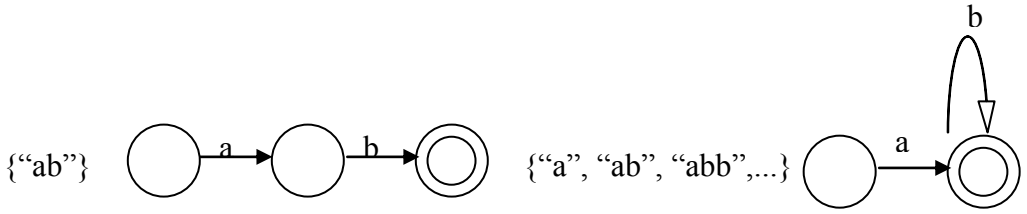


Figure 5: A simple Finite State Automata

FSA is also generative in the sense that it can generate all possible strings of a formal language. Formal or regular languages encoded in FSA are able to represent smaller parts of natural languages, like phonology or morphology. It is also very convenient to encode a lexicon of words and morphemes of a language in FSA. The following network illustrates a Persian lexical item in a FSA:

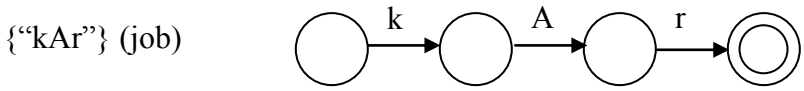


Figure 6: FSA for the word “kAr” (job)

This simple network is not only able to represent a lexical item, but also is able to recognize it as a valid lexical item. This FSA allows two Persian words /kAr/ (job) & /kAr-hA/ (jobs). The [ε] arc is encoded as a bypass or empty transition.

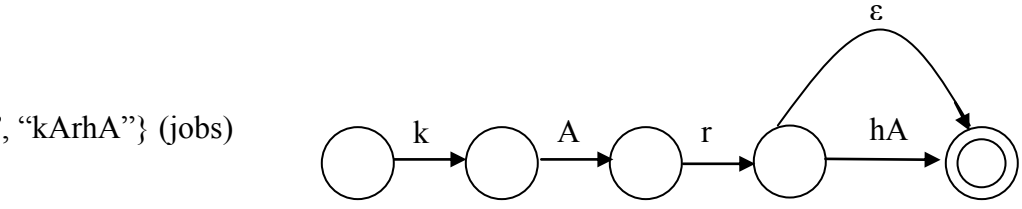


Figure 7: The FSA for Persian words /kAr/ (job) & /kAr-hA/ (jobs).

More morphemes can be added as suffixes and can be allowed at the end of this network. For instance, the Persian inflectional plural morpheme /hA/ can be added to a noun root. This morphotactic rule can be encoded in FSA however having an alternative transition path makes the network non-deterministic, allowing for both strings to be accepted by the FSA. To make FSA more efficient, Koskenniemi (1983) proposed a two-level morphology framework in which a word can be represented as a relation between its lexical and surface forms.

The lexical form concatenates the morphemes and the grammatical category of a word in the upper level, while the lower or surface level would be the actual spelling of the word. These two levels are in a regular relationship in a parallel design. In this design, morphological analysis and generation is done by using morphotactic rules that map the surface to its lexical form. The resulting automaton is called a finite state transducer or FST (Jurafsky & Martin, 2000).

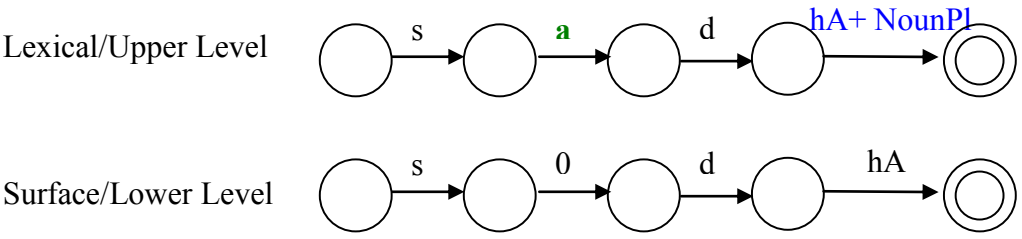


Figure 8: The two-level morphology design representing a finite state transducer or FST. Adapted from Jurafsky & Martin (2000) for the Persian example.

This design is more efficient than a single network and can handle analysis and generation in just one pass. Feasible pairs are defined to show the pairs of string symbols related in the two levels. In this case, the plural Persian morpheme /hA/ is paired with its part of speech [+NounPl]. If Σ is defined as the set of string symbols of the FST in the above figure then Σ ={k, A, r, hA:+NounPl}.

The right hand side of this regular relationship is the lexical form, and the left hand side is the surface form of the plural morpheme /hA/. Any phonological alternation can be simply defined as phoneme insertion and deletion in the lexical form of the morphemes, which is the case in this research. We can insert missing Persian diacritics (short vowels) in words and morphemes in the lexical level and retrieve them in the surface simultaneously with the morphological analysis. The following figure illustrates a simple transducer that generates a diacritized output /s**a**d/ (dam) from an accepted Persian input /sd/.

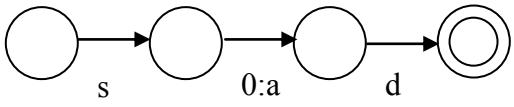


Figure 9: A simple transducer that generates diacritized form of an accepted input.

What we need to have then is a lexicon consisting of stems or roots and associated affixes, as well as morphotactic rules to be encoded as regular relations between the two levels. The following FST network illustration represents a possible lexicon:

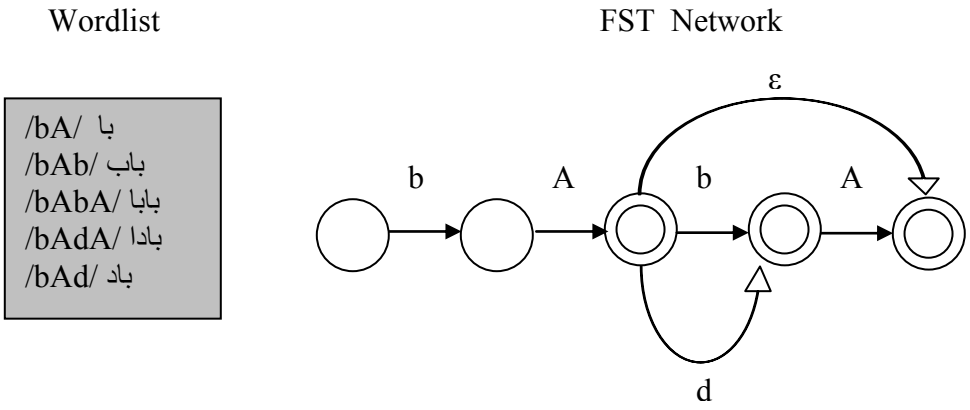


Figure 10: A wordlist that is represented by a network transducer.

In the next section, the Xerox implementation of the two-level morphological analysis and generation as well as the way this design is used to implement Persian morphological analyzer and generator will be explained.

3.6 Xerox Finite State Tools

Xerox has developed an integrated set of software tools for creating finite state networks. One of their tools is called XFST, an interface providing access to the finite state networks and algorithms that compile an extended version of regular expressions (Beesley & Karttunen, 2003). XFST can virtually encode any complicated regular expression; therefore, it is a very sophisticated tool to compile phonological alternation rules. The Xerox finite state application package is language independent and contains several other tools including[39]:

TOKENIZE: a tool that compiles regular expressions and tokenize a running text

LEXC: a lexicon to FST compiler, this is the core engine of the Xerox morphological analyzer/generator

LOOKUP: an interface that can combine FSTs and applies them to input tokens

XFST has been able to integrate the phonological/orthographical alternation rules[40] and the morphotactic rules into a full-featured morphological analyzer and generator by utilizing the above tools efficiently. Since FSA and FST can encode and integrate regular expressions and relations, mathematical functions including iteration, concatenation, complementation, composition, union, intersection, subtraction, substitution, cross product,

[39] I could not get Xerox disambiguator and shallow parser because of licensing restrictions.
[40] It should be mentioned that the current system is going to generate diacritics and as such, it does not change the orthographical representation of words. Phonological alternation rules can be easily developed in later stages of analysis, if required, for example by a TTS grapheme-to-phoneme pre-processor.

inversion, and reversion make these networks even more efficient. The Xerox finite state tools have made it possible to compose several FSTs into powerful morphological analyzers/generators. They can easily deal with concatenative and even non-concatenative languages like Persian and Arabic. The main advantage of these tools is their robustness. They use little amount of memory and allow a potentially large lexical coverage. "Compared to the alternatives, applications based on the finite state networks are usually smaller, faster, more elegant, and easier to maintain and modify" (Beesley & Karttunen, 2003:1).

As seen in the previous section on Persian morphology, Persian has a concatenative morphology, as well as compounding. The system uses about 25,000 Persian base forms/roots and 150 affixes, and it just takes less than 400 Kbytes of memory; yet, it can potentially analyze several millions of words.

Many NLP applications, including machine translation, spell and grammar checkers, text-to-speech synthesizers, etc. use a morphological analyzer and generator in their core processing engine. Developing a morphological analyzer and generator is time-intensive and laborious, however, and takes years if access to language resources and lexicons is limited.

After this short introduction to the implementation of morphological analyzer and generator, several strategies in the implementation of the three main stages of the morphological processing algorithm will be explained.

4 Implementation and Evaluation

Tokenization and normalization, core morphological processing and post-processing models are formulated strategies for the development of this research. The implementation of the diacritizer will be discussed within the two stages of analysis and then the results of the developed morphological analyzer and generator on the test corpus will be presented.

4.1 Pre-processing Module

In this stage, the test corpus was tokenized. The result was piped down to the next stage for analysis and generation. After a token was defined, the XFST language would be capable of formulating it in pattern matching regular expressions.

4.1.1 Tokenizer

The main strategy in developing a tokenizer for Persian was to formulate the space as a word boundary character and to integrate a list of dynamic multi-units to override the defined word boundary. Digits, punctuation signs and abbreviations could be also defined in this module. The following XFST code is the beginning section of the tokenizer:

```
# ======================
# Persian Finite-State Tokenizer
# Author: Peyman Nojoumian
# Created: 10-10-2010
# Updated: 03-02-2011
# ========================
# Usage: xfst -f FarsiTokenizer.fst
# ========================
clear stack
define SP " ";
define TAB "\t";
define NL "\n";
define CR "\r";
define WS [SP|NL|TAB];
define PUNCT [%.|%"|%,|%;|%:|%!|%?|%(%(|%)%)|%(|%)|%[|%]|%{|%}|%.%.%.];
define Char \[WS|PUNCT];
define WORD [Char]+;
```

```
define Digit [%0|1|2|3|4|5|6|7|8|9];
define NumSep [%.|%/];
define NUM [Digit|NumSep]+ & $[Digit];.
#==============================.
read text < MWEList.txt
define MWE
#==============================
define Token [WORD|NUM|PUNCT];
define TOK1 [Token @-> ... NL];
#=========================
define WS1 [WS]+ & $[NL];
define WS2 [TAB|SP];
define TOK2 [WS1 @-> NL] .o. [WS2 @-> SP];
#================================
read regex [TOK1 .o. TOK2];
invert net
save stack FarsiTokenizer.fst
```

The first few lines of the code simply defines control characters: space (SP), tab (\t), newline (\n), and carriage return (\r) as white space (WS) and punctuation marks as PUNCT then it defines a character to be everything except white space and punctuations. With this definition, a word is one or more occurrences of a character. The rest of the tokenizer code reads the dynamic multi-units. Therefore, a Token is defined as a word, number, punctuation mark or a multi word and a new line will be put after each longest match.

This simple and robust tokenizer is able to tokenize any Persian running text because its list of multi-units is expandable[41]. The only precaution here is to avoid morpheme overlaps, by defining and examining multi-units carefully. For example, if we define the word “bA hm” (together) as a multi-unit “bA-hm”, then the word “bA hmh” (with-all) is also tokenized as two wrong tokens “bA-hm” and a single meaningless “h”. Therefore, both multi-units should be predicted when the tokenizer is trained to avoid dangerous morpheme

[41] The XFST compiler is slow in processing MTUs because of “longest match” algorithm. However, Xerox has introduced a new FST compiler claimed to have no problems in dealing with big lists of multi-units. At the time of implementation of this project, no educational license was available to get the new FST compiler.

overlaps. The following figure illustrates the tokenized version of the Persian sentence in the example:

21)

هند ۸۰ درصد از نفت مصرفی خود را وارد می کند و «ایران»، دومین تأمین کننده نفت خام این کشور است[۴۲].

hnd, 80 drCd Az nft mCrfy xvd rA vArd my knd v ((AyrAn)) dvmyn tImyn knndh nft xAm Ayn kSvr Ast.

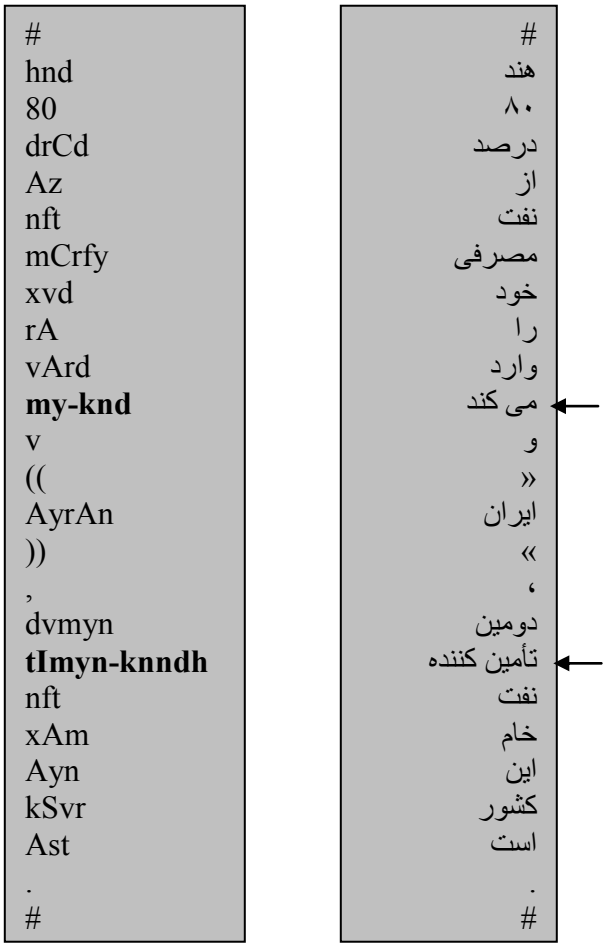


Figure 11: A Persian tokenized sentence. The left column is transliterated version of the words.

The first line is the original Persian sentence from the test corpus, which has two multi-unit words underlined for illustration. The second line is the Romanization version of the

[42] India-80-percent-of-oil-needs-its-OM-import-does-and-((Iran))-second-provide-doer-oil-raw-this country-is. (India imports 80% of its oil needs and Iran is the second raw oil provider to this country.)

sentence and the column shows tokenized items of this sample sentence. The boldface tokens are recognized as multi-units and the space is replaced with a dash. The double parenthesis, the comma and the full stop are punctuation marks and tokenized accordingly. The result is ready to be piped down to the core morphological analyzer and generator as described in the next chapter.

4.2 Core Processing Modules

In this section, lexicon and grammars developed under XFST are going to be discussed. The morphological process is done in a modular way in order to avoid many morphological overlaps[43], an orthographic feature of the Persian language and its missing short vowels.

4.2.1 Morphological Analyzer & Generator

All the following lexical and morphotactic rules have been extracted from the training corpus manually and are tested on the test corpus. The presented results will show coverage and accuracy rates for individual tokens with presumed diacritization of high frequency heterophonic homographs. No syntactic parser has been developed to insert Ezafe (short vowel "e") within the noun phrase constituents and disambiguate heterophonic homographs in the context of a sentence. However, a number of strategies are proposed for the insertion of Ezafe and disambiguation of homographs. These strategies have been tested on limited output data from the test corpus to pave the way for the future development of a parser.

[43] See example (8) under section 2.2.2.2

4.2.1.1 Heterophonic Homographs

Homograph lexicon is the first lexicon that examines the tokens. A total of 650 lexical heterophonic homographs have been listed in this lexicon. This is the only lexicon which contains full-forms and no affixation has been encoded in it; however, each entry has alternative diacritizations with correspondent frequency information coded in its part of speech tag. Most of the heterophonic homographs are binaries; meaning that one phonological realization has a higher frequency than the other does. A number of homographs with three to four alternative diacritizations have been included in this module too. The following example shows a binary heterophonic homograph as represented in the XFST lexicon:

```
!**************************************
!-Heterophonic Homograph Lexicon for Persian
!-Author: Peyman Nojoumian
!-Ver:1.0
!-Date: 02/12/2011
!-H: High frequency          L: Low frequency
!**************************************
Multichar_Symbols
        +NounHH +NounHL +PlHL ...
....
AeEmAl:AEmAl N_SING_HH;
AaEmAl:AEmAl N_PL_HL;
...
LEXICON N_SING_HH
+NounHH:0 #;
...
LEXICON N_PL_HL
+PlHL:0 #;
....
```

The “HH” code shows high-frequency phonetic realization of the homograph while the “HL” code indicates its low-frequency representation of the lexeme. In the above example, the surface forms are the same “AEmAl”, but the lexical forms represent two phonetic realizations: “A**e**EmAl” (function) and “A**a**EmAl” (deeds), which differ in one short vowel. The lexical and surface forms are denoted by “:” respectively. The tag is

encoded through a series of concatenative sub-lexicons. The "N_SING_HH" sub-lexicon encodes a high-frequency noun tag "+NounHH" in the lexical form and "N_PL_HL" encodes a low-frequency plural noun tag "+PlHL" in the lexical form. When the surface form or token is piped to this module, it is matched with the two lexical representations and it gets diacritized accordingly; the output contains two lexical candidates with corresponding tags, as illustrated in the following figure:

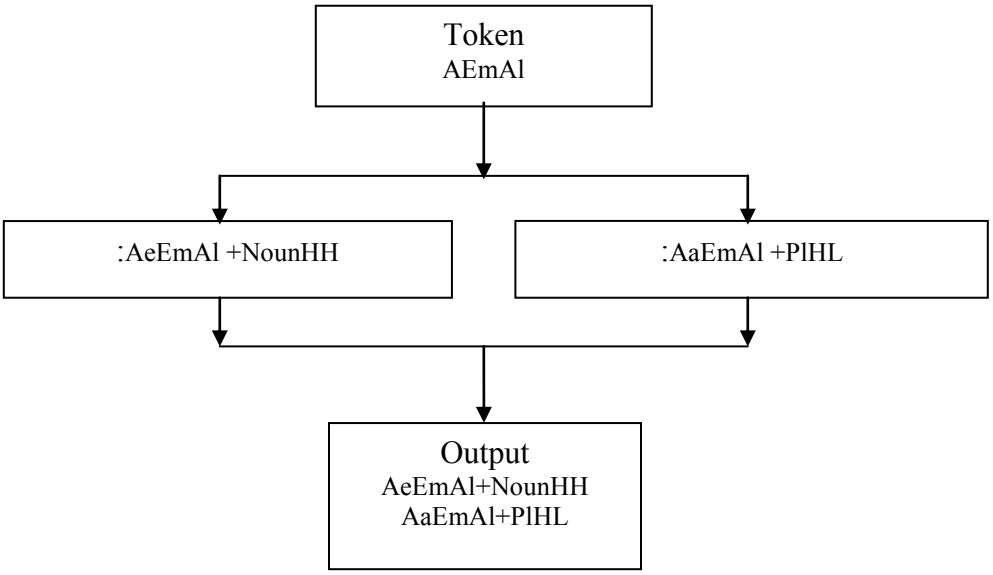


Figure 12: analysis of a single heterophonic homograph token resulting in two different analyses (candidates)

The homographs should be disambiguated later in the post-processing stage. The tags are crucial because they carry part of speech and frequency information. Only the lexical ambiguities, which resulted from the heterophonic homographs, have been accounted for, and a disambiguator model has been proposed that works based on the enriched homograph tags. Please refer to post-processing strategies for more on the disambiguator model.

