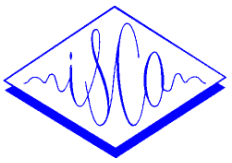

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

A RULE-BASED APPROACH TO FARSI LANGUAGE TEXT-TO-PHONEME CONVERSION

M. R. Sadigh,[1,2] H. Sheikhzadeh,[1] M. R. Jahangir[1,2] and A. Farzan[2]

[1]Dept. of Elect. Eng., Amirkabir Univ. of Tech. (Tehran Polytechnic), Tehran and [2]PSA co. Ltd., Tehran.

Email: hsheikh@cic.aku.ac.ir

ABSTRACT

A conversion from orthographic (written) form to a phonetic transcription is the first stage in a text-to-speech system. In this study, algorithms are presented to facilitate the text-to-phoneme (TTP) conversion for the Farsi language. Using a lexicon of about 15000 base morphemes, word formation rules are investigated and implemented. Moreover, a word segmentation of the written sentence has to be done prior to any phonetic transcription of the text. Due to special form of Farsi orthography, the word segmentation process is a complicated one. To solve the problem, a fast and on-line algorithm and a more complicated off-line algorithm are presented. The overall performance of the TTP conversion is evaluated to be more than 90%.

1. INTRODUCTION

The first stage in a text-to-speech (TTS) system has to be a conversion from a written text (orthography) to a phoneme sequence. While this text-to-phoneme (TTP) conversion might be a relatively trivial process for some languages (like English), it is a rather complicated task for some other languages like Farsi (Persian). There are a few reasons for this complication in Farsi: 1) Like English there are inconsistencies in Farsi orthography, a single letter can stand for more than one sound, several letters can stand for the same sound, and there are letters that are written but are not pronounced. What complicates the problem is the issue of "missing vowels" in Farsi. The three short vowels of /@/, /e/, and /o/ are not written at all and diacritics are not used either. Also the long vowels /i/ and /u/ can be confused with /y/ and /v/ phonemes, respectively. 2) A TTP conversion starts with parsing a written sentence to obtain words. This is easily accomplished in many languages by using space characters and punctuations between words. However, in Farsi script most letters can be attached to others, making a compound word consisting of a few words without spaces in between. Also, a word ending in a "discrete" letter (defined in Section 4.1) may be followed by another word without a space between the two. In some situations, an additional space in the middle of a word is not regarded as a serious mistake. As a result, one cannot only rely on space characters for parsing. 3) Words following each other affect the phonetic content. The most important case is an additional (unwritten) phoneme /e/ that is inserted between two words in many situations, e.g. when an adjective follows a noun, and between nouns in possessive form. Without the proper use of the additional phoneme /e/, uttered sentences can be hard to understand. While similar scripts are employed in Arabic and Farsi, diacritical marks used in Arabic scrip reduce the TTP conversion difficulties by a large extent, however they also increase the orthographic complexity for computer manipulations.

In this research, some of the many issues involved in the process of automatic TTP conversion of Farsi are addressed. The approach is a rule-based one, using a lexicon of base morpheme entries. Since there are many compound words in Farsi, research has been conducted to implement word formation rules to obtain the phonetic transcription of the words using the lexicon. Among the important implemented cases are rules for verbs, most prefixes and suffixes, and compound words. The early results show a success rate of more than 90% for our large vocabulary TTS system [1]. In order to parse a Farsi sentence to its comprising meaningful words, we have innovated and implemented recursive algorithms, which rely both on the lexicon and the written letter types. The success rate of the algorithm for artificially manipulated written sentences (by adding extra spaces into the words) is more than 90%. Of course the algorithm is computationally heavy and has to be run offline.