4.2.1.2 Exceptions

Lexical exceptions are an inevitable part of any language. Words that are irregular, non-productive, or cause orthographical overlaps with other lexical items should be identified and stored in a well-designed module. For example, in Persian there are two-letter words that are not productive; yet, they produce a significant amount of orthographical overlap with other words and cause overgenerations. As it was argued in the Background chapter of this thesis, the mental lexicon tends to decompose multi-morphemic units that are semantically transparent. However, orthographic pairs entailing pseudo-morphemes or lexical overlaps are not decomposed (at least in Persian) and a single entry should represent each of those items in the mental lexicon. We need to do the same and have a separate lexicon for the exceptional items. The current exception lexicon contains 1200 lexical items from all grammatical categories and most of the inflectional suffixes. The morphotactic rules are customized carefully to avoid overgeneration. The following code illustrates a part of Persian Exception lexicon in XFST code:

```
!*****************************
!-Exceptions Lexicon for Persian
!-Author: Peyman Nojoumian
!-Ver:1.0
!-Date: 12/02/2010
!*****************************
Multichar_Symbols
        +Noun

LEXICON Root
Or:Or POS1; !-H-
Oz POS1; !-H-Ozemvn
Os:Os@U.ANM.Absent@ N_SING; !-OSpaz
OS:OS@U.ANM.Absent@ N_SING; !-OSpaz
Ol:Ol@U.ANM.Absent@ N_SING; !-Ol
Oh:Oh@P.POS.H@ N_SING; !-Oh
...
LEXICON POS1
+Noun:@D.POS.HA@@D.POS.HAY@ #;
+Pl:@R.POS.HA@ #;
+PlEz:@R.POS.HAY@ #;
```

4.2.1.3 Function Words, Abbreviations & Proper Nouns

The next lexicon is Function Words. It stores and encodes words with grammatical function, rather than meaning. This lexicon belongs to an almost closed set and is not very productive (except for abbreviations). However, it is very important to cover many stop words[44] and proper nouns with accurate tags to make the post-processing disambiguation process efficient. The evaluation, which is reported in the last chapter, shows that most of the words that are not diacritized belong to the proper noun category. The abbreviation lexicon contains roughly 100 items, the stop words lexicon contains about 900 words, and the proper noun lexicon contains 6,000 Persian and non-Persian (Arabic, English, French, etc.) lexical items. It is worth mentioning that a list of additional locations and proper names, as well as foreign names, will increase the coverage and accuracy rate of the system. Morphological analysis cannot help with the closed categories, because they are not productive. Nevertheless, tagging and vowel generation is still crucial in this research. This is one of the potential places for lexical expansion which will improve the coverage of a morphological generator. Comprehensive development strategies empowered this module to diacritize and tag lexical items accurately as Persian, English or Arabic proper nouns, location name, dates, etc. While the information will be used in post-processing for disambiguation and insertion of Ezafe, they are beneficial to other applications such as Name Entity Recognizers.

[44] Stop words are words with only grammatical functions. For example, prepositions such as “for” and “in” are stop words.

4.2.1.4 Numbers

The number lexicon contains 60 number names and 50 nouns and adjective that encodes compounds. Inflectional suffixes are also included in this lexicon. This lexicon enables different combination of numbers and it is potentially able to analyze them.

4.2.1.5 Predicates

The combination of nouns, adjectives and adverbs with shortened forms of the verb "st" (is) in Persian make a predicate constituent. The predicate lexicon contains 900 nouns, adjectives and adverbs that constituted predicates in the training corpus. However, it is not known if overgeneration would be a problem when including all possible nouns, adjective and adverbial roots in this lexicon. To increase the coverage of the system, this lexicon should be expanded and more roots should be added to it.

4.2.1.6 Noun, Adjective, and Adverbs

The main lexicon in the system contains almost 14,000 lexical items and 100 derivational affixes in three sets of adjective, noun and adverb makers. Inflectional morphemes include superlative, comparative, plural, and possessive suffixes and their combinatorial rules. Thirty parametric values have been defined as "flag diacritics" to encode necessary restrictions on lexical transducers.

The following sample code shows a small part of the noun lexical transducer. The tags and restrictional flags are defined in the "Multichar-Symbols" line. The flag "@U.PL.Present@" is a unification-feature that constraints pluralization process. XFST has made it possible to define combinatorial restrictions using the "flag diacritics" feature. "LEXICON Root" is the main root lexicon that contains all main lexical entries with the

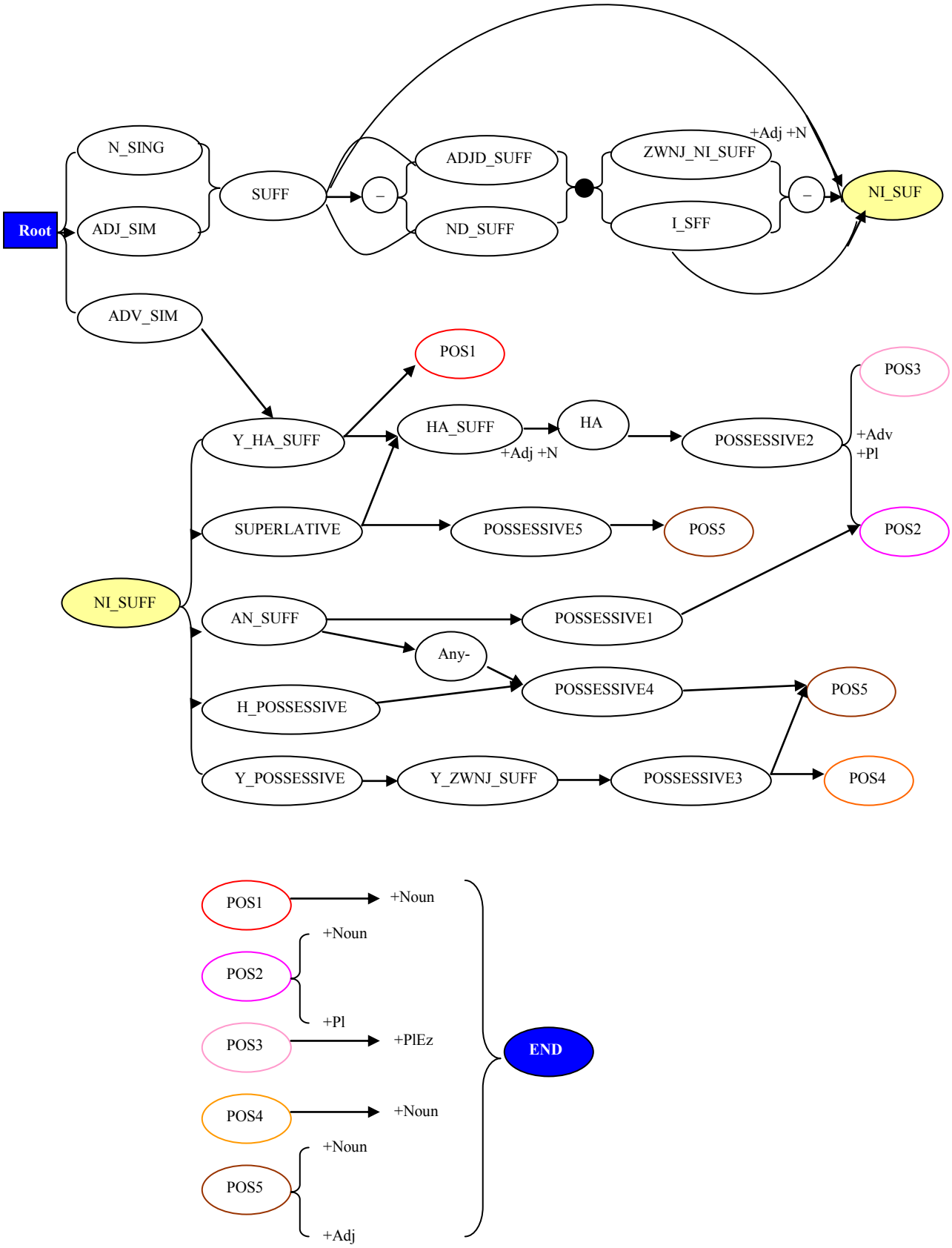
relevant grammar. “N_SING” is the first grammar which allows concatenation of suffixes on the next grammar called “SUFF”. Finally, the plural grammar “HA_SUFF” allows the plural morpheme “hA” to be analyzed and generated at the lexical level. If “ktAbhA” (books) is the input token to this lexical transducer, the output will be “ketAb-hA+Pl” (book-s+Plural).

```
!**************************************
!- Noun/Adjective/Adverb Lexicon for Persian
!- Author: Peyman Nojoumian
!- Ver:1.0
!- Date: 11/25/2010
!**************************************
Multichar_Symbols
        +Noun +Pl
        ...
!------------- Flag diacritics are defined to constrain affixations ------------------
!----------------- U: unification, Absent: No, Present: Yes-------------------------
!--------- "P.X.Y" >> P.: Preset, X: variable like POS, Y: values like ADJ ------
!--"C.X.Y" >> C: Clean --- "D.X.Y" >> D: Disallow -- "R.X.Y" R: Required--
!-----------------------------------------------------------------------------------------
@U.PL.Present@ @U.PL.Absent@ !-Plural
...
LEXICON Root
ketAb:ktAb N_SING; !-
...
LEXICON N_SING
SUFF;   ! All noun+suffixes
...
LEXICON SUFF
NI_SUFF;        !--Bypass-->> inflectional suffixes
...
LEXICON NI_SUFF
HA_SUFF;        !--OPEN--hA/+Noun/+Adj -plural suffix
....
LEXICON HA_SUFF
-hA:@U.PL.Present@hA@P.POS.HA@ POSSESSIVE2;
...
LEXICON POSSESSIVE2
yam:ym POS2;    !--"ym" may conflict verb "ym"
...
+Pl:@D.POS.A@@D.POS.ADV@ #; !--Disallow Noun ending "A"
...
LEXICON POS2
+Noun:@R.POS.A@0 #;
+Pl:@D.POS.A@0 #;
....
END
```

The compilation of this transducer shows several million potential combinations.

The following illustration shows the main lexicon and network for nouns, adjectives and adverbs representing two main layers of suffixes. The first layer contains derivational morphemes in two categories: noun makers "ND_SUFF" and adjective makers "ADJD_SUFF". Please note that "ZWNJ" is the Unicode Zero Wide None Joiner denoted by "-". It represents a dummy space between morphemes of a single word. The second layer contains inflectional morphemes "NI_SUFF" in possible combinations. The last grammars, denoted by terminal nodes, are POS tags designed to specify parts of speech as accurate as possible in four categories: noun, adjective, plural, and adverb. The arcs show bypasses in order to allow partially or non-affixed words.

Figure 13: The main lexicon network for nouns, adjectives and adverbs



4.2.1.7 Verbs

The verb lexicon has fewer roots than the previous category, but allows structures that are more complex. There are almost 1,000 verb roots in two separate present and past stem sets. Each set has defined its particular prefixes and suffixes. Each set allows 20 prefixes denoted by V_PRS and V_PA in figure 14.

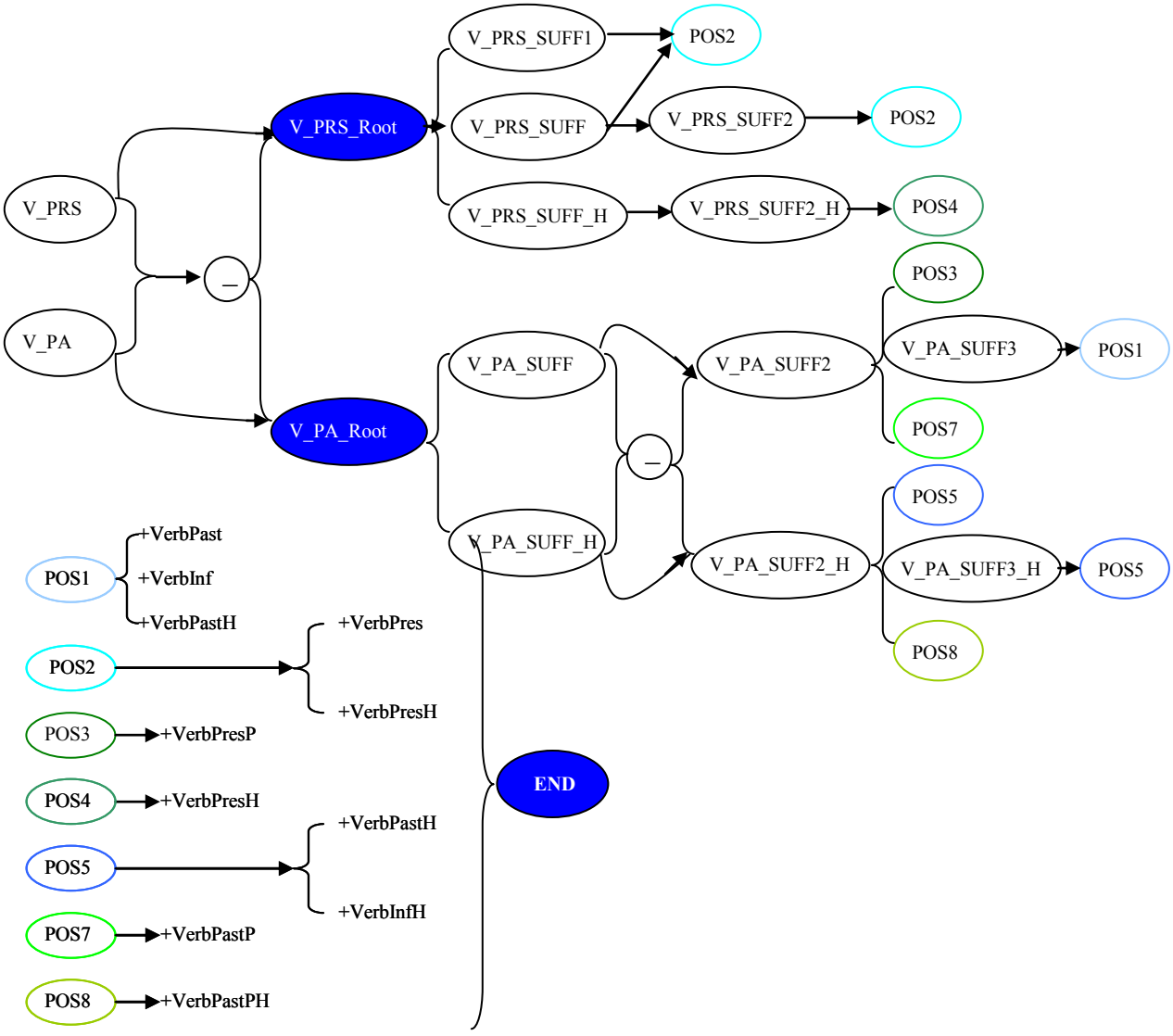


Figure 14: The main lexicon network for verbs

The present stems allow 6 x 6 subjects and object pronominal suffixes in two syntagmatic paradigms whereas the past stems allow for more than 40 suffixes in three syntagmatic or concatenative paradigms. Verbs in all possible tenses and conjugation forms, as well as infinitives, will be analyzed and diacritized here. Ambiguous homographs are specified in past and present tense for certain verbs. In any node that has an option, a defined flag, diacritics choose the allowed path. Thus, the diacritized output is always unique, unless it is a homograph. Figure 15 illustrates the designed network for Persian verbs in present, past, present & past perfect, and infinitive forms.

4.2.1.8 Compounds

Compound, complex and compo-complex words are coded in this lexicon. It contains all nouns, adjectives and adverbs in the main lexicon, as well as present and past verb stems. Suffixes are the same as for the main lexicon. Verb stems concatenate to the root after derivational morphemes. Inflectional morphemes are optionally concatenated next to allow compo-complex forms. The following figure illustrates the general compound network.

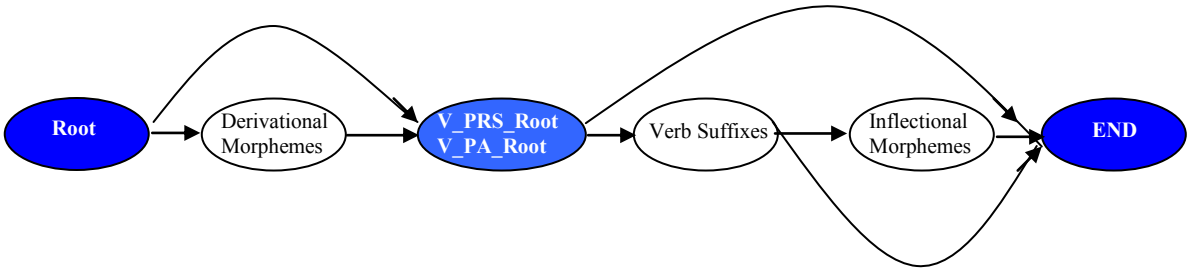


Figure 15: Compound / compo-complex network

4.2.1.9 Punctuations

Digits and punctuation marks are tagged in this module. Punctuation marks carry important contextual information, which can be used by a syntactic parser. Commas are disconnectors in Ezafe constituents. XFST provides pattern schemes as regular expressions to define punctuations and signs. Depending on the application, punctuation tags can be manipulated and defined as needed.

4.2.1.10 Unknown tokens

Even if morphotactic rules have been well defined, words might not be analyzed if their corresponding stems are not included in the lexicon (Beesley & Karttunen, 2003). Developing large databases of stem lexica is laborious; therefore, it is necessary to define a callback strategy to get unknown tokens analyzed and tagged. However, with missing stems, only partial generation/diacritization of unknown tokens would be possible. Any algorithm trying to formulate and develop syllabotactic rules might be useful to the unknown tokens. Nevertheless, it would not generate an accurate diacritization of the said tokens, especially if they are foreign words.

Beesley & Karttunen (2003) call this module a "Guesser", which works on a "phonologically possible stem". These kinds of stems are possible to be defined by regular expressions. The idea is to define a possible combination of consonants and vowels to make potentially viable stems for each category. Guessers can use the affixation rules of the other modules through a placeholder. This placeholder is a pre-defined stem put on the root lexicon of the other modules. The "^GUESSROOT" denotation in the following Exception lexicon will enable the pre-defined GUESSROOT to use the affixation rules of this lexicon.

```
!************************
!-Guesser Lexicon for Persian
!-Author: Peyman Nojoumian
!-Ver:1.0
!-Date: 12/02/2010
!************************
Multichar_Symbols
+Noun +Pl +PlEz
^GUESSROOT
!---------------------------------------
LEXICON Root
^GUESSROOT N_SING; !-guesser
bad:bd N_SING; !- a sample stem
...
LEXICON N_SING
HA_SUFF; !-the plural suffix
...
LEXICON HA_SUFF
hA:hA@P.POS.HA@ POSSESSIVE;
...
LEXICON Y_POSSESSIVE
y:y@P.POS.Y@ POSSESSIVE;
...
LEXICON POSSESSIVE
emAn:@D.POS.HA@mAn POS1;
...
LEXICON POS1
+Noun:@D.POS.HA@ #;
+Pl:@R.POS.HA@ #;
...
END
```

The Guesser strategy is defined through another module:

```
!************************************
! Callback Strategy for unknown Persian Nouns
! Author: Peyman Nojoumian
! Date: 02/02/2011
! Version: 1.0
! Usage: xfst -f [this file]
!************************************
clear stack
!- defined Persian vowels and consonants
define Vowel [a|e|o|A];
define Semi [y|v];
define Cons [O|b|p|t|c|j|K|H|x|d|M|r|z|J|s|S|C|Z|T|D|E|G|f|q|k|g|l|m|n|h|W|I|L|N|P|R|U|Y];

!- defined shortest possible noun syllabus as a Noun root CV
define Syllable1 Cons Vowel;
...
!- read the Noun root lexicon - or a lexicon with rules only
read lexc < EXCEPT.txt

!- replace the placeholder mulitchar-symbol in EXCEPT
```

```
substitute defined PossStems for "^GUESSROOT"
define AllInclusive
!- if there were alternation rules, they would be applied
!- to the lower side of AllInclusive

!- extract Attested.fst and Gusser.fst
read regex ~$["+Guess"] .o. AllInclusive;
save stack 13ATTEST.fst

read regex $["+Guess"] .o. AllInclusive;
save stack 12GUESSER.fst
```

The output of this module will be a guesser transducer able to strip the given affixes in the Exception lexicon off the pre-defined stem pattern and tag them accordingly.

As the result of the diacritizer will be discussed in the next section, it should be mentioned that no parser is developed in this study. However, a model will be suggested for resolving lexical ambiguity and the insertion of Ezafe in the post-processing module. The model has been manually tested on a few tokens from the testing corpus to demonstrate its feasibility. The interim results contain analysis and diacritization of each token plus its part of speech tag.

4.3 Results: Coverage & Accuracy Rates

The test corpus was tokenized using the developed tokenizer and piped down to the lookup strategy. Table 3 shows an example of a long tokenized and diacritized sentence which was processed using the lookup strategy. There are 31 tokens in this sentence. The first column shows correctly tokenized multi-words “yvm-Allh” and “Oyt-Allh” and punctuation marks such as comma, which plays an important role in defining NP boundaries and preventing a parser from inserting Ezafe incorrectly. The second column shows all correctly diacritized tokens in this sentence without any over-generation, and the POS column shows correct part-of-speech tags for all tokens. POS tags that are generated by the analyzer can be modified based on the output requirements. The last column shows missing Ezafe, which is going to be inserted by a modeled parser in the post-processing module. There are no homographs in this sample.

Persian tokens	Transliterated tokens	Diacritized	Diacritized (phonemes)	POS Tags	Correct diacritics ?	Correct tag?	Ezafe place
در	dr	دَر	dar	+P	y	y	
آستانه	OstAnh	آستانِه	OstAneh	Noun	y	y	x
یوم الله	yvm-Allh	یُومُالله	yovmo-Allh	+ArabicNoun	y	y	x
۲۲	22	۲۲	22	+date	y	y	x
بهمن	bhmn	بَهمَن	bahman	+ProperNoun	y	y	
،	,	،	,	+punct	y	y	
آیت الله	Oyt-Allh	آیَتُالله	Oyato-Allh	+Noun	y	y	
هاشمی	hASmy	هاشِمی	hASemy	+ProperNoun	y	y	x
رفسنجانی	rfsnjAny	رَفسَنجانی	rafsanjAny	+Location	y	y	
،	,	،	,	+punct	y	y	
رئیس	rYys	رَئیس	raYys	+Noun	y	y	x
مجلس	mjls	مَجلِس	majles	+Noun	y	y	x
خبرگان	xbrgAn	خُبرگان	xobregAn	+Noun	y	y	x
رهبری	rhbry	رَهبَری	rahbary	+Noun	y	y	
و	v	و	va	+Conj	y	y	
رئیس	rYys	رَئیس	raYys	+Noun	y	y	x
مجمع	mjmE	مَجمَع	majmaE	+Noun	y	y	x
تشخیص	tSxyC	تَشخیص	taSxyC	+Noun	y	y	x
مصلحت	mClHt	مَصلَحَت	maClaHat	+Noun	y	y	x
نظام	nDAm	نِظام	neDAm	+Noun	y	y	
در	dr	در	dar	+P	y	y	
یک	yk	یِک	yek	+Number	y	y	
مصاحبه	mCAHbh	مُصاحِبِه	moCAHebeh	+Noun	y	y	x
اختصاصی	AxtCACy	اِختِصاصی	AexteCACy	+Noun	y	y	
به	bh	بِه	beh	+P	y	y	
سؤالات	sUAlAt	سُؤالات	soUAlAt	+Pl	y	y	x
متعدد	mtEdd	مُتِعَدِد	moteEadWed	+Adj	y	y	x
خبرنگاران	xbrngArAn	خَبَرنِگاران	xabarnegArAn	+CompPl	y	y	x
روزنامه	rvznAmh	روزنامِه	rvznAmeh	+Noun	y	y	x
جمهوری	jmhvry	جُمهوری	jomhvry	+Noun	y	y	x
اسلامی	AslAmy	اِسلامی	AeslAmy	+Noun	y	y	
پاسخ	pAsx	پاسُخ	pAsox	+Noun	y	y	
دادند	dAdnd	دادَند	dAdand	+VerbPast	y	y	
.	.	.	.	+punct	y	y	

Table 3: A sample sentence from the test corpus tokenized, diacritized and tagged

4.3.1 Coverage

The preliminary running results of the diacritizer on the test corpus show 99.5% coverage with the following statistics. The first column shows different categories used by the diacritizer. These categories were defined based on the common morphotactic rules for each data type. For example, the same rule applies to all abbreviations in the abbreviation lexicon. The second column shows number of each token in each category and its percentage in the fourth column. As it can be seen, the biggest number of categories belongs to Nouns, adjectives and adverbs then function words. Only 413 tokens were not found. The following table shows the exact number of tokens analyzed by each strategy. There are also 45 tokens that were not found but their POSs were guessed by the diacritizer based on their syllabic patterns.

Category/Strategy	Tokens	Percentage
Binary Homographs	3,455	3.31 %
Exceptions	6,213	5.94 %
Abbreviations	92	0.09%
Proper Nouns	3,285	3.14 %
Function Words	30,083	28.78%
Number Words	1,375	1.32 %
Predicates	234	0.22 %
Nouns, Adjectives, Adverbs	38,826	37.15 %
Verbs (also verb homographs)	8,062	7.71 %
Compounds	1372	1.31 %
Verb Derivative Nouns	728	0.70 %
Punctuation Marks	10,333	9.89 %
Guessed Tokens	45	0.04 %
Not Recognized	413	0.40 %
Total tokens	**104,516**	**100 %**

Table 4: The coverage result of diacritizer on tokens by category

Table 5 shows the 99.5% coverage of the diacritizer on 104,516 tokens. Please note that coverage does not show the accuracy of the system. It only shows the analysis and generation rate and there might be incorrect analyses and generations. Only 17 fully voweled tokens were found, most of which were homographs. Native-speakers put diacritics on words difficult to disambiguate, even in a biasing context. Although the test corpus was not large, the 99.5% coverage rate is promising because it shows that most of the stems were found. The application was also run on Bijankhan[45] corpus of 2,597,554 tokens. The preliminary results showed 97% coverage on this large corpus. However, the accuracy of the system on Bijankhan cannot be determined at this time since it needs huge amount of time and human resources. The Iranian national corpus "Peykareh" was not accessible either because it is still under development. "Peykareh" has 100 million words of which 10 million words have been tagged and manually checked so far.

The details of the diacritizer on the test corpus is shown in the following table:

Category	Tokens	Note
Total tokens	104,516	
Not recognized because of no data in the training corpus	413	250 proper nouns, and foreign names, a few compounds
Guessed	45	some correct POS tags but no diacritics
Fully voweled	17	some homographs
Coverage	**99.5%**	

Table 5: The coverage analysis of diacritizer

It is worth mentioning that the morphological analyzer/generator is capable of dealing with partially voweled tokens as well because of the flexibility associated with the FST design. Most of the words that are not found belong to foreign proper names. A good

[45] Bijankhan corpus is available to research at: http://ece.ut.ac.ir/dbrg/bijankhan/

diacritizer should have access to a comprehensive foreign and proper names lexicons. It is crucial for a parser to have POS tags for all the lexical items in order to put Ezafe and further disambiguate word senses.

Guessed tokens are based on a very simple strategy that encodes basic syllables and incorporates inflectional morphotactic rules. The guesser, however, is not able to diacritize the tokens but it can generate POS tags to facilitate syntactic analysis in the post-processing stage. Out of 45 guessed tokens, 15 tokens were tagged accurately. However, because most of the non-found tokens are proper names, the ability of the guesser is limited. The guesser can be well designed to tag at least proper names as "noun" which was the case for the rest of the 25 tokens in this test.

The following example shows how the guesser tries to strip the known suffix "hAye" (plural plus Ezafe) and tag the unknown token based on the defined minimal syllable structure as CC (to be diacritized as CVC). The correct diacritics for this token is "/Caf-hAye/".

22)	Cf-hAye	+PlEz+Guess

4.3.2 Accuracy

Table 6 shows the accuracy rates for homographs and other categories separately. Only 657 individual tokens were diacritized incorrectly, most of which were the alternative readings of the homographs. It should be mentioned that once the rules are extracted from a corpus, they should be tuned on the training corpus to capture overlaps and wrong analysis. The rule tuning is a hectic process and needs well-designed and modular programming techniques to ease the debugging process. In this study, additional codes were put on tags,

for example +Noun1, Noun2, etc., in order to trace the analysis and generations back easily. Otherwise, in the case of a typical FST module, which has at least 1000 lines of codes, roots and affixes would create a huge network of overlaps and overgenerations if it were not well maintained and designed. Flag diacritics and filters are a valuable asset in the FST architecture, which can save a programmer a lot of time and effort. You can define "flag diacritics" for all prefixes and suffixes in order to constrain the network path that would otherwise end up in a forest of connections.

Category	Number of generations	Wrong Diacritics	Wrong POS Tags
All homographs (verbs & binary)	3,882	446	0
other categories	101,185	211	550
Total generations	105,067	657	550
Accuracy (correctness rate)		**99.4%**	**99.5%**

Table 6: The accuracy analysis of the diacritizer[46]

While this result does not capture Ezafe insertions and a parser is still required to complete the job, it shows a robust analysis especially for the ambiguous tokens. The high accuracy rates (99.5%) for the tags are indebted to well-defined morphotactic rules. If we are able to reach a high accuracy rate at this stage, the parser will probably do its job easier because correct POS tags and accurate frequency information would help a syntactic parser in the disambiguation and Ezafe insertion process.

[46] The author and an independent evaluator who was an educated native-speaker of Persian and was trained on using the evaluation tools checked the results on the test corpus. Whenever there were discrepancies on the correctness of the data between the testers, the matter was discussed to maintain the consistency of the decisions.

A detailed analysis of the data on heterophonic homographs reveals interesting findings on this category of ambiguous words. Table 7 shows detailed results for each category of the binary homographs. As it will be shown in the study on mental lexicon, the frequency of the meaning of heterophonic homographs is an important factor in accessing their phonological representations. It was just tested to see if the frequencies of the two meanings of the heterophones play any role in an isolated unbiased context. The 88.5% accuracy rate, which reflects the accuracy rate for **all** homographs (high frequency + low frequency) captures the important lexical information in heterophonic homographs.

Heterophonic Homographs	Number of generations	Correct	Error
High frequency (HH) error	95		2.5%
High frequency (HH) correct	3,338	86%	
Low frequency (HL) error	351		9%
Low frequency (HL) correct	98	2.5%	
Total generations	3,882		
Total Accuracy (correct rates)		**88.5%**	**11.5%**

Table 7: Accuracy rates for diacritization of heterophonic homographs out-of-context, based on their high frequent phonological representations

The accuracy rate for **only** high-frequency homographs is 97.2% ([3338/(3338+95)]), which is much higher than the accuracy rate for the only low-frequency homographs (21.8%). It should be mentioned, however, that low-frequency homographs comprise only 0.43% (449/105067=0.43%) of the training corpus. Only 449 words are in this category, i.e., less than 1% of all tested words and only 11% of all homographs (449/3882=11%). Therefore, the low accuracy rate on this type of homographs would not have a big effect on the systems' performance. Nevertheless, a perfect diacritizer should be able to handle low-frequency homographs as well. This is where the context can play an important role.

Contextual clues may increase the above overall accuracy results even more. In the next sections, possible improvements to these results will be discussed by looking at contextual clues based on Yarowsky (1995).

4.3.3 Discussion

The 99.5% accuracy rates for all unambiguous tokens seems to be very promising and shows how powerful a corpus-based and rule-based system can perform. A closer investigation on errors reveals that only a few proper nouns and foreign names were not diacritized at all because they were not included in the lexicon. The same problem happens in the real world when a native-speaker faces an unknown proper noun or foreign name. One solution to this problem is to improve the proper noun lexicon by adding as many nouns as possible. The other solution is to define a guesser algorithm for each category of words separately. Guesser algorithms can be very powerful analyzers if defined well. Basic syllabotactic features are sometimes very useful and able to detect the internal phonological structure of the words. A guesser can easily incorporate syllable information into morphotactic rules and perform well. The current diacritizer uses a general guesser designed for a general category of nouns. However, a more comprehensive guesser can significantly increase the productivity of the system in dealing with unknown tokens.

The incorporation of frequency information into binary homographs, inspired by psycholinguistics research, produced results with at least an acceptable accuracy rate (97.2%) and should be considered an important finding in Persian NLP.

In the next chapter, I will discuss the path to the future development of this project. More emphasis will be put on the disambiguation of homographs since they play an

important role in increasing the accuracy rating and performance of a diacritizer. Psycholinguistic evidence will also be studied to find a solution for the disambiguation of binary heterophonic homographs.

5 The Path to Future Development

The diacritizer is able to fulfill its duty in the word level. However, a full diacritizer still need to insert Ezafe and choose the correct candidate for ambiguous homographs so a syntactic parser has to be developed. This is part of the third stage of processing or post-processing.

In the post-processing stage, I am going to propose a simple syntactic parsing model for the insertion and discovery of Ezafe and for the disambiguation of binary heterophonic homographs in Persian. I did not intend to go this far, but I tried to suggest two models and I tested them on a few examples from the test corpus in order to show a possible direction for the future implementation of the proposed algorithms.

In the next section, different accounts for Ezafe in Persian are discussed when relevant, and a model parser is proposed for the automatic insertion or omission of Ezafe, based on a preferred framework. Then, disambiguation of heterophonic homographs is accounted for, by proposing another model.

5.1 Insertion and Omission of Ezafe

Elements of a Persian noun phrase are functionally linked together by the unstressed short vowel /e/ named Ezafe. According to Ghomeshi (1997) and Samiian (1983), the enclitic Ezafe is a phoneme with a grammatical linking function, whereas Kahnemuyipour (2000) and Moinzadeh (2001) treat Ezafe as the head of a Maximal Projection (EzP) with a [+N] feature complement, realized phonetically as an unstressed short vowel /e/. I will discuss some of the recent accounts and analyses on Ezafe when it is relevant to this

research. The insertion and possible over-generation of Ezafe in context, and possible issues associated with the deletion process, should be dealt with.

In the following example, a complex noun phrase shows the absence of the enclitic in the orthography.

23)

(a) پسر بزرگ پادشاه کشور انگلیس

(b)	/psr	bzrg	pAdSAh	kSvr	anglys/
(c)	/pesar-**e**	bozorg-**e**	pAdeSAh-**e**	KeSvar-**e**	aengelys/
	N-Sg +/e/	Adj +/e/	N-Sg +/e/	N-Sg+/e/	N-Sg

(The elder son of the British king)

According to this example, Ezafe can be put between two or more nouns and adjectives, but what if the last word is a predicate or an NP constituent of a compound verb or it comes after a pause indicated by a comma in the sentence? To this extent, the appearance of Ezafe is unpredictable without a proper syntactic analysis. The following example illustrates the insertion of Ezafe within the bigger constituents in a sentence.

24)

(a) هم پیمانان خاتمی، رییس جمهور تبرئه شدند.

(b)	/hm-pymAnAn	xAtmY,	ryys	jmhvr	tbrIh	Sdnd./
(c)	/ham-peymAnAn-**e**	xAtamY,	rayys-**e**	jomhvr,	tabraIeh	Sodand/
	N-Pl+/**e**/	N-Pn ,	N-Sg+/**e**/+N-Sg,		N-Sg	V-Pl

(The President Khatami's alliances were exonerated.)

In this example, Ezafe is not inserted between the group of nouns and adjectives as usual. The first noun phrase is distinguishable according to the comma separator in the text. However, the second group of nouns cannot be classified correctly without a parser. There will be no Ezafe in this case, because the third noun /tabraIeh/ is the predicate of the stative verb /Sodand/ and is not linked with Ezafe to the preceding noun /jomhvr/.

5.1.1 Syntactic Relaxation Strategies

To reach an acceptable accuracy in the prediction of the enclitic Ezafe, a model, which can be embedded into a parser, will be proposed. Simply based on a sequence of POS tags, Ezafe can be predicted in the majority of cases. The model proposes that by developing a few simple strategies, a rule-based parser might be able to insert Ezafe within the constituents of a noun phrase correctly and omit the wrongly inserted Ezafe between the constituents.

The output of the morphological analyzer put Ezafe on 6% of the cases on individual lexical items whose word boundary is closed by affixes in a way that Ezafe can already be realized. An example would be the frequent plural suffix /hA/ in which an orthographic realization of Ezafe on the basis of the semi vowel "/y/" can occur /ktAb-hAy**e**/". This Ezafe can easily be captured and the default insertion in its POS tag such as "+NounPlEz" can be reflected. Once writing the parser algorithm, the "enriched tags" will be taken into consideration. It should be also mentioned that inserting Ezafe on the basis of POS tag sequences requires very accurate generation of POS tags in the morphological analyzer or diacritizer and tuning the proposed rules on a large corpus. Comparison of different methodologies such as inductive and statistical with a rule-based system would shed more light on the efficiency of each system or a hybrid system based on both techniques.

At this stage, some general rules that were captured during the corpus development and processing are proposed.

5.1.1.1 The General Ezafe Insertion Rule

The general rule suggests the insertion of Ezafe between a head and its modifier. The heads are singular and plural nouns[47] (except the ones, which already carry Ezafe as reflected in its POS such as PlEz), adjectives, pronouns, numbers, proper nouns, determiners and certain kinds of adverbs.

Rule 1:| | → |**e**| / {Noun, Adj, date ... } _ {Noun, Adj, Pronoun, location ...}

25) /psr xvb/ پسر خوب
/pesar-**e** xvb/
Noun +/**e**/+ Adj
(good boy)

26) /klas drs/ کلاس درس
/kelAs-**e** drs/
Noun+ /**e**/ + Noun
(classroom)

The insertion rule[48] can be applied repeatedly until all constituents of a noun phrase are encountered. The problem then is that there is no clue where an NP boundary would be detected. For example, if another noun like "/dAneSgAh/ (university) follows the phrase in the above examples 25 and 26, then there is a possibility that it is linked to the last noun or adjective.

Look how this simple rule would insert Ezafe in the following example from the test corpus in table 8. Out of 20 Ezafe insertion based on Insertion Rule1, only 2 insertions are wrong. That is 90% accuracy in a single sentence. Please note that correct punctuation mark is an important factor to increase accuracy. However, in order to further increase the accuracy rate, there is a possibility to stumble upon some light syntactic parsing rules using the information contained in the POS tags to discover more accurate rules. To this extent,

[47] All noun categories: Singular & Plural, Compound, and Proper Nouns
[48] It is a tentative rule. A thorough corpus analysis should be done for more precise sequence of POS tags.

two more rules will be proposed that can eliminate wrong Ezafe insertions considering the incorporation of a noun into a verb or complementation of a noun to a predicate.

Persian tokens	Transliterated tokens	Diacritized	Diacritized (phonemes)	POS Tags	Ezafe Insertion	Correct Ezafe?	number	Ezafe place
در	dr	دَر	dar	+P				
آستانه	OstAnh	آستانِه	OstAneh	Noun	e	y		x
یوم الله	yvm-Allh	یُومُ‌الله	yovmo-Allh	+ArabicNoun	e	y		x
۲۲	22	۲۲	22	+date	e	y		x
بهمن	bhmn	بَهمَن	bahman	+ProperNoun				
،	،	،	،	+punct				
آیت الله	Oyt-Allh	آیَتُ‌الله	Oyato-Allh	**+Noun**	**e**	**n**	7	
هاشمی	hASmy	هاشِمی	hASemy	+ProperNoun	e	y		x
رفسنجانی	rfsnjAny	رَفسَنجانی	rafsanjAny	+Location				
،	،	،	،	+punct				
رئیس	rYys	رَئیس	raYys	+Noun	e	y		x
مجلس	mjls	مَجلِس	majles	+Noun	e	y		x
خبرگان	xbrgAn	خُبرِگان	xobregAn	+Noun	e	y		x
رهبری	rhbry	رَهبَری	rahbary	+Noun				
و	v	و	va	+Conj				
رئیس	rYys	رَئیس	raYys	+Noun	e	y		x
مجمع	mjmE	مَجمَع	majmaE	+Noun	e	y		x
تشخیص	tSxyC	تَشخیص	taSxyC	+Noun	e	y		x
مصلحت	mClHt	مَصلَحَت	maClaHat	+Noun	e	y		x
نظام	nDAm	نِظام	neDAm	+Noun				
در	dr	در	dar	+P				
یک	yk	یِک	yek	+Number				
مصاحبه	mCAHbh	مُصاحِبِه	moCAHebeh	+Noun	e	y		x
اختصاصی	AxtCACy	اِختِصاصی	AexteCACy	+Noun				
به	bh	بِه	beh	+P				
سؤالات	sUAlAt	سُؤالات	soUAlAt	+NounPl	e	y		x
متعدد	mtEdd	مُتِعَدِد	moteEadWed	+Adj	e	y		x
خبرنگاران	xbrngArAn	خَبَرنِگاران	xabarnegArAn	+CompPl	e	y		x
روزنامه	rvznAmh	روزنامِه	rvznAmeh	+Noun	e	y		x
جمهوری	jmhvry	جُمهوری	jomhvry	+Noun	e	y		x
اسلامی	AslAmy	اِسلامی	AeslAmy	**+Noun**	**e**	**n**	31	
پاسخ	pAsx	پاسُخ	pAsox	+Noun				
دادند	dAdnd	دادَند	dAdand	+VerbPast				
.	.	.	.	+punct				
Total					19	2		17

Table 8: A sentence from the test corpus to examine Ezafe Insertion Rule1.

5.1.1.2 Ezafe Deletion Rule 1

This rule requires the verb's POS to be checked first, in order to delete wrongly inserted Ezafe. If there is a verb or predicate at the right context of Noun2 or Adj2, then delete the Ezafe between the Noun1 and Noun2 or Adj2. This rule applies when a noun is incorporated to the predicate.

```
If      Verb/Predicate @Right context of a Noun or Adjective
        ([Noun2 | Adj2] Verb #.)

        {
                |e| → || / [Noun1] _ [Noun2 | Adj2]
        }
```

Figure 16: Ezafe Deletion Rule 1

By applying this rule, the wrongly inserted Ezafe at place 31 in the above table would be eliminated. It should be reiterated that accurate POS tags are crucial for this kind of rule to work. The application developed in this study created very accurate POS tags for verbs categorizing them based on tense, and aspect information and number of arguments as valence for different types of verbs.

It is worth mentioning that the other case of incorrectly inserted Ezafe in table 8 in position "7" could be prevented if the POS for the noun at the said position would have been refined. This noun is actually a title which occurs before the name of people like "Dr.". A POS like "+Title" would prevent Ezafe from being inserted at position 7 and could make a perfect accuracy rate. This clearly shows the need for more scrutiny of real data and test results in order to use them in refining POS tags.

5.1.1.3 Ezafe Deletion Rule 2

In order to further refine insertion rules or prevent Ezafe to be inserted wrongly in a running text, other Ezafe elimination rules can be defined. For example, the following rule defined as Ezafe Deletion Rule 2, can eliminate wrongly inserted Ezafe after pronouns or before certain Adverbial phrases.

|e| → || / _ [Adv] &&

|e| → || / {Prepositions, Pronouns, ...}_

Figure 17: Ezafe Deletion Rule 2

The efficiency of the rules proposed in this section cannot be determined before thorough testing and scrutiny of the corpus data. However, it should pave the way for the future studies in developing rule-based syntactic parsers for the Persian language. In the meantime, results from studies based on inductive and statistical methods will probably shed more light on Ezafe parsing issues. The latter methodology requires diacritized and already inserted Ezafe data in a training corpus. The diacritized version of the training corpus when containing correctly inserted Ezafe would be a good source for such systems. At the time of writing this thesis, I understood[49] that a small portion of the Persian national corpus Peykareh contains Ezafe insertions. In that case, the diacritizer developed in this thesis would be a complement to prepare large training corpora for modeling NLP applications based on inductive learning techniques.

[49] Communication with Dr. Bijankhan in April 2011.

5.2 Resolving Morphological Ambiguity

One of the main issues in NLP is resolving ambiguity or word sense disambiguation (WSD). The human lexical processor uses both syntactic and semantic lexical information and contextual clues in order to solve different types of ambiguities. In order to develop applications that are able to deal with ambiguities, we need to encode and generate as much lexical information as possible for the disambiguation process. Because of the nature of natural languages, a disambiguation model should also consider contextual clues.

To this end, the amount of lexical information that can be encoded into lexical items will be explored in order to benefit a context-enabled homograph disambiguator. Ambiguity might result from lexical, semantic or syntactic categories. In this study, lexical ambiguity in the form of heterophonic homographs will be discussed and a disambiguation model will be proposed to solve it.

It should be mentioned, however, that leaving ambiguity in the lower levels of the analysis, for example in the morphological analyzer/generator, in an NLP system is problematic because it propagates itself exponentially to higher levels of syntactic analysis later in the process. Therefore, it would be more efficient and robust for a system to deal with less ambiguity if ambiguity resolutions were possible in the lower stages of the analysis (Attia, 2008).

The morphological analyzer and generator explained in the previous chapter is able to deal with a number of ambiguities such as multi-word entries in the main processing stage. A number of homographs will surface in to the post-processing stage if they are not dealt with in the lower levels of the analysis. For example, the ambiguous homograph noun "gl" (flower/mud) in 28 is not ambiguous anymore in the collocation or compound word "/gl

frvS/" (27) if the collocation was tokenized correctly. This particular example shows the importance of the dummy space within the two words. Therefore, an early discovery of the collocation or compound in the tokenizer and later analysis by the analyzer will automatically disambiguate the token, resulting in fewer burdens on a parser at the post-processing stage. In fact, collocations can be helpful in disambiguation of certain homographs.

27)

(?) /gl frVS/ گل فروش

/gl frVS/ گل فروش

/**gol-forvS**/ گُل‌فُروش

(flower-seller)

28)

/gol/ (flower) گُل

/gl/ گل

/gel/ (mud) گِل

Different disambiguation possibilities will be explored by first looking at some psycholinguistics evidence on the disambiguation of isolated heterophonic and homophonic homographs in the mental lexicon. For the case of Persian homographs especially heterophones, the role of frequency of the dominant and subordinate meanings of the ambiguous homographs will be explored and then a model for word-sense disambiguation will be suggested based on "one sense per collocation" proposed by David Yarowsky (1993). Though psycholinguistic evidence might not be fully programmable in NLP models, it would be definitively insightful because the ultimate goal of a natural language processing application is to be able to simulate human cognitive ability in language processing.

5.2.1 Psycholinguistic Considerations

Although most words in languages are linked to multiple nuances of meaning, a certain subset of words is clearly ambiguous. That is, a single word represents more than one distinct meaning. Ambiguity can exist at both the spoken and written levels of language. Words that share identical phonological forms are termed homophones. In written languages, two words with identical orthographic forms are termed homographs. Words with identical pronunciations and spelling are called homophonic homographs. The ambiguity of a homophonic homograph derives solely from the relations between its single phonological form and the two or more semantic representations that the phonological form is linked to. These semantic representations can vary in frequency. For example, although the English word "bank" has only one pronunciation and is thus not phonologically ambiguous, it has at least two meanings. One that is always its high frequency meaning: "financial institution" and the other is its less frequent meaning: "bank of a river". However, homographs may have more than one pronunciation in languages whose writing systems do not abide by strict phoneme grapheme correspondence rules (e.g., English) or do not include representations for vowels (e.g., Hebrew and Persian). The resulting ambiguous words, termed heterophonic homographs, have multiple pronunciations, each associated with a different meaning and frequency. In heterophonic homographs, the ambiguity derives from the relations between the orthographic and multiple phonological forms of a word, as well as from its multiple semantic representations. For example, the English word "wind" is phonologically and hence semantically ambiguous since it can be pronounced either as /wind/ meaning "moving air" or as /waynd/ meaning "turn coils or springs of something like a clock" (Frost et al., 1990:569).

Thus heterophonic homographs can be characterized as having a kind of double-ambiguity compared to homophonic homographs.

The study of ambiguous words has the potential to provide insight into mechanisms for accessing multiple meanings of words in the lexicon (Gorfein, 2001). Psycholinguistic research on the disambiguation of homographs in the mental lexicon has tried to address several major questions in this regard. They include: whether both the dominant (more frequent) and the subordinate (less frequent) meanings are activated, whether access occurs in parallel or in order of relative frequencies, i.e. in serial order, and whether activation is differently affected by biasing contexts versus isolated presentation. Languages that incorporate both homophonic and heterophonic homographs provide an especially fertile testing ground for investigating these questions and for attempting to account for the relationships between the written, phonological and semantic representations of words.

The double ambiguity of heterophonic homographs may be reflected in their being processed differently from homophonic homographic. For example, Frost & Bentin (1992) report that the semantic processor seems to interact with the phonological system in the disambiguation of homographs to the extent that "the access to meaning is mediated by phonology" (Frost & Bentin, 1992:67). To date, most of the studies on lexical ambiguity have focused on homophonic homographs, as these are more common in English (see Hogaboam & Perfetti, 1975, Forster & Bednall, 1976, Holley-Wilcox & Blank, 1980, Onifer & Swinney, 1979, 1981, and Simpson & Burgess, 1985). However, languages such as Persian, Arabic, Hebrew and Serbo-Croatian utilize orthographic systems that are more prone to heterophonic homography, providing an opportunity to study both types of

ambiguities (see Frost et al., 1990 on Serbo-Croatian, Frost & Bentin, 1992 on Hebrew, and Carpenter & Daneman, 1981 on English).

Semantic priming is one of the preferred experimental techniques used to address the ambiguity associated with both homophonic and heterophonic homographs, since it permits investigation of the access level and activation time course of the different meanings of a homograph in the mental lexicon. Semantic priming studies involving homographs have generally been designed to test three different models of lexical access, although there are several points of view and somewhat different nomenclature with respect to each model. The models are the exhaustive (parallel) access model, the ordered (serial) access model and a combination of the two as well as some variations[50] of the two (Frost & Bentin, 1992:58). Both of these models are also referred to as multiple-access models in contrast with a single-access model in which only one meaning of an ambiguous homograph would be accessed in a biased context. The latter is also called a context-dependent access model (Holley-Wilcox & Blank, 1990).

The exhaustive access model states that all the meanings of an ambiguous homograph are accessed in parallel in the mental lexicon regardless of a contextual bias or the frequency of their meanings (Onifer & Swinney, 1981). However, according to Onifer & Swinney (1979), the alternative meanings of a homograph are not available to conscious introspection during this brief period of parallel access. They claim that eventually only one meaning becomes available to the listener at a conscious level.

[50] A variation to the ordered-access model is reordered-access model in which context reorders the sequence of lexical access and a variation to the exhaustive-access model is selective-access model in which all meanings are activated but the biased meaning is selected and other meanings are deactivated or so-called suppressed (Gadsby, 2008).

Holley-Wilcox and Blank (1980) used a lexical decision paradigm to examine disambiguation of homophones in isolation (e.g. the word "bank"). Their results showed facilitation to targets related to all of the meanings of the ambiguous homograph. They interpret the results of their study as evidence for the multiple-access hypothesis, suggesting that all the meanings associated with an ambiguous homograph are accessed from the mental lexicon in parallel (Holley-Wilcox & Blank, 1980).

In the version of the exhaustive access model advanced by Onifer & Swinney (1981), all of the meanings of an ambiguous word are accessible to the comprehension device as soon as it is encountered; access, which is thought to be independent of any relevant context, can happen in either a parallel or a serial manner. In either case, access is viewed as a pre-processing subroutine in the comprehension device, while contextual cues seem to operate in a post-processing routine when a candidate meaning is selected and made available to the conscious process (Onifer & Swinney, 1981). Figure 18 illustrates both versions of the exhaustive-access model:

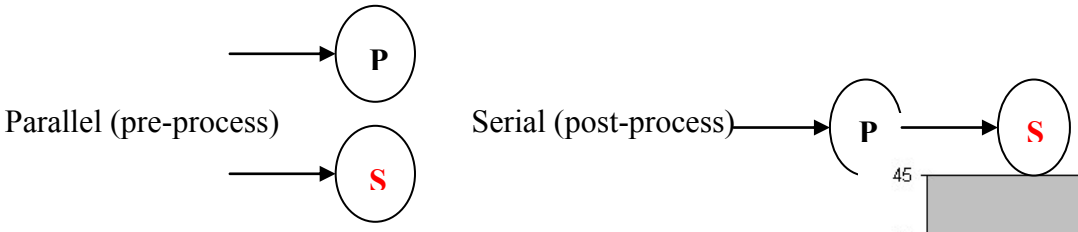
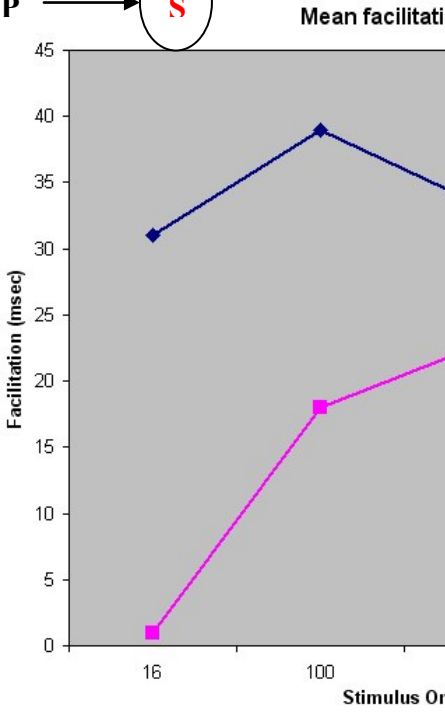


Figure 18: Exhaustive access model (**P:**Primary meaning, **S:**Secondar

In contrast to exhaustive models, the ordered access model states in ambiguous words may involve a terminating ordered search pro subsequent meanings of the ambiguous word are ordered based on their search process is terminated once it encounters a contextually related mea Perfetti, 1975).



Support for the ordered access model was provided via a sentence classification experiment, in which Hogaboam & Perfetti (1975) asked participants to listen to sentences and decide whether the last word they heard was ambiguous or not. Their experiment showed a faster detection of the ambiguity when the context was close to the second meaning of the ambiguous word. In contrast, when the more frequent meaning of the ambiguous word was detected, it was difficult for the participants to recognize that the word was ambiguous. Based on these findings, they argued that participants accessed only the more frequent meaning of an ambiguous word when the contextual bias favored it, but accessed both meanings when the context favored the secondary meaning.

However, Onifer & Swinney (1981) argued that Hogaboam and Perfetti's (1975) sentence classification task simply demonstrated a conscious selection task in which it was not clear whether both meanings of an ambiguous word had been accessed unconsciously or not. Even if both words had been accessed, it was possible that only the contextually relevant meaning was available to participants at a conscious level. It is also likely that the conscious decision reflected in their post-sentence task took place long after the initial lexical access, which is a highly automatic unconscious process. Furthermore, Onifer & Swinney (1981) argue that when the place of ambiguity is predictable, participants might be able to use specific task performance strategies to detect it. This would compromise the validity of the data reported by Hogaboam and Perfetti (1975).

Onifer & Swinney (1981) also suggested that a cross-modal lexical priming task would be a good technique for investigating the unconscious level of lexical access while avoiding participants' task performance strategies. They predicted the following results: If the ordered access model were correct, facilitation would be limited to lexical decisions made to words

relevant to the primary (more frequent) meaning of an ambiguous word in a biased context. Conversely, lexical decisions to words related to the secondary meaning of the ambiguous word should not be facilitated in the presence of such contexts (Onifer & Swinney, 1981). However, if the exhaustive access model was correct, lexical decisions made to words related to both meanings of an ambiguous word would be facilitated regardless of the biasing context.

In their first experiment on English ambiguous words[51], Onifer & Swinney (1981) asked participants to listen to a sentence (biased towards the primary[52] or secondary meaning[53] of an ambiguous word) while making lexical decisions to words (and non-words) visually primed by a related word to the primary or secondary meaning of the ambiguous word. The results showed that lexical decisions for words related to both readings of the ambiguity were facilitated regardless of the sentential bias conditions. Therefore, their results appeared to support the exhaustive access hypothesis in which all the meanings of an ambiguous homophone were accessed regardless of which meaning the context sentence highlighted.

Onifer & Swinney (1981) suggested, however, that the sentential bias might be used in a conscious process that takes effect only after the meanings of the ambiguous word have been accessed. Thus, they proposed two stages of lexical access for ambiguous words: a pre-access stage in which both frequent (dominant) and non-frequent (subordinate) meanings of an ambiguous word are accessed and activated and a post-access stage in which the dominance of the meaning is determined rapidly based on a biased context. Onifer &

[51] Homophonic homographs like *"scale"* with related meanings to "**weight**" and "fish" (Onifer & Swinney, 1981: 232).
[52] (P): The postal clerk put the package on a postal ***scale*** to see if it had enough postage.
[53] (S): The dinner guests really enjoyed the specially prepared river bass, although one guest did get a *scale* caught in his throat.

Swinney (1981) ran their second experiment to investigate the time course of activation of the various meanings of an ambiguous word in order to determine the nature of the post-access decision process. They observed that only the contextually relevant meanings of the ambiguous word showed facilitation when there was a 1500 ms delay in showing the ambiguous word. They took this facilitation as evidence for the post-access conscious process and claimed that it had nothing to do with the unconscious lexical access process[54]. Nevertheless, they noted that this time course might have been overestimated, as other studies, e.g. by Simpson (1981), had suggested that the access process should be over by 150ms from the onset of the stimulus**.**

In an unbiased context, Simpson (1981) showed that only targets related to the dominant meaning of the ambiguous word were primed and took that as evidence for an ordered lexical access model as proposed by Forster & Bednall (1976), and Hogaboam & Perfetti (1975). However, Simpson (1981) used a priming technique in which participants had to decide on the word/non-word status of both primes and targets. In addition, a lapse of 120 ms between the presentation of the prime and target in his experiment made it difficult to determine which meaning was retrieved faster, because the results possibly "arose from a decision process occurring after the initial retrieval of all meanings" (Simpson & Burgess, 1985:29).

Simpson and Burgess**'** (1985) investigation of the role of frequency attempted to take into account Onifer & Swinney's (1981) methodological criticism of Simpson (1981) and Hogaboam & Perfetti (1975). They studied the role of meaning frequency in isolated

[54] Onifer & Swinney (1981) also predicted that in an experiment with isolated ambiguous homographs (out of context and therefore unbiased), the dominant meanings might be "brought to consciousness" earlier than the subordinate meanings (page 232).

ambiguous words (homophones), in an attempt to support the ordered access hypothesis. In two lexical decision priming experiments, ambiguous primes were followed by targets that were related to the prime either through their more frequent or less frequent meanings or were unrelated to the prime (Simpson & Burgess, 1985). This approach was reported to be "probably preferable for those cases in which the meanings of the ambiguous word would not occur with equal frequencies because it would allow separate assessment of activation levels for dominant (more frequent) and subordinate (less frequent) meanings" (Simpson & Burgess, 1985:28).

Simpson & Burgess (1985) also set out to address the time course of activation of the dominant and subordinate meanings of ambiguous words in "a wider range of prime-target intervals" than those of previous studies. Given that Onfier & Swinney (1981) showed no effect for meaning frequency with only two vastly distant intervals (0 and 1500 ms), Simpson & Burgess (1985) assumed that "at some intermediate interval" they might find a role for the meaning frequency. In both their experiments, Simpson & Burgess (1985) selected different intervals between the onset of the prime and the target. They used SOAs of 16, 100, and 300 ms in their first experiment and 300, 500 and 750 ms in their second experiment, reasoning that if both dominant and subordinate meanings of the ambiguous prime are equally facilitated at the shortest SOA (relative to unrelated targets), then lexical access must be exhaustive. On the other hand, greater facilitation for dominant meaning of the ambiguous prime, even at the shortest SOA, would indicate that lexical access is ordered according to the meaning frequency (Simpson & Burgess, 1985:29).

Simpson and Burgess (1985) presented different results in the two experiments. In their first experiment, they showed that responses to targets related to the dominant meaning of

the primes were faster than the targets related to the subordinate meaning of the primes in all SOAs. Thus**,** in the absence of a biased context, when an ambiguous word was presented, its dominant meaning was available and accessed almost immediately (as early as 16 ms), whereas the subordinate meaning of the ambiguous word was activated more slowly (after almost 100 ms). This pattern indicated a tendency to move from an ordered to an exhaustive access as SOA increased from 16 ms to 300 ms**.** However, the results of the second experiment showed a tendency to move from exhaustive access to ordered access as the SOA increased from 300 ms to 750 ms (Simpson & Burgess, 1985:4). The SOA can be seen as how long it takes for the either of the meanings (dominant or subordinate) to be accessed. In the shorter SOAs such as 16 ms, the prime stimulus is still not conscious to the subject whereas longer SOAs such as 750 ms bring the stimulus to a conscious level of processing.

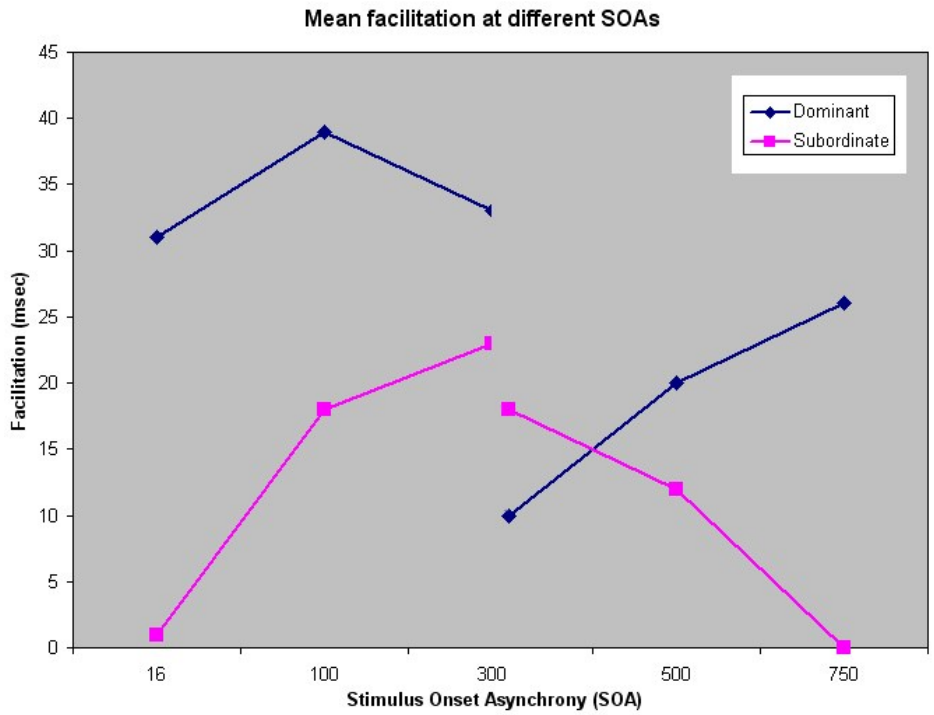


Figure 19: Simpson & Burgess (1985),

Page 32: Mean facilitation of dominant and subordinate associates at five SOAs.
(16-300-ms SOAs are from Experiment 1 and 300-750-ms SOAs are from Experiment 2).

Based on these findings, Simpson & Burgess (1985) believed that in fact no single distinction could be drawn between the exhaustive and ordered access models**.** Nonetheless, since both meanings were eventually activated, they concluded that lexical access in the processing of ambiguous words is most likely exhaustive, and to the extent that the activation rate of the meanings is dependent upon their frequencies ordered (Simpson & Burgess, 1985).

This interpretation was consistent with what Onifer & Swinney (1981) described as the pre-process (exhaustive) and post-process (ordered) access of the meanings of an ambiguous word. It was also found to be compatible with the two-process model of word recognition proposed by Neely (1977) and Stanovich & West (1979, 1981)**,** in which initial activation made all meanings of an ambiguous word available in the first stage and, in the second stage, focused attention on the dominant meaning, while inhibiting the other meanings**.** It should be noted here that MacKay (1970) was the first to argue that activation of a meaning inhibits the selection of the other meaning(s) of an ambiguous word in the later stages of word processing (Simpson & Burgess, 1985).

As was noted above, most lexical access studies of ambiguous words designed to test the exhaustive and ordered models have focused on **homophonic** homographs. In the following paragraphs, I review some of the findings reported to date for **heterophonic** homographs and discuss their implications for distinguishing between the two access models.

Frost et al. (1990) explored the effects of phonological ambiguity in Serbo-Croatian[55]. A limited number of heterophonic homographs occur in Serbo-Croatian as a result of its use of both the Roman and Cyrillic alphabets, with the same letters in the two alphabets

[55] Frost et al's language categories are no longer used in Balkan sociolinguistic discussions. There are now two different languages, Serbian and Croatian, and each uses a different alphabet.

representing different phonemes[56]. Written and spoken words or non-words were presented concurrently to participants, who were asked to decide if the two forms matched. The participants showed slower responses to ambiguous words relative to unambiguous ones. Their responses were significantly slower when the ambiguous printed words matched with the less frequent spoken alternative than with the more frequent one. These results were later interpreted in support of a multiple-access model (or a serial access model), in which the more frequent spoken alternatives of the ambiguous words activated faster relative to the less frequent ones (Frost & Bentin, 1992:59).

Frost et al. (1990) initially claimed that their results were consistent with those of Hogaboam and Perfetti (1975) to the extent that the frequency-ordered access model could be extended to their experiment on heterophonic homographs. However, their matching task was methodologically different from the sentence classification task of Hogaboam and Perfetti (1975). Hogaboam and Perfetti's (1975) results on lexical access were interpreted by Onifer & Swinney (1981) as a conscious selection process rather than an unconscious lexical access (Frost et al., 1990).

Frost et al. (1990) also compared the results of their study with Simpson nad Burgess's (1985) study on homophonic homographs. It is worth noting that **heterophonic** homographs add a second dimension of ambiguity, i.e. phonological ambiguity, to the lexical access task and might change the time course of activation. Furthermore, comparing lexical access in homophones with heterophones might elucidate the role of double ambiguity (phonological and lexical ambiguity) in the mental lexicon. Phonological ambiguity was defined by Frost et al. (1990) as "a complex, non-isomorphic connection between letters and phonemes"

[56] For example, letter "p" represents /p/ in Roman but /r/ in Cyrillic. Thus, "Potop" can be pronounced as both /potop/ meaning "flood" and /rotor/ meaning "rotor" in Serbo-Croatian.

whereas lexical ambiguity is characterized by "ambiguous connection between the orthographic and phonological system at the level of whole words" (Page 578). In a direct comparison between heterophones and homophones, Kroll and Schweickert (1978) also found that heterophones like "wind" took longer to name than homophones (Frost and Bentin, 1992). Thus, any comparison of this nature between homophones and heterophones would provide more information on the interaction of phonological and orthographic forms in an ambiguous orthography like that of Persian.

Carpenter and Daneman (1981) studied phonological ambiguity in English heterophonic homographs. They used fourteen English heterophonic homographs in noun-noun, noun-verb and verb-verb pairs like "lead" /led/ (a metal) and /li:d/ (clue), tears /terz/ (rips), and /tierz/ (droplets), close /kloz/ (shut), and /klos/ (near) and integrated them into a passage related to the heterophone's more frequent or less frequent meanings. They found that in reading a passage that primed a high-frequency meaning of a heterophonic homograph, the duration of eye fixations was shorter than when the semantic context primed the low-frequency phonological alternative of the ambiguous word (Carpenter and Daneman, 1981). Comparing this evidence with the results of previous studies**,** e.g. Frost et al. (1990) on Serbo-Croatian, one might conclude that heterophonic homographs are processed the same way in English as in Serbo-Croatian. However, in both languages "heterophonic homographs are processed differently than homophonic homographs" (Frost and Bentin, 1992:59).

Frost & Bentin (1992) examined the disambiguation process in Hebrew heterophonic and homophonic homographs in isolation, without any biasing context. They used a semantic paradigm at different SOAs to investigate any meaning activation pattern similar to that of Simpson and Burgess (1985). Participants were presented with an ambiguous prime

and asked to decide whether an unambiguous target was a word or non-word. Some of the targets were either related to the dominant (more frequent) meaning of the prime or related to the subordinate (less frequent) meaning of the prime relative to unrelated targets. Lexical decisions to targets related to the dominant meaning of the heterophones were facilitated at all SOAs, but subordinate meanings of the primes were only facilitated at SOAs of 250 ms or longer. Homophonic homograph primes, however, facilitated lexical decisions to targets at all SOAs regardless of the dominancy of the prime's meaning. These results suggested a multiple-access model in both heterophonic and homophonic homographs regardless of meaning dominancy**,** with phonological ambiguity accounting for the different time courses of activation. All meanings of homophones were activated as early as 100 ms from stimulus onset, whereas only dominant meanings of heterophones were activated at 100 ms SOA, while subordinate meanings of heterophones were delayed. These findings are consistent with the onset-pattern of meaning activation in homographs suggested by Simpson & Burgess (1985) and demonstrate that heterophonic and homophonic homographs are disambiguated differently in Hebrew. The slowed access to the less frequent meaning of heterophones, compared with that of homophones, was considered as evidence for a single entry in the mental lexicon for homophones, but several lexical entries for heterophones, since they are by definition represented by several phonological realizations (Frost & Bentin, 1992).

The "multiple-entries structure and ordered-access process" for heterophonic homographs appears to support a two-layered disambiguation process in which the phonological processor mediates between the orthographical form of the ambiguity and its meaning. If several phonological units are realized for a heterophonic homograph, in which

each phonological representation is connected to its own specific meaning, then the dominancy of the meaning frequency must be more relevant in heterophones than homophones. This is because, in homophones, only one lexical entry or phonological representation is connected to several semantic nodes (Frost & Bentin, 1992:66). Thus, all the meanings of a homophone can be accessed in parallel directly from the orthography and sent to the comprehension device. In contrast, the disambiguation process in heterophones demands more processing time because there is no direct connection between the orthographical form of the heterophone and its meanings. The connection, however, seems to be mediated by the phonological processor. Thus, heterophones in isolation are assumed to be primarily connected to their dominant meaning for initial phonological disambiguation. However, with longer SOAs or in biasing contexts, subsequent meanings of the heterophones would be available to the processor as well.

The study described below is devoted to an exploration of the interaction of phonological and semantic ambiguity in the lexical access of homographs in Persian. As is explained in the following sections, the orthography of Persian makes it uniquely suited to test the accessing of the ambiguous meanings of both heterophonic and homophonic homographs.

5.2.1.1 Heterophonic Homographs in Persian

Heterophonic homographs in Persian, like Hebrew, are words with two or more pronunciations**,** each associated with a different meaning. Although "the spelling-sound correspondences in alphabetic Persian are entirely consistent", the three short vowels /a/, /e/ and /o/, called diacritics, are not usually written in Persian orthography. Words lacking short vowels in their written form are considered to be phonologically opaque (Baluch, 1993:22)

and account for the ambiguity in heterophonic homographs. For example, the phonologically opaque[57] word "کرم" /**krm**/ in Persian can be pronounced as /k**e**rm/ (worm), /k**ere**m/ (cream), /k**ara**m/ (affection), or /k**oro**m/ (chrome) with four distinct meanings. It is worth mentioning that these kinds of heterophones are rare and most of the observed heterophones in the corpus were binary.

Although heterophonic homographs in isolation form a small part of the Persian lexicon, they are productive and are widely used. Only ten percent of this category of words has more than two alternative meanings and pronunciations. The following example shows a binary heterophonic homograph in Persian:

29)

تن /t**a**n/ (body) , Noun, dominant meaning frequency=317

تن /t**o**n/ (ton) , Noun, subordinate meaning frequency=131

Heterophonic homographs are also formed via productive morphosyntactic processes. Example (30) illustrates how a Persian homograph like /**mrdm**/, finds its heterophonic counterpart in syntax[58]:

30)

مردم /m**a**rd**o**m/ (people)

مردم /m**o**rd-**a**m/ (I died)

[57] Phonologically transparent words are written with three long vowels presented as letters "ا" /A/, "و" /v/, and "ی" /y/ in Persian orthography. For example, "بار" /bAr/ can only be pronounced as /bAr/.

[58] This creates a three-dimensional ambiguity in Persian heterophones in which the orthographic form is connected to the semantic and syntactic processors mediated by phonological forms.

5.2.1.2 Homophonic Homographs in Persian

In contrast to Persian heterophonic homographs, homophonic homographs share both their printed orthography and pronunciation, but with different meanings. For example, the Persian word "تاب" /tAb/ can only be pronounced as /tAb/, with the meanings "swing" or "tolerance". It is worth noting that borrowing has introduced some heterophonic and homophonic homographs into Persian. For instance, the homophone word "راکت" /**rAke**t/, borrowed from English, has two distinct meanings of "rocket" and "racket". Obviously, these kinds of ambiguities arise when the borrowed word is adapted to the target language's phonological system. Example (31) shows another Persian homophone with its dominant and subordinate meaning frequencies:

(31)

دفتر /daftar/ "office" (dominant freq=280), "notebook" (subordinate freq=9)

There has been relatively little research on the disambiguation of homographs in Persian. In a naming task, Baluch & Besner (1991) found that semantic priming and word frequency had significant effects on phonologically opaque and transparent Persian words in isolation (Baluch, 1992). "The naming task, however, could not disclose covert phonological selection processes" and whether or not alternative pronunciations were accessed (Frost & Bentin, 1992:59). Thus, a more robust measurement technique is required to investigate if other meanings of a homograph are activated and whether there would be any connection between the relative frequency of each meaning and the ambiguous word.

To address this issue, a semantic priming paradigm similar to the one used by Frost & Bentin (1992) and Simpson & Burgess (1985) was utilized in the present research to

examine the disambiguation of heterophonic and homophonic homographs in Persian. As mentioned earlier, this priming technique is preferable for those cases in which the meanings of an ambiguous word occur with different frequencies, because it can discern the activation levels of the more frequent meanings from the less frequent ones (Simpson & Burgess, 1985).

If the dominancy of the meaning of an ambiguous homograph proves to have a significant role in accessing its phonological realization, then using an accurate frequency for different meanings of the homograph can be useful for an automatic diacritizer. Frequency lists extracted from large corpora can easily be added to the lexicons and used efficiently by the diacritizer. However, it is worth noting that this is just one simple solution to the disambiguation of homographs. Inductive or statistical methods, for instance, might have easier solutions. Nevertheless, having accurate frequency information for different meanings of ambiguous homographs would be very useful for many NLP applications.

To this extent, a psycholinguistic study on the level of access to Persian homographs and the role of meaning frequency in determining the phonological realization of these ambiguous words was designed. One of the primary goals of this psycholinguistic study, however, was to assess the explanatory value of exhaustive access versus frequency ordered access models for the activation of ambiguous homographs. Thus, I defined my goals in this way: in the case of heterophonic homographs, if responses to targets related to the dominant meaning of the prime in a lexical decision task turned out to be faster than responses to the subordinate meaning of the prime, then I could conclude that semantic priming is affected by the frequency of the primes' meaning. On the other hand, a slower reaction time to the subordinate meaning of the prime relative to an unrelated target indicates activation of both

meanings, but a faster retrieval of the dominant meaning. By including both heterophonic and homophonic homographs, I was also able to explore two-dimensional ambiguity associated with heterophones in contrast with the semantic-only ambiguity of homophones and possibly account for apparent discrepancies among the findings of previous studies.

Most of the recent studies on lexical access and disambiguation of homographs have included ambiguous primes followed by targets related to either the dominant or the subordinate meaning of the prime (Simpson & Burgess, 1985). A similar priming paradigm was also employed for the two experiments in this study. Experiment 1 was designed to investigate the disambiguation process of heterophonic homographs, while Experiment 2 examined the same process in homophonic homographs. To control for potential influences of subject-related factors on the outcome[59], the same subjects participated in both experiments in a single testing session.

Experiment 1: In the first experiment, the role of priming in heterophonic homographs was examined via a visual masked priming technique in which participants responded to word or non-word targets primed by ambiguous heterophonic homographs. Each prime was followed by an unambiguous target related to either the dominant or the subordinate meaning of the prime or to neither (in the control condition). Facilitation in lexical decisions to any target related to the dominant or subordinate meanings of the prime relative to the control condition was taken as evidence for disambiguation of the homograph.

Participants: Twenty graduate students from the University of Ottawa with ages ranging from 20 to 35 years old participated in both experiments[60]. All participants were native-speakers of Persian who had been raised and mostly educated in Iran. They were right-

[59]In the study by Frost and Bentin (1992), different subjects participated in different experiments on the same homography phenomenon.
[60] Experiment 1 was followed by Experiment 2.

handed and had normal or corrected-to-normal vision. Participants received a small sum as compensation for their time.

Stimuli: The stimuli consisted of 30 ambiguous primes out of a pool of Persian heterophonic homographs represented by high and low frequency words in the two experimental conditions. In the dominant condition, the target items comprised the more frequent meaning of the prime while in the subordinate condition target items represented the less frequent meaning. A full-form lexicon or a word frequency list would normally be required to identify the dominant and subordinate meanings of each homograph, as well as to ensure accurate distribution of the primes throughout the various conditions.

The corpus developed by Nojoumian (2003) and the homograph frequency lists developed for the diacritizer was used in this experiment. To ensure accuracy, two native speakers pronounced heterophonic homographs in context to verify their correct phonological representations. The primes' dominant and subordinate meanings were ordered according to the extracted frequencies. A group of ten native speakers (other than participants in the two experiments) verified and rated both meanings of the bipolar homographs.

The primes were distributed randomly throughout the experimental conditions according to their more frequent or less frequent meaning realizations. Each of the 30 heterophonic homographs selected for Experiment 1 represented two primes in the two dominant and subordinate meaning conditions with different frequencies. Two targets from the matched ordered meanings were selected for each prime. One was semantically related to the dominant and the other to the subordinate meaning of the prime. Both targets were unambiguous and had the same syntactic category and part of speech as the prime.

Control Pairs: Each of the 60 targets was also paired with an unrelated prime. These 30 items were chosen from a different set of heterophonic homographs derived from the same original pool. The set of experimental and control primes had equal numbers of graphemes. Because of the unrelatedness of the control primes to the targets and the lack of relevance to the dominancy of the primes' meanings, the same prime was used for both unrelated targets (Frost & Bentin, 1992).

Fillers. In addition to the prime-target pairs, a total of 120 words and non-words formed the fillers. Although the filler words or primes were chosen from the original heterophonic homograph database, they differed from the previous 120 pairs.

An example of all the sets in 2x2 conditions is illustrated in the following table:

Ambiguous Prime	**درک**			
Phonological Alternatives	/d**a**rk/		/d**ara**k/	
Semantic Alternatives	Dominant Meaning		Subordinate Meaning	
	"understanding"		"hell"	
Conditions	Related	Unrelated	Related	Unrelated
Primes	**درک**	نسخ /nasax/ "abolish"	**درک**	نسخ /nosax/ "copy"
Targets	**فهم** "understanding"	**فهم** "understanding"	**جهنم** "hell"	**جهنم** "hell"

Table 9: Persian heterophonic homograph prime-target pairs in four conditions

Design: Two lists of words were formed; each contained four experimental conditions and 120 pairs in total comprised of 30 true prime-target pairs, 30 pseudo prime-target pairs and 60 fillers. The total 240 prime-target pairs were rotated across two lists (A & B) by a Latin-

square design in order to counterbalance prime-target relations (Related vs. Unrelated) across participants (Nojoumian et. al, 2006:6). Semantically related prime-target pairs in list "A" had unrelated primes with the same targets as in list "B". Hence, each target word served as its own control for the measurement of semantic facilitation in an across-subject design (Frost & Bentin, 1992:61).

Procedure: The participants were presented with instructions in Persian. They were asked to focus at the center of a computer monitor and make a lexical decision on an unambiguous target word (or non-word) by pressing either the designated "Yes" or "No" key on a standard keyboard. The keys were labeled in Persian. The right dominant hand was always used for the "Yes" key. The latest version of the DMDX application[61], which supports Unicode scripts like Persian, was used to calculate an accurate reaction time from the onset of the target display. Subjects were instructed to respond as quickly as possible. The prime was visually presented in size 12 black Times New Roman font on a white background for 6 ticks[62] (6 x 16.69 ms) or 100 ms and masked by the target. Ten hash-marks in size 30 intended for fixation pre-masked the prime for 500 ms. A font size of "30" was used to eliminate the discrepancy between the hash-mark size and the Persian fonts. The target stayed on the screen until the participant responded by pressing the 'Yes' or 'No' key. The inter-stimulus interval was 2600 ms from the participant's response to the onset of the following prime. If the participant failed to respond, the program would go to the following item after the time-out. The experimental session was preceded by a practice session of ten trials. The stimuli were presented in two blocks of 60 pairs with a short pause in between. Participants' responses to the lexical decision task were measured in terms of Reaction

[61] DMDX Version 3.1.4.1 by Jonathan C. Forster (2005)

[62] Tick is the unit used to refer to the refresh rate of a computer graphic interface. It is a number in milliseconds. The system used to run this experiment had a refresh rate of 16.69 ms.

Times (RTs) and accuracy. RTs were computed in milliseconds by the DMDX application and the results were recorded in a text file by the software. DMDX randomized the pairs in real-time while presenting them to the participants according to a header instruction.

Results: The mean RTs and standard deviations for correct responses and percentage of errors (ERs) were calculated for each participant in each of the four experimental conditions. Within each participant-condition combination, reaction times greater than a range of 2 standard deviations from the respective means were set aside and the mean was recalculated. The outliers accounted for less than 3% of all responses. Any single item which was responded to at a 60% accuracy level or worse was eliminated. The deletions accounted for only two items in unrelated pairs.

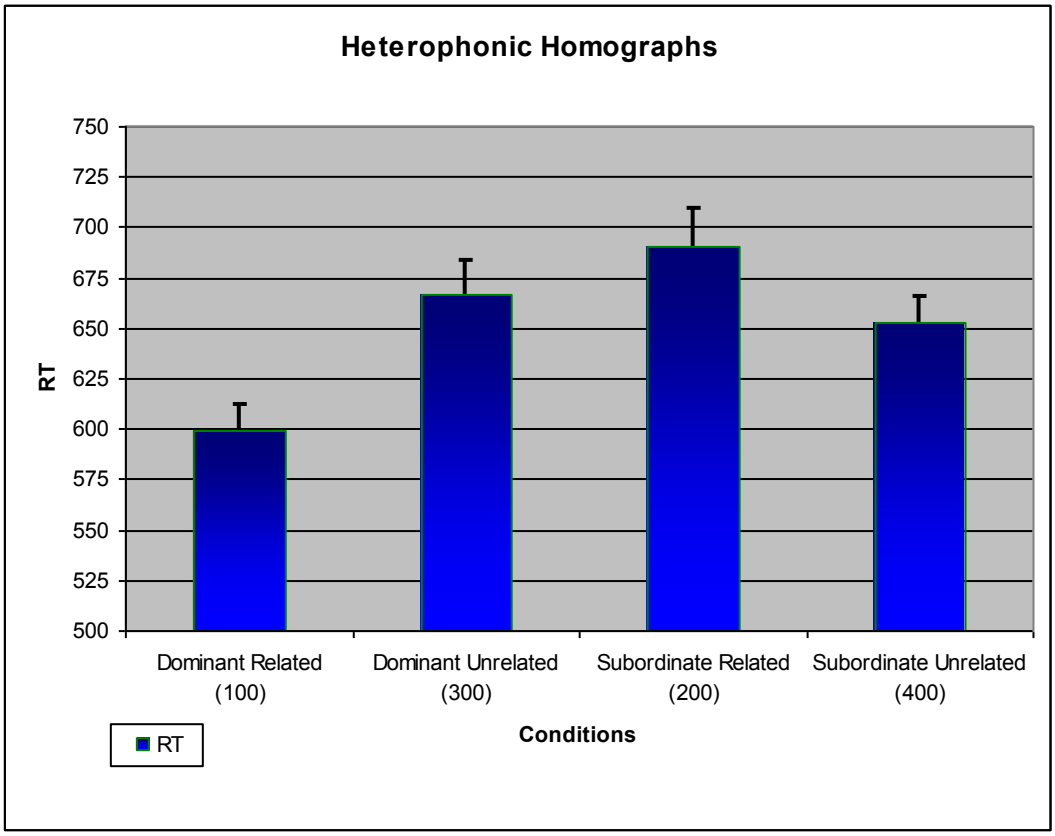
RTs and ERs for all four experimental conditions are presented in Table 9. Only data for experimental and control items were subjected to statistical analysis, since non-word targets were not relevant to the issues discussed in this paper. Subjects' mean RT were analyzed in a 2 (dominant vs. subordinate) x 2 (related vs. unrelated) within-subjects analysis of variance (ANOVA).

Heterophonic Homographs – /jang/ (war) جُنگ – جَنگ /jong/ (show)			
Variables	**Dominant Meaning**	**Subordinate Meaning**	**Non-words**
Unrelated – RT (ms) Standard Deviation	**666.36** (151)	**652.37** (137)	935.3
% of Errors	2.0	5.0	6.7
Related – RT (ms) Standard Deviation	**599.25** (106)	**690.14** (161)	
% of Errors	2.0	3.3	
Priming Effect (ms)	**67.11**	**-37.77**	

Table 10: Reaction Times & Percentage of Errors to Related & Unrelated Targets in Four Conditions with Heterophonic Homographs

The main effect of dominance (F(1,19)=19.228, *p<0.001*, MSe=1537) was significant. Responses were faster to dominant targets than to subordinate ones. There was no main effect of relatedness (F(1,19)=2.458, *P=0.133*). There was, however, a significant interaction between dominance and relatedness (F(1,19)=11.893, *p=0.003*, MSe=4624). To further investigate the significant interaction, separate ANOVA's were performed at each level of dominance. It was found that the significant interaction reflects the fact that RTs to targets related to the dominant meaning of the prime were faster than responses to unrelated targets (F(1,19)=19.051, *p<0.001*, MSe=2364). This is shown in Figure 20. Relatedness, however, was not significant for the subordinate meanings, though a trend towards significance was observed: F(1,19)=3.556, *p=0.075*. The differences in error rates within the dominant and subordinate conditions were not significant (F(1,19)=1.097, *p=0.308*).

Figure 20: Marginal means of dominant vs. subordinate meanings of the heterophonic homographs



The results of Experiment 1 indicate that in processing an isolated heterophonic homograph word within the mental lexicon, its dominant meaning is available to the processor before its subordinate or less frequent meaning. Therefore, the meanings of heterophonic homographs appear to be accessed in order of their frequency.

The slight trend to inhibition ($F(1,19)=3.556$, $P=0.075$) in accessing the subordinate meaning of heterophonic homographs seems compatible with MacKay (1970) and Simpson & Burgess (1985). MacKay (1970) was the first to argue that in the active processing of ambiguous sentences, "selection of one meaning required the inhibition of the other" (Simpson & Burgess, 1985:32).

The slightly later access to the subordinate meaning of heterophonic homographs compared with access to their dominant meaning could also be explained by the existence of multiple lexical entries for heterophonic homographs, which are by definition represented by several phonological realizations (Frost & Bentin, 1992:66). This "multiple-entries structure and the ordered-access process in heterophonic homographs" suggest a two-layered disambiguation process in which the phonological processor mediates between the orthographic form of the ambiguous word and its meanings.

Although the results suggest that each phonological realization of a heterophonic homograph is associated with its specific meaning and lexical entry, they do not necessarily apply to homophonic homographs. Previous findings demonstrating simultaneous access to all meanings of a homophonic homograph, regardless of their relative frequencies, support Frost & Bentin's (1992) proposal that only one lexical entry or phonological representation is connected to a homophone's several semantic nodes. Thus, while the data for heterophonic homographs appear to support a serial access model, in which the frequency of the meanings of ambiguous word play an important role, the data from other languages on homophonic homographs, such as English and Hebrew (Carpenter & Daneman, 1981; Frost & Bentin, 1992), provide evidence for an exhaustive access model in which meaning dominance plays a much less prominent role. In a variant of this proposal, Seidenberg et al. (1982) suggested that this relationship might be mitigated by part of speech. Thus, there would be two lexical entries for English noun-verb homophone pairs, such as "train". However, same-category homophones in noun-noun pairs (e.g., "boxer") would share a single lexical entry (Frost & Bentin, 1992).

In light of these conflicting interpretations, a comparison between heterophonic and homophonic homographs in the same language might clarify which factors differentiate between the two forms of ambiguities. Experiment 2, was designed for this purpose.

Experiment 2: This experiments examined activation of dominant (more frequent) and subordinate (less frequent) meanings of Persian homophonic homographs. Further, as Experiment 2 employed the same design and participants as for Experiment 1, a comparison of their results would allow us to discuss differences between the processing of heterophonic and homophonic homographs in Persian. The results could provide additional evidence regarding both the role of meaning dominance in the retrieval of homographs and serial and parallel access models of retrieval. If the two types of homographs showed significant interaction, it might also suggest that phonological ambiguity plays an important role in lexical access. If this prediction were borne out, then the results of the experiments would be compatible with findings of Frost and Bentin (1992) for Hebrew.

With the exception of the stimuli, described below, Experiment 2 employed the same subjects and methodology as Experiment 1.

Stimuli : For Experiment 2, thirty homophonic homographs were selected as experimental primes from the same database of Persian homographs described for Experiment 1. The dominant meaning of the homograph represented the more frequent realization of the prime, while the subordinate meaning represented its less frequent meaning. Two targets from the semantically ordered meanings of the homograph were selected for each prime. The first target was related to the dominant and the second target to the subordinate meaning of the prime. Both targets were unambiguous and had the same syntactic category and part of speech as the prime. Each of the 60 targets was also paired with an unrelated pseudo prime

in the control conditions. The 30 control items were chosen from the same homophonic homograph database, but were different from the main stimulus set. The experimental and control primes had equal numbers of graphemes.

In addition to the 120 prime-target pairs, a total of 120 different words and non-words from the original homograph database were also selected as fillers. The fillers were all different from those used in the first experiment. An example of all the sets in four conditions is illustrated in the following table:

Ambiguous Prime	جفت			
Phonological representation	/joft/			
Semantic Alternatives	Dominant Meaning		Subordinate Meaning	
	"pair"		"placenta"	
Conditions	Related	Unrelated	Related	Unrelated
Primes	**جفت**	ریش "beard"	**جفت**	ریش "injury"
Targets	**کفش** "shoe"	**کفش** "shoe"	**جنین** "embryo"	**جنین** "embryo"

Table 11: Persian homophonic homographs in four conditions

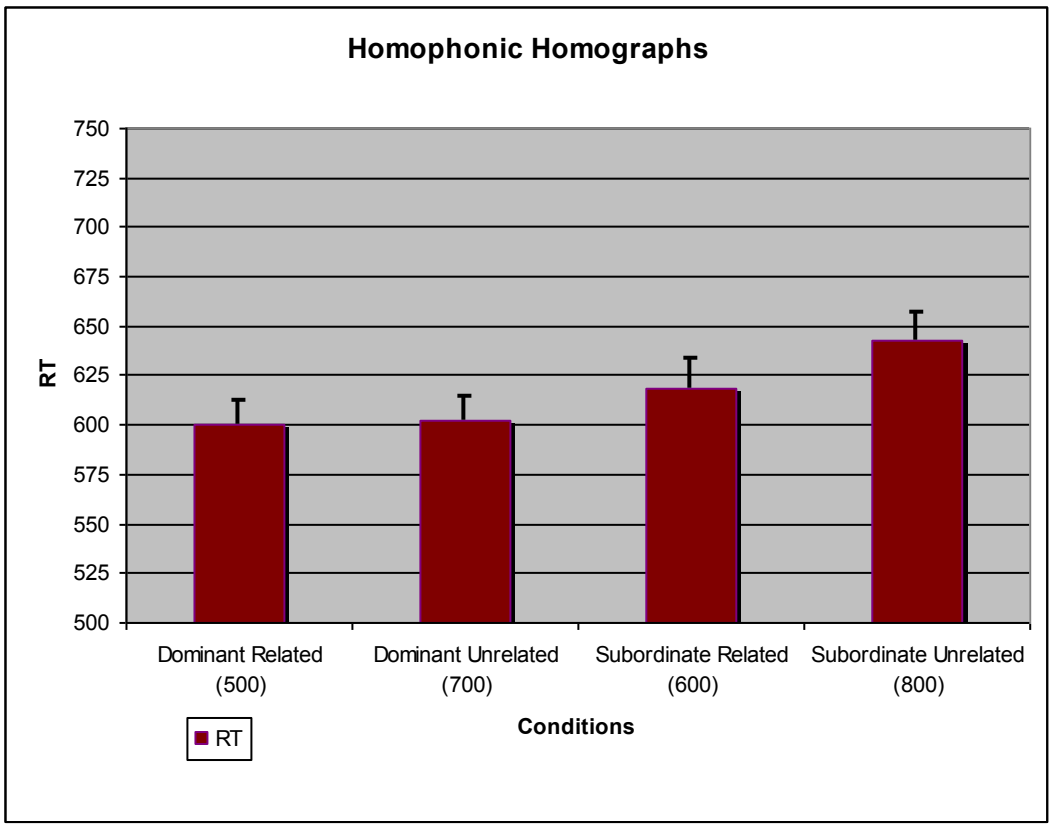
Results: Mean RTs and standard deviations for correct responses and the percentage of errors were calculated for each participant in each of the four experimental conditions (see Table 10). Within each participant-condition combination, the reaction times greater than a range of 2 standard deviations from the respective mean were set aside and the mean was recalculated. The outliers accounted for less than 3% of all responses. Only data for experimental and control targets were submitted to statistical analysis.

Table 12: Reaction Times & Percentage of Errors to Unrelated & Related Targets in four Conditions with Homophonic Homographs

Homophonic Homographs - جفت - جفت			
Variables	**Dominant Meaning**	**Subordinate Meaning**	**Non-words**
Unrelated – RT (ms) Standard Deviation	**602.15** (114)	**642.93** (128)	832.68
% of Errors	3.0	4.0	4.0
Related – RT (ms) Standard Deviation	**599.82** (113)	**618.19** (132)	
% of Errors	2.0	3.0	
Priming effect (ms)	**2.33**	**24.74**	

Analysis of the RTs showed a significant main effect of dominance ($F(1,19)=12.796$, $p=0.002$, $MS_e=1367$), but there was no significant interaction between the dominance and relatedness factors ($F(1,19)=1.49$, $P=0.24$). The main effect of relatedness was not significant either ($F(1,19)=2.49$, $p=0.13$) (see Figure 21). Therefore, the dominant meaning of the homophone was not activated faster than its subordinate meaning. However, because the effect of relatedness in the overall ANOVA was nearing a trend towards significance, I looked at this effect at each level of dominance and found a trend present for the subordinate items ($F(1,19)=3.36$, $p=.085$). The differences in error rates between the various conditions did not yield any significant effect ($F(1,19)<1$, $p>0.05$).

Figure 21: Means of dominant vs. subordinate meanings of the homophones



The results of Experiment 2 indicated no priming effect for the dominant meaning, but a trend toward significance in subordinate associates. The lack of any significant priming effect for dominant associates cannot be explained at this point. It might be due to artifacts of unrelated stimuli or to experimental factors such as length of SOA, which was 100ms. It is noteworthy that Frost and Bentin's (1992) findings seem to show a similar lack of priming effect at 100 ms SOA. When they used longer SOAs, they obtained faster RTs (Frost & Bentin, 1992). The ordering of the two experiments may also have affected the results of Experiment 2, which consistently followed shortly after Experiment 1. Furthermore, subjects may have been subconsciously cued to attend to ambiguity after exposure to the stimuli in Experiment 1.

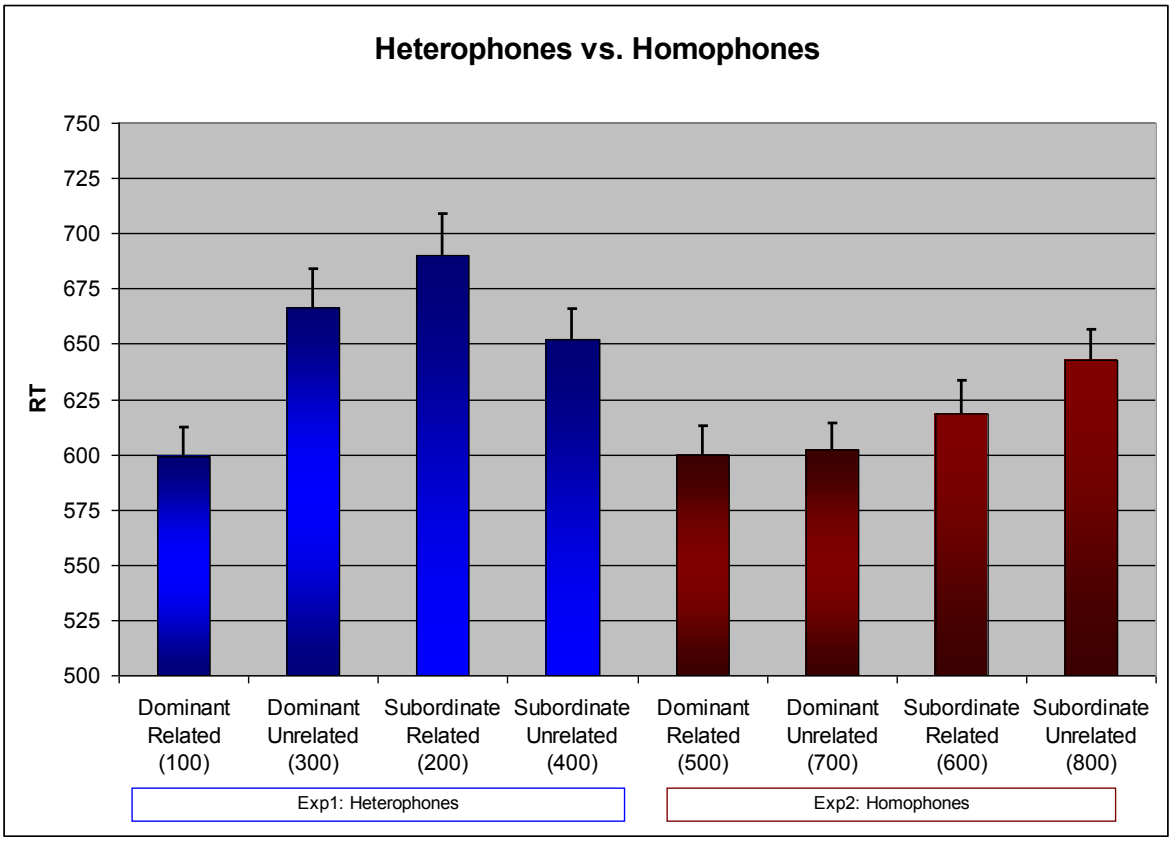
5.2.1.3 Conclusion on Psycholinguistic Evidence

The studies outlined above investigated the disambiguation of Persian heterophonic and homophonic homographs in isolation, without a biasing context. No priming effect was found in the process of disambiguating homophonic homographs. For heterophonic homographs, however, faster reaction times were measured for dominant than for subordinate meanings. The existence of separate phonological representations was posited to account for the different speeds of access in heterophonic homographs. A statistical comparison of the results of the two studies provided additional insight into the implications of these findings.

A 3-way cross experiment analysis of variance between Experiments 1 and 2 revealed a significant main effect of experiment ($F(1,19)=15.592$, $p=0.001$). The main effects of dominance and relatedness were also significant (for dominance: $F(1,19)=29.53$, $p<001$) and for relatedness: $F(1,19)=5.085$, $p=0.036$. Means of reaction times to subordinate meanings of homophonic homographs were significantly faster than for those of heterophonic homographs.

The interaction between the two factors of relatedness and dominance was also significant ($F(1,19)=4.521$, $p=0.047$) (see Figure 22). The three way interaction between the said factors was significant as well ($F(1,19)=15.90$, $p=0.001$), indicating that dominant and subordinate meanings were accessed differently in the two experiments. The remainder of the conclusion addresses the implications of these findings for our understanding of the lexical representation of homographs and for the access of ambiguous words.

Figure 22: Comparing Marginal Means of all Conditions in Heterophones and Homophones. (The first four bars in the left are heterophones, the rest are homophones)



These findings suggest that heterophonic and homophonic homographs may be accessed differently in the mental lexicon. The delayed activation of the subordinate meanings of heterophones is likely due to the doubly ambiguous nature of these words. Apparently, their phonological and semantic ambiguities demand more processing time during the disambiguation process. This delay suggests an activation pattern for Persian heterophonic homographs in line with the ordered-access model supported by Simpson & Burgess (1985) and confirmed by Frost & Bentin (1992). According to this model, lexical access happens one at a time in a serial manner. Once the dominant meaning of the ambiguous word is found in the lexicon and the semantic ambiguity is resolved, the search is paused and the

correct phonological representation is retrieved. Figure 23 illustrates the ordered-access model as it was put forward by Simpson & Burgess (1985) and Frost & Bentin (1992).

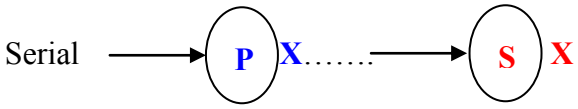


Figure 23: Ordered-access model (Pause sign → X, P=Primary meaning, S=Secondary meaning)

This model can be further refined based on the results of the present study, which demonstrated that meanings of heterophonic homographs are activated as a function of their relative frequencies. Once the print form of a heterophonic homograph is encountered in isolation, several lexical entries are activated in a serial manner in order of their meaning frequency. When the phonological processor accesses the phonological realization of the dominant meaning of the print form, it pauses the search process. However, if the dominant meaning is not the correct target, the search resumes until the subordinate meanings are accessed. This would account for the delay in accessing subordinate meanings of heterophonic homographs in the present study.

Onifer & Swinney (1981) proposed a serial access hypothesis for homophones as well, while still maintaining that their meanings were accessed simultaneously. This raises the question of how simultaneous access could occur in a serial manner, given that any concurrent access implies a parallel mechanism. However, Onifer & Swinney's (1981) proposal makes sense if we consider that parallel access occurs during a pre-conscious stage, while selection of alternative meanings takes place in a serial manner at a conscious selection stage in their task. While these experiments do not confirm any model for the processing of homophones, the comparison between the two experiments suggests that heterophones are processed differently from homophones.

The difference between the priming effects of the subordinate meaning of the heterophonic and homophonic homographs in the two experiments in this study suggests that phonological ambiguity is costly for the disambiguation processor of the mental lexicon, as it requires a 'detour' through the phonological processor. This suggestion is a good justification for my attempt to reduce such ambiguities in the orthography by diacritizing the input. A lexical access model for heterophonic homographs may illustrate a possible 'detour' in Figure 24.

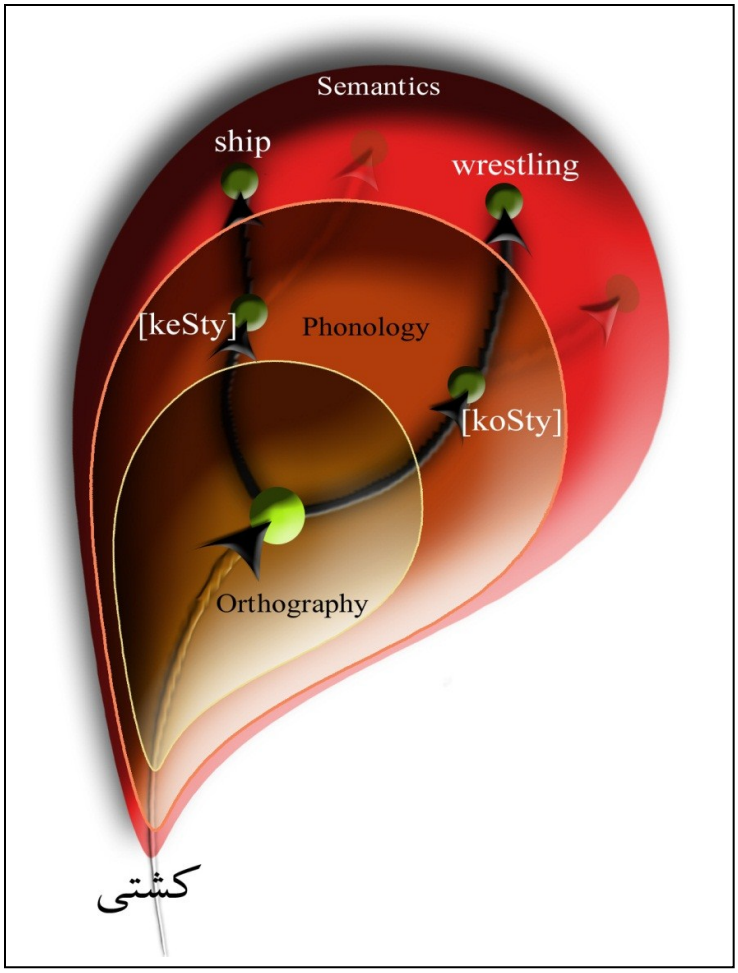


Figure 24: A proposed model for lexical access. Phonology mediates between the orthographical and semantic forms in Persian heterophonic homographs.

The phonological component plays a mediating role in the disambiguation of heterophones. It also indicates that there is no direct connection between the orthography and semantics in the processing of heterophones. However, follow-up studies are needed before any firm conclusions can be drawn on the processing of Persian homographs. Future replications of the present experiments could test for ordering effects[63] by counterbalancing the stimuli from Experiments 1 and 2 or by incorporating both sets into the same presentation grid. The experiments could also be replicated with longer SOAs to determine whether this factor yields significantly different RTs.

It would also be interesting to examine the effects of biasing contexts on access to the meanings of both types of homographs. For example, Hogaboam & Perfetti (1975) & Onifer & Swinney (1981) showed that when the context biased the subordinate meaning of the prime, both meanings were accessed with a slight facilitation of the biased one.

The consistency of the results of the current study with those of Frost and Bentin (1992) appear to rule out certain language specific effects on the process of disambiguation of heterophonic homographs, as Persian does not resemble Hebrew, a Semitic language, in its morphosyntax. Nonetheless, research encompassing other languages would help to further address the issue of language specific effects. As Whitaker (2006:3) notes, "studying lexical processing and structure in different languages will continue to refine our linguistic and psychological models of the mental lexicon."

[63] For the current study, the homophones were not subject to phonological disambiguation so the potential ordering effect was not relevant here.

5.2.2 Implications of the Psycholinguistic Study

A possible conclusion that can be drawn from practical simulation of these findings may also shed more light on the feasibility of the proposed lexical access models described above. The results of running diacritizer on the test corpus showed that the frequency of different meanings of a heterophonic homograph plays an important role in disambiguating the homograph even in isolation (out of a context). Therefore, the mechanisms such as the frequency information that was used to retrieve the correct phonological representation of an ambiguous homograph by the mental processor, can be pretty much simulated by a computer application. But, what happens if only the high frequency candidate is chosen? Then, the context should be checked. Although the results showed high accuracy rates in disambiguating the heterophonic homographs, a robust diacritizer should be able to deal with lower frequency candidates as well. Therefore, the effect of frequency of the heterophonic homographs on the disambiguation of data from the test corpus in a biasing context should also be examined.

5.3 Homograph Disambiguation in the Context

So far, the role of frequency in the activation and access of isolated homographs in the mental lexicon was studied. In NLP, the context can also be used in order to disambiguate word sense. As Simpson and Burgess (1985:32) suggest “the most widely accepted explanation of the processing of ambiguous words in sentence context holds that when a word first occurs, all of its meanings are initially retrieved. Subsequently, the context is used to select the appropriate meaning, maintaining it and discarding irrelevant ones”

(Simpson and Burgess, 1985). In fact, context is the only means to identify the meaning of a polysemous word (Ide et al., 1998).

Corpus linguistic methods have enabled researchers to scrutinize words and collocations in qualitative and quantitative studies of word sense disambiguation. While early rule-based results reported to have encouraging results (e.g., Weiss, 1975; Kelley and Stone, 1975), recent works encouraged by advances in technology have focused more on statistical and hybrid approaches to word sense disambiguation depending on different NLP application needs (Ide et al., 1998).

In speech technology and particularly speech synthesis, where pronunciation of words and sentences are the main task, a disambiguator is required to distinguish between dominant and subordinate meanings of a binary homograph in order to generate a correct phonological form for the ambiguous input. However, more complex language applications like machine translation systems would need “finer granularity of the sense” to deal with different types of lexical and syntactical ambiguities in a broader context (Ide et al., 1998:28).

In this study, I will explore possibilities in disambiguation tasks in which the correct diacritization of a binary homograph should be chosen. A binary heterophonic homograph has two distinct senses with two different frequencies (coded in its part of speech) in which "sense distinctions correspond to pronunciation differences". Yarowsky (1993) determined that in cases of binary ambiguity, there exists “one sense per collocation,” that is, in a given collocation a word is used with only one sense with 90-99% probability (Yarowsky, 1993:266).

In order to explore Yarowsky's idea, we need to know the definition of a collocation first. In Yarowsky's term, a collocation is "the co-occurrence of two words in some defined relationship" (Yarowsky, 1993:267). The relationship can be defined as how far the two words could be from each other. The closest proximity is one word to the left or right and the farthest proximity in defined windows of several words in each direction. Part of speech of the words in a collocational relation plays an important role because function words and content words do not behave the same way. Yarowsky (1993 & 1997), proposed a decision list model containing the probability distribution of the senses and collocations in an ordered or weighted list. A word sense disambiguation algorithm could then use the list and choose the more probable meaning of a homograph based on the decision list data. The probabilities are calculated based on the following formula (adapted from Yarowsky, 1993:268) [64]:

$$ABS\left(Log\left(\frac{pr(sense1 \mid collocation i)}{pr(sense2 \mid collocation i)}\right)\right)$$

Yarowsky (1993) showed significant precision rates above 90% for different relations between a content word to the right or left of an ambiguous word. He also showed that nouns are best disambiguated by the adjectives that modify them while verbs are more responsive to their objects than to their subjects. According to Yarowsky, models that consider an immediate content word to the left or right of an ambiguous word are better defined especially for adjectives and verbs. As it could be seen from the above study, the performance of Yarowsky's model is high even with low frequency samples which suggest that modeling local collocations would be beneficial to a word sense disambiguation algorithm (Yarowsky, 1993).

[64] Sense1 and Sense2 can also represent different pronunciations of a homograph (Yarowsky, 1997:163).

Yarowsky's (1993, 1997) model seems to be a robust technique in disambiguation of homographs. However, for the Persian binary homographs in which several collocations would have zero probability[65] different techniques should be explored to calculate a smoothing formula to capture this complexity.

For the sake of simplicity and mere demonstration of Yarowsky's significant contribution to the field of word-sense disambiguation, some examples from Persian heterophonic homographs will be explored and a prototype model[66] will be suggested in which the collocations for the less frequent meaning of a homograph[67] are examined first to determine the correct phonological representation of the ambiguous word. In case of no context for the less frequent meanings of the homograph, the search may continue to examine the more frequent meaning context, or stops and chooses the dominant form of the homograph with more frequent meaning[68]. The advantage of this model is twofold. First, there are fewer contexts for less frequent meanings of homographs than more frequent ones, resulting in faster and more efficient search. Second, the lack of context effect would be eliminated in case of no context for the less frequent candidates if we choose to stop the search process and announce the phonological representation of the more frequent meaning as the winner. Nevertheless, a significant amount of data and instances of collocations is required to model a reliable and robust context for both meanings of the homographs.

[65] For low frequency homographs without any single occurrence in the corpus, the probability becomes zero. However, there are mathematical ways to remedy this issue. Please refer to Yarowsky (1997) for the solutions.
[66] Note that disambiguation issue will be explored from a corpus linguistic approach in order to put forth research possibilities for the disambiguation of homographs in Persian NLP, which is still far from well-researched languages.
[67] coded as HL in POS tags
[68] coded as HH in POS tags

Table 13 shows the context with a window of one word to the right[69] and one word to the left of the less frequent meaning of the heterophonic homograph (کشتی) /kSty/ realized as [keSty] (ship) in the training Persian corpus. The dominant meaning of this homograph realized as /koSty/ (wrestling) has a frequency of 151, whereas its subordinate meaning has a frequency of 50 in the training corpus. Table 13 also shows the subordinate meanings of the heterophone in 44 different contexts[70].

Table 14, however, shows the context with the same criterion said above for both phonological representation of the same heterophone in the TEST corpus. A quick examination and comparison between the two tables reveals an interesting result in this example.

[69] The examples contain function words as well.
[70] The frequency of a sample collocation of "koSty OzAd" (freestyle wrestling) is 48.

Left Context	کِشتی keSty (ship)	Right Context	No.		Left Context	کِشتی keSty (ship)	Right Context	No.
کانتینر	1	فروند	23		باربری	1	چندین	1
باری	1	یک	24		که	1	یک	2
کیلومترها	1	این	25		در	1	یک	3
باربری	1	یک	26		حامل	1	کردند	4
با تکانهای	1	این	27		به	1	با	5
در	1	نخستین	28		جنگی	1	یک	6
در	1	تاجر	29		غیرنظامی	3	یک	7
حمل	1	دو	30		غیرنظامی	2	هویت	8
باربری	1	دو	31		می‌سازد	1	نفر	9
با	1	برفراز یک	32		آرامش	1	با	10
حامل	1	کرد	33		متانت	1	به	11
سوسان	1	در	34		ناشناس	1	مورد	12
که	2	این	35		از	1	دو	13
یک	1	این	36		به‌طوری	1	به	14
سوسان	1	مصادره	37		تجاری[71]	1	در	15
حمل	1	با	38		عن‌قریب	1	این	16
فرماندهی	1	داد	39		تجاری	1	یک	17
نیرو	1	در	40		تجاری	1	این	18
در	1	ناخدای	41		کره‌ای	1	یک	19
کارین	1	سناریوی	42		حامل	3	توقیف[72]	20
ناوچه	1	یک	43		اقیانوس‌پیما	1	ساخت	21
نظامیان	1	یک	44		توسط	1	ساخت	22
Total Frequency (HL): 50								

Table 13: Subordinate (Less-frequent meaning) right and left contexts for Persian homograph /kSty/ realized as /keSty/ (ship) in the training corpus. Total frequency for the dominant meaning is 151.

Left Context	کُشتی koSty (wrestling)	Right Context	No.		Left Context	کِشتی keSty (ship)	Right Context	No.
آزاد	3	برتر	1		تجاری	1	یک	1
همدان	1	پیشکسوتان	2		تجاری	1	سلاح	2
در	1	علاقمند	3		بدون	1	توقیف	3
با	1	فدراسیون	4					
ورزش	1	»	5					
مازندران	1	هیأت	6					
بود	1	بی‌احترامی به	7					
Total Frequency (HH): 9					Total Frequency (HL): 3			

Table 14: Right and left context for both subordinate and dominant meanings of the homograph /kSty/ in the test corpus.

[71] commercial (... **commercial ship**)
[72] confiscating (**confiscating a ship** ...)

All the three collocations in the test corpus have a correspondent disambiguating context in the training context coded in colors. Item 1 in the test corpus has its counterpart in training corpus collocation 17. Item 2 corresponds to collocation 15 and item 3 to collocation 20 in the training corpus. The advantage here is that we had only examined three (or two, if we consider the same left context in items 1 and 2) collocations with the training corpus. Otherwise, we should have checked twice as many contexts for the more frequent meaning of the homograph.

Nevertheless, the decision lists proposed by Yarowsky (1993) would result in more accurate grading of the collocations and perhaps higher accuracy rates to determine correct pronunciation of heterophones but it definitively needs more extensive quantitative studies and requires considerable annotated and diacritized data which is still not available for the Persian language.

In case of lack of context for the less frequent meaning of a homograph, we have two options. The first is to continue the search process as was proposed in the mental lexicon findings and retrieve the context for the more frequent meanings of the homograph; the second option is to stop the search right away and choose the more frequent meaning of the homograph. The test results on the test corpus showed that more frequent meanings of the heterophonic homographs had a chance of being correctly recognized at 88.5% of the cases without even looking at their context (see table 7). It should be mentioned, however, that further research and larger test corpus are required to make a firm claim on this issue. the test corpus contained about 100,000 words and probably was not able to represent a collocation context for both less and more frequent meanings of the homographs. I hope that the findings of this study and the collocation demonstration be used as a starting point for future

exploration of this issue. Furthermore, ambiguities are not limited to binary homographs and depending on each NLP application, different types of models and approaches should be accounted for. For the purpose of diacritization, however, the findings seem to be promising and encouraging enough to pursue further research and evaluations.

6 Conclusions

This study described the overall design and implementation of a diacritizer for the Persian orthography based on the FST tools developed at Xerox[73]. I developed a system that could potentially act as the core component of an NLP application for Persian. While a number of morphological analyzers have been developed for Persian, none is reported to handle diacritics. The diacritizer will eventually incorporate an enriched phonetic transcriber into the Grapheme-to-Phoneme converters used in a speech-enabled application. Evidently, psycholinguistics studies of human mental lexicon have contributed to our understanding of cognitive mechanisms used by the brain to deal with words and their corresponding meanings. I tried to benefit from this contribution by bridging the gap between psycholinguistic and computational linguistics research. The findings emphasized the role of morphological analysis and frequency of the ambiguous words in word sense disambiguation by the human lexical processor.

The results of the application of the diacritizer showed an acceptable coverage both on the testing corpus and on another corpus (Bijankhan). The diacritizer has the potential to improve disambiguation rates by creating enriched part of speech tags containing frequency information for Persian homographs. The system was tested and documented during every phase of implementation and the results were used to refine rules that are more efficient in improving the final performance.

[73] Xerox FST tools are licensed free for research purposes.

The following conclusions summarize this study:

1) A rule-based approach using state-of-the-art FST for the diacritization of Persian orthography works efficiently. It is definitely more laborious and time intensive than statistical based methods, but it generates results that are more accurate. Once there are enough resources to do this task by inductive methods, a hybrid approach, which takes morphological rules into consideration, would be recommended.

2) Psycholinguistics research on morphology and disambiguation of homographs can shed light on our understanding of how the human mental lexicon works and can pave the way for a better simulation of the human processor in NLP.

3) The starting point for the development of NLP applications is corpora. Careful development of corpora would result in developing robust applications.

6.1 Possible Extensions and Future Works

Possible extensions to this thesis would include evaluating the developed diacritizer on a larger test corpus and the fine-tuning of its rules on a larger training corpus. Enriching the list of static multi units (compound nouns) would definitely make the Tokenizer more robust and efficient. Finally, the addition of proper and foreign names and the development of better guesser algorithms would greatly benefit the coverage rate of the diacritizer.

Other extensions would include the development of a robust parser capable of inserting Ezafe and disambiguating homographs. This should include all types of syntactic and morphological ambiguities. More accurate frequency lists containing frequency information for heterophonic homographs would be very useful. Collocation information would also increase the rate of disambiguation success and it is an important factor in developing a word-sense disambiguator. The proposed model for the disambiguation of heterophonic homographs and insertion of Ezafe would probably contribute to our understanding of this understudied phenomenon in Persian and could be considered as a starting point for relevant future research on word-sense disambiguation in Persian.

The ultimate and ideal product of this study can be used to develop computational linguistic applications including speech synthesizer, speech recognizer, machine translation and information retrieval systems. The developed diacritizer can also be used as an embedded e-learning technology in the field of teaching Persian language as a foreign language. Persian language learners would greatly benefit from a diacritizer at the early stages of reading and writing skills development.

References

Amtrup, Jan W., H. Mansouri Rad, K. Megerdoomian & R. Zajac (2000). *Persian-English Machine Translation: An Overview of the Shiraz Project,* NMSU, CRL, Memoranda in Computer and Cognitive Science Report (MCCS-00-319).

Attia, M. (2008). *Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation*, PhD Thesis, University of Manchester, UK.

Azimzadeh A. & M.M. Arab (2007). *"The Persian Morphological Parser using POS Tagger"*. In proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages, Linguistic Institute, Stanford, California

Baluch, B & D. Besner (2001). *Basic Processes in Reading: Semantics Affects Speeded Naming of High-frequency Words in an Alphabetic Script*. Canadian Journal of Experimental Psychology, Vol. 55(1).

Baluch, B. (1992). *Reading With and Without Vowels: What are the Psychological Consequences?* Journal of Social and Evolutionary Systems, Vol. 15 (1), pp. 95-104.

Bateni, M. (1995). *"towsife sakhteman-e dasturi-ye zaban-e farsi"* (Persian syntax). Tehran: Amirkabir Publishers.

Beckley, R. (2003). *The Design, Implementation, and Effectiveness, a Farsi Word Stemmer*. Masters Thesis, University of Nevada, Las Vegas.

Beesley, K. R. & L. Karttunen (2003). *Finite State Morphology*, CSLI Studies in Computational Linguistics, Stanford University, California.

Beesley, K. R. (1996). "*Arabic Finite State Morphological Analysis and Generation"*. In COLING 1996, Copenhagen, pp. 89-94

Bentin, S. & Frost, R. (1987). *Processing Lexical Ambiguity and Visual Word Recognition in a Deep Orthography*, Memory and Cognition, Vol. 15 (1), 13-23.

Bijankhan M. & S. Moradzadeh (2002). *Homographs in Persian Orthography*, In proceedings of the 1st Workshop on Persian Language & Computer, Tehran, pp53-63.

BijanKhan, M. (2004). *The Role of the Corpus in Writing a Grammar: An Introduction to a Software*. Iranian Journal of Linguistics, 19(2).

Bijankhan, M. et al. (2004e). *The Persian text corpus*. In Proceedings of the 1st Workshop on Persian Language & Computer, Tehran University, Tehran, pp. 143-144.

Boudelaa, S., & Marslen-Wilson, W.D. (2000). Non-concatenative morphemes in language processing: Evidence from Modern Standard Arabic. Proceedings of SWAP, Nijmegen, MPI for Psycholinguistics, 23–26.

Boudelaa, S., & Marslen-Wilson, W.D. (2001). Morphological units in the Arabic mental lexicon. Cognition, 81 (1), 65–92.

Buchner A., A. Zabal, and S. Mayr (2003). *Auditory, visual, and cross-modal negative priming*. Psychonomic Bulletin & Review, 10 (4), pp 917-923.

Carpenter, P. A., and M. Daneman (1981). *Lexical Retrieval and Error Recovery in Reading: A Model Based on Eye Fixation.* Journal of Verbal Learning and Verbal Behavior, Vol. (20), 137-160.

Davy, Graham; Eds. (2006). Encyclopaedic Dictionary of Psychology, Hodder Arnold, UK.

Dehdari, J. & D. Lonsdale (2002). *A link grammar parser for Persian*, Brigham Young University Provo, UT.

Forster, Jonathan C. (2005). DMDX Version 3.1.4.1, http://www.u.arizona.edu/~jforster/

Frost, R., Forster, K.I., & Deutsch, A. (1997). *What can we learn from the morphology of Hebrew? A masked-priming investigation of morphological representation.* Journal of Experimental Psychology: Learning, Memory, and Cognition, 23 (4), 829–856.

Frost, R. & M. Kampf (1993). *Phonetic Recoding of Phonologically Ambiguous Printed Words*, Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol. 19 (1), 23-33.

Frost, R. & Sh. Bentin (1992). *Processing Phonological and Semantic Ambiguity: Evidence From Semantic Priming at Different SOAs*, Journal of Experimental Psychology: Learning, Memory, and Cognition Vol. 18 (1), 58-68.

Frost, R., Feldman, L.B. & Katz, L. (1990). *Phonological Ambiguity and Lexical Effects on Visual and Auditory Word Recognition*, Journal of Experimental Psychology: Learning, Memory, and Cognition Vol. 16 (4), 569-580.

Gadsby, N., Arnott W. L., and Copland D. A. (2008). *An Investigation of Working Memory Influences on Lexical Ambiguity Resolution, Neuropsychology*, Vol. 22 (2), 209–216.

Ghomeshi, J. (1997). *Non-Projecting Nouns and the Ezafe Construction in Persian,* Natural Language and Linguistic Theory, Vol (15): p729-788, Kluwer Academic Publishers, Nederland

Ghayoomi, M., S. Momtazi, and M. Bijankhan (2010). *A Study of Corpus Development for Persian*, International Journal on Asian Language Processing, Number 20, Volume 1, PP 17-33.

Gottlob, L. R., Goldinger, S. D., Stone, G. O. & Van Orden, G. (1999). *Reading Homographs: Orthographic, Phonologic, and Semantic Dynamics*, Journal of Experimental Psychology: Human Perception and Performance, Vol. 25 (2), 561-574.

Gorfein, D. S. ed. (2001). On the Consequences of Meaning Selection: Perspectives on Resolving Lexical Ambiguity, American Psychological Association: Washington DC.

Hankamer, J. (1989). Morphological Parsing and the Lexicon. In Marslen-Wilson, W. (Ed.), Lexical Representation and Process, pp. 392-408. MIT Press, Cambridge, MA.

Hogaboam, T. W., & Perfetti, C.A. (1975). *Lexical Ambiguity and Sentence Comprehension,* Journal of Verbal Learning and Verbal Behavior, Vol. 14, 265-274.

Holley-Wilcox, P. & Blank, M. A. (1980). Evidence for Multiple Access in the Processing of Isolated Words, Journal of Experimental Psychology: Human Perception and Performance, Vol. (6), 75-84.

Ide, N. & J. Véronis (1998). *Word Sense Disambiguation: The State of the Art*, Computational Linguistics, 24 (1)

Jurafsky, D. S. & J. H. Martin (2000). *Speech and Language Processing*, Prentice-Hall, Inc. NJ

Kahnemuyipour, A. (2000). *Persian Ezafe Construction Revisited: Evidence for Modifier Phrase*, in Jensen, John T and Gerard van Herk (eds.) Cahiers Linguistique d'Ottawa, proceedings of the 2000 annual conference of the Canadian Linguistic Association, p173-185

Kellas, G, Ferraro, F. R., and Simpson, G. B. (1988). *Lexical Ambiguity and the Time course of the Attentional Allocation in Word Recognition*, Journal of Experimental Psychology: Human Perception and Performance, Vol. 14 (4), 601-609.

Keshani Kh. (1993). *"farhange zansu" (Dictionnaire inverse de la langue persane)*, Tehran: University Publisher's Center

Keshani Kh. (1993). *"eshteqaqe pasvandi dar zabane farsi ye emruz" (La derivation suffixale en persan contemporain)*, Tehran: University Publisher's Center

Koskenniemi, K. (1983). *Two-level Morphology: A General Computational Model for Word-form Recognition and Production*, Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Kroll, J. F., and Schweickert, J. M. (1978). *Syntactic Disambiguation of Homographs*. Paper presented at the Nineteenth Annual Meeting of the Psychonomic Society, San Antonio, TX.

Lambton A. K. (1996). *"Persian Grammar"*, London: Cambridge University Press

Lazard, G. (1992). *"A Grammar of Contemporary Persian"*, Mazda Publishers: California

Loeng, C. K. & P. Cheng (2003). *Consistency effects on lexical decision and naming of two-character Chinese words*, Reading and Writing, Vol. 16 (5), 455-474.

Longtin, C., J. Segui and P. A. Halle (2003). Morphological Priming without Morphological Relationship, Language and Cognitive Processes, 2003, 18 (3), pp 313-334.

MacKay, D.G. (1970). *Mental Diplopia: Towards a Model of Speech Perception at the Semantic Level*. In G. B. Flores d'Arcais & W. J. M. Levelt (Eds.), Advances in Psycholinguistics (pp 76-100). Amsterdam: North-Holland.

Manning, C. D. & H. Schutze (2001). *Foundations of Statistical Natural Language Processing,* The MIT Press

Marslen-Wilson, W.D., Tyler, L.K., Waksler, R., & Older, L. (1994). Morphology and meaning in the English mental lexicon. Psychological Review, 101 (1), pp 3–33.

Marslen-Wilson, W.D. (2001). Access to lexical representations: Cross-linguistic issues. Language and Cognitive Processes, 16 (5/6), pp 699–708.

Megerdoomian, K. (2000). "*Unification-Based Persian Morphology",* In proceedings of CICLING 2000, Alexander Gelbukh, ed. Centro de Investigación en Computación-IPN, Mexico

Megerdoomian, K. (2004). "*Finite-State Morphological Analysis of Persian",* In proceedings of COLING 2004, University of Geneva: Geneva, Switzerland

Megerdoomian, K. (2006). "*Extending a Persian Morphological Analyzer to Blogs",* University of Maryland, College Park

Moinzadeh, A. (2001). An Antisymmetric , Minimalist Approach to Persian Phrase Structure, Ph.D. dissertation, University of Ottawa

Najafian, A. (2008). *A Survey of the Lexical Access of Derived Words: A Probe into Persian Mental Lexicon*. PhD Thesis, University of Tarbiat Modares, Tehran.

Neely, J. H. (1977). *Semantic Priming and Retrieval from Lexical Memory: Role of Inhibitionless Spreading Activation and Limited-capacity Attention*, Journal of Experimental Psychology: General, Vol. 106, 226-254.

Neysari S. (1996). *"dasture khatte farsi" (Persian Orthography)*, Tehran: Ministry of Culture Press

Nojoumian, P., P. Shabani Jadidi, and A. Tangestanifar (2006). *Morphological Priming of Multi-morphemic Persian Words in Mental Lexicon*, Proceedings of the 2nd Workshop on Computer and Persian Language, Tehran University Press, pp28-38

Nojoumian, P. (2003). *Developing a Multidisciplinary Monolingual Text Corpus for Persian***,** Catholic University of Leuven, Belgium

Oflazer, K. (1993). "*Two-level Description of Turkish Morphology"*. In proceedings, Sixth Conference of the European Chapter of the ACL

Onifer, W. & D. A. Swinney (1981). Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias, Memory & Cognition, Vol. 9 (3), 225-236.

Persian Academy of Language and Literature (2005). Persian Orthography. http://www.persianacademy.ir/fa/dastoorpdf.aspx

Pexman, P. M., S. J. Lupker, and Hino Y. (2007). *Cross-modal repetition priming with homophones provides clues about representation in the word recognition system*. The Mental Lexicon, Vol. 2 (2), pp. 183-214.

Philips, C. (2005). "*The Real-Time Status of Island Phenomena"*, Department of Linguistics; Neuroscience and Cognitive Science Program, University of Maryland.

Riazati, D. (1997). *Computational Analysis of Persian Morphology*. Masters Thesis, Department of Computer Science, RMIT.

Roark, B. & R. Sproat (2007). *Computational Approaches to Morphology and Syntax*, Oxford University Press

Samareh, Y. (1999). *"avashenasi ye zabane farsi" (Persian language phonetics)*, Tehran: University Publishers Centre.

Samiian, V. (1983). Origins of Phrasal Categories in Persian, an X-bar Analysis, Ph.D. dissertation, UCLA

Seidenberg, M. S., Tanenhaus, M. K., Leiman, J. M., & Bienkowsky, M. (1982). *Automatic Access of the Meaning of the Ambiguous Words in Context: Some Limitations of Knowledge-based Processing*. Cognitive Psychology, 14, 489-537.

Sereno, S. C., O., Patrick J., and Rayner, K. (2006). *Eye Movements and Lexical Ambiguity Resolution: Investigating the Subordinate-Bias Effect,* Journal of Experimental Psychology: Human Perception and Performance, Vol. 32 (2), 335–350.

Shamsfard, M. & M. Sadrmousavi (2007). *"A Rule-based Semantic Role Labeling Approach for Persian Sentence"*, in proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages, Linguistic Institute, Stanford, California

Sharifi-Atashgah M. & M. Bijankhan (2008). "*Corpus-based Analysis for Multi-Token Units in Persian*", The Faculty of Letters and Humanities, Tehran University

Simpson, G. B. (1981). *Meaning Dominance and Semantic Context in the Processing of Lexical Ambiguity,* Journal of Verbal Learning and Verbal Behavior, Vol. 20, 120-136.

Simpson, G. B., Burgess C. (1985). *Activation and Selection Process in the Recognition of Ambiguous Words*, Journal of Experimental Psychology: Human Perception and Performance, Vol. 11 (1), 28-39.

Sproat, R. (1992). *Morphology and Computation*, MIT Press, Cambridge, Massachusetts

Stanovich, K. E. & West, R. F. (1979). *Mechanisms of Sentence Context Effects in Reading: Automatic Activation and Conscious Attention*, Memory & Cognition, Vol. 7, pp. 77-85.

Stanners, R. F., J. Neiser, W.P. Hernon, & R. Hall (1979). *Memory representation for morphologically related words*, Journal of Verbal Learning and Verbal Behavior, Vol. 18, pp. 399-412.

Weiner, I. B., A. F. Healy, D. K. Freedheim, R. W. Proctor, J. A. Schinka (2003). The Handbook of Experimental Psychology. New Jersey: John Wiley & Sons Inc.

Whitaker, H. A., (2006). *Words in the Mind, Words in the Brain*, The Mental Lexicon, Vol.1 (1): John Benjamin's Publishing Company, 3-5.

Yarowsky, D. (1997). *Homograph Disambiguation in Text-to-speech Synthesis*. In van Santen, Jan T. H.; Sproat, Richard, Olive, Joseph P., and Hirschberg, Julia. Progress in Speech Synthesis. New York: Springer-Verlag, pp. 157-172.

Yarowsky, D. (1993). *One Sense per Collocation*. In Proceedings, ARPA Human Language Technology Workshop, Princeton, NJ, pp 266-271

Yoosofan, A., A. Rahimi, M. Rastgoo & M. M. Mojiri (2010). *Automatic Stemming of Some Arabic Words Used in Persian Through Morphological Analysis Without a Dictionary*. World Applied Sciences Journal, 8 (9), pp. 1078-1085

Appendix I: Persian Phonetic & Phonological Table

Grapheme	Transliteration	Phoneme	Phonetic [IPA]	Unicode	Example
ا	A	/a/	[ɒ:]	0627	ایران
آ	O	/A/	[ɒ:]	0622	آدم
ب	b	/b/	[b]	0628	باران
پ	p	/p/	[p]	067E	پا
ت	t	/t/	[t]	062A	تو
ط	T	/t/	[t]	0637	طب
ث	c	/s/	[s]	062B	اثیری
س	s	/s/	[s]	0633	سم
ص	C	/s/	[s]	0635	صد
ج	j	/j/	[dʒ]	062C	جم
چ	K	/ch/	[tʃ]	0686	چپ
ح	H	/h/	[h]	062D	حال
ه	h	/h/	[h]	0647	هم
خ	x	/kh/	[x]	062E	خوب
د	d	/d/	[d]	062F	داد
ز	z	/z/	[z]	0632	زار
ذ	M	/z/	[z]	0630	ذال
ض	Z	/z/	[z]	0636	ضعف
ظ	D	/z/	[z]	0638	ظالم
ر	r	/r/	[r]	0631	راه
ژ	J	/zh/	[ʒ]	0698	ژرژ
ش	S	/sh/	[ʃ]	0634	شب
ع	E	/?/	[ʔ]	0639	عمه
ء	R	/?/	[ʔ]	0621	املاء
ق	q	/gh/	[ɢ]	0642	قلب
غ	G	/gh/	[ɣ]	063A	غم
ف	f	/f/	[f]	0641	فوت
ک	k	/k/	[k]	06A9	کم
گ	g	/g/	[g]	06AF	گم

ل	l	/l/	[l]	0644	لب
م	m	/m/	[m]	0645	من
ن	n	/n/	[n]	0646	نان
و	v	/v/	[v] [u]	0646	وام
ی	y	/y/	[i:] [j]	06CC	آری
أ	I	/a?/	[æ?]	0623	تأکید
ؤ	U	/o?/	[o?]	0624	مؤمن
ئ	Y	/y?/	[i?]	0626	رئیس
◌َ	a	/a/	[æ]	064E	بَر
◌ِ	e	/e/	[e]	0650	سِر
◌ُ	o	/o/	[o]	064F	بُرد
◌ّ	W	-	-	0651	Gemination
◌ً	N	/an/	[æn]	064B	لطفاً
Numbers and punctuation marks					
۱	1	-	yek	06F1	
۲	2	-	dow	06F2	
۳	3	-	se	06F3	
٤	4	-	chahAr	06F4	
٥	5	-	panj	06F5	
٦	6	-	shesh	06F6	
۷	7	-	haft	06F7	
۸	8	-	hasht	06F8	
۹	9	-	noh	06F9	
۰	0	-	dah	06F0	
(	(	-	parAntez bAz	0028	
)	)	-	parAntez baste	0029	
«	«	-	giyume bAz	00AB	
»	»	-	giyume baste	00BB	
:	:	-	do noqte	003A	
؛	;	-	Noqte-virgul	061B	
،	,	-	virgul	002E	
؟	?	-	noqte	061F	
“”	“”	-	?alAmate-so?Al	201C 201D	
-	-	-	naqle-qol	0640	

Appendix II: The Transliteration function in VBA

```
Option Compare Binary
Function Transliteration (fldIn As Variant) As String
Dim strOut, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10 As String
Dim s11, s12, s13, s14, s15, s16, s17, s18, s19, s20 As String
Dim s21, s22, s23, s24, s25, s26, s27, s28, s29, s30 As String
Dim s31, s32, s33, s34, s35, s36, s37, s38, s39, s40 As String
Dim s41, s42, s43, s44, s45, s46, s47, s48, s49, s50 As String
Dim s51, s52, s53, s54, s55, s56, s57, s58, s59, s60 As String
Dim s61, s62, s63, s64, s65, s66, s67, s68, s69, s70 As String
Dim s71, s72, s73, s74, s75, s76, s77, s78, s79, s80 As String

strIn = Nz(fldIn)
s1 = Replace(strIn, ChrW(&H622), "O") 'alef madde
s2 = Replace(s1, ChrW(&H627), "A") 'alef
s3 = Replace(s2, ChrW(&H628), "b") 'be
s4 = Replace(s3, ChrW(&H67E), "p") 'pe
s5 = Replace(s4, ChrW(&H62A), "t") 'te
s6 = Replace(s5, ChrW(&H62B), "c") 'se
s7 = Replace(s6, ChrW(&H62C), "j") 'jim
s8 = Replace(s7, ChrW(&H686), "K") 'che
s9 = Replace(s8, ChrW(&H62D), "H") 'he
s10 = Replace(s9, ChrW(&H62E), "x") 'khe
s11 = Replace(s10, ChrW(&H62F), "d") 'dal
s12 = Replace(s11, ChrW(&H630), "M") 'zal
s13 = Replace(s12, ChrW(&H631), "r") 're
s14 = Replace(s13, ChrW(&H632), "z") 'ze
s15 = Replace(s14, ChrW(&H698), "J") 'zhe
s16 = Replace(s15, ChrW(&H633), "s") 'sin
s17 = Replace(s16, ChrW(&H634), "S") 'shin
s18 = Replace(s17, ChrW(&H635), "C") 'sad
s19 = Replace(s18, ChrW(&H636), "Z") 'zad
s20 = Replace(s19, ChrW(&H637), "T") 'ta
s21 = Replace(s20, ChrW(&H638), "D") 'za
s22 = Replace(s21, ChrW(&H639), "E") 'eyn
s23 = Replace(s22, ChrW(&H63A), "G") 'gheyn
s24 = Replace(s23, ChrW(&H641), "f") 'fe
s25 = Replace(s24, ChrW(&H642), "q") 'ghaf
s26 = Replace(s25, ChrW(&H6A9), "k") 'kaf
s27 = Replace(s26, ChrW(&H6AF), "g") 'gaf
s28 = Replace(s27, ChrW(&H644), "l") 'lam
s29 = Replace(s28, ChrW(&H645), "m") 'mim
s30 = Replace(s29, ChrW(&H646), "n") 'nun
s31 = Replace(s30, ChrW(&H648), "v") 'vav
s32 = Replace(s31, ChrW(&H647), "h") 'he
s33 = Replace(s32, ChrW(&H6CC), "y") 'ye
s34 = Replace(s33, ChrW(&H64A), "y") 'Arabic ye
s35 = Replace(s34, ChrW(&H623), "I") 'alef zebar hamze
s36 = Replace(s35, ChrW(&H625), "L") 'alef zir hamze
s37 = Replace(s36, ChrW(&H626), "Y") 'ye hamze
s38 = Replace(s37, ChrW(&H624), "U") 'vav hamze
s39 = Replace(s38, ChrW(&H621), "R") 'hamze
s40 = Replace(s39, ChrW(&H64B), "N") 'tanvin -an
```

```
s41 = Replace(s40, ChrW(&H651), "W") 'tashdid
s42 = Replace(s41, ChrW(&H64D), "eN") 'tanvin -en
s43 = Replace(s42, ChrW(&H6C0), "F") 'he hamze
s44 = Replace(s43, ChrW(&H629), "P") 'ta tanis
s45 = Replace(s44, ChrW(&H643), "k") 'Arabic kaf
s46 = Replace(s45, ChrW(&H652), "Q") 'sokun
s47 = Replace(s46, ChrW(&H64E), "a") 'fathe
s48 = Replace(s47, ChrW(&H650), "e") 'kasre
s49 = Replace(s48, ChrW(&H64F), "o") 'zamme
s50 = Replace(s49, ChrW(&H6F1), "1") '1
s51 = Replace(s50, ChrW(&H6F2), "2") '1
s52 = Replace(s51, ChrW(&H6F3), "3") '1
s53 = Replace(s52, ChrW(&H6F4), "4") '1
s54 = Replace(s53, ChrW(&H6F5), "5") '1
s55 = Replace(s54, ChrW(&H6F6), "6") '1
s56 = Replace(s55, ChrW(&H6F7), "7") '1
s57 = Replace(s56, ChrW(&H6F8), "8") '1
s58 = Replace(s57, ChrW(&H6F9), "9") '1
s59 = Replace(s58, ChrW(&H6F0), "0") '1
s60 = Replace(s59, ChrW(&H28), "(") '(
s61 = Replace(s60, ChrW(&H29), ")") ')
s62 = Replace(s61, ChrW(&HAB), "((") '((
s63 = Replace(s62, ChrW(&HBB), "))") '))
s64 = Replace(s63, ChrW(&H3A), ":") ':
s65 = Replace(s64, ChrW(&H61B), ";") ';
s66 = Replace(s65, ChrW(&H60C), ",") ',
s67 = Replace(s66, ChrW(&H2E), ".") '.
s68 = Replace(s67, ChrW(&H61F), "?") '?
s69 = Replace(s68, ChrW(&H21), "!") '!
s70 = Replace(s69, ChrW(&H23), "#") '#
s71 = Replace(s70, ChrW(&H40), "@") '@
s72 = Replace(s71, ChrW(&H24), "$") '$
s73 = Replace(s72, ChrW(&H25), "^") '^
s74 = Replace(s73, ChrW(&H26), "&") '&
s75 = Replace(s74, ChrW(&H2A), "*") '*
s76 = Replace(s75, ChrW(&H2B), "+") '+
s77 = Replace(s76, ChrW(&H2D), "_") '_
s78 = Replace(s77, ChrW(&H640), "") 'keshide
s79 = Replace(s78, ChrW(&H200D), " ") 'ZWJ
s80 = Replace(s79, ChrW(&H64C), "") 'shift-w sign
        strOut = Replace(s80, ChrW(&H200C), "-") 'ZWNJ
        Transliteration = strOut
End Function
```