2. THE LEXICON

In the rule-based approach chosen for the TTP conversion, the lexicon plays a crucial role. The lexicon consists of about 15000 base morphemes at this stage, and will be extended later according to the application. Each entry is comprised of the

orthographic form of the word, its phonetic transcription, and its grammatical function. For words with multiple possible grammatical functions, separate entries are included for each function. While using only base morpheme entries helps to limits the lexicon size, it necessitates the investigation and development of word formation rules for compound words using only the base morphemes. Since the process of TTP conversion involves several refers to the lexicon for different tasks (as it will be discussed), the lexicon search has to be a fast process. Thus the lexicon is sorted for the orthographic entries, and a table of indices is made that contains the addresses to the lexicon entries based on the first two letters of the orthographic form. A fast binary search is then employed to look up the entries. Application of the lexicon is not limited to only extraction of the phonetic transcription of the words. As it will be discussed, in the process of word segmentation and syntax and semantic analyses, the lexicon has to be used.

3. WORD FORMATION RULES

To be able to expand the use of lexicon from the base morphemes to most of the language words, one has to investigate and implement the rules of forming words out of base morphemes by combining multiple base morphemes and affixes. The most important classes of word formation rules in Farsi are those of verbs and affixes.

Verb Word Formation Rules

Farsi verbs (in their written forms) are made of three segments: the initial segment (ISG), the verb base morpheme (VBM), and the last segment (LSG)[2]. Either of the ISG and LSG portions might not exist for some forms of verbs. There are two VBM types: the past tense and the present/future tense base. The LSG portion shows the first, second and third person objects, and plural/singular forms of the verb (altogether six different types). The LSG portion identifies important issues such as the verb tense and form, and positive/negative forms. To complicate the matters more, the ISG and LSG portions affect and change the phonetic transcription of the VBM too. The approach to solve the verb TTP conversion in this study is to make a list of rules (of phonetic transcription of the whole verb) for all possible ISG and LSG items. A total of 29 such rules are identified. Given a verb (in its orthographic form), an exhaustive search is started to match the first few letters of the verb with a valid ISG entry (in orthographic form). Then the last letters are compared to LSG forms to find a match or multiple matches. Finally the remaining letters after the identified ISG and before the identified LSG(s) are searched to match a base morpheme (of verb type) in the lexicon. Once a valid sequence of ISG+VBM+LSG is found, the phonetic transcription starts based on the lexicon entries and ISG and LSG concatenation rules. For compound verbs, the base morphemes are included in the lexicon and the same word formation rules are applied to them. The performance of the suggested approach is evaluated using an inventory of about 1000 Farsi verbs of different forms and tenses (that have their VBMs included in the lexicon), and the success rate has been about 98%.

Affix Analysis

Similar to other languages, prefixes and suffixes are vastly used in Farsi. As multiple prefixes (or suffixes) can be used in a single word, we chose to employ an algorithm to properly identify the affixes. Similar to the verb formation rules, the algorithm recursively searches for a valid prefix and suffix at the beginning and end of the word. Once the sequences of found prefixes, base morpheme and suffixes are all valid words, the search ends. At this stage, about 30 suffixes and 5 prefixes are included in the lexicon, and it seems that for Farsi one needs much more suffixes than prefixes for a successful affix analysis. Affixes that are rarely used are not included in the list since they slow down the search process by enlarging the search space. In the case of such affixes, it is preferred to include the words together with the affixes in the lexicon.

4. WORD SEGMENTATION ALGORITHMS

Before any other processing of the written forms, it is essential to segment the sentence to its comprising words. We have developed two different segmentation methods. The first one is an on-line, real-time and simple method, and the second one is an off-line, complicated method with a much better performance.

4.1. On-Line Word Segmentation

In this method, punctuation marks, and spaces are the primary delimiters for word segmentation. Moreover, some of the rules governing the orthographic forms of the Farsi letters can be exploited in the process. A brief introduction to the rules follows. The 32 letters of the Farsi alphabet can be classified into four categories: 1) 7 letters are "*discrete*" letters; they are never attached to their following letters in a word. Examples include Farsi equivalents of letters D, R, Z, and V. 2) 22 letters can appear in two forms: a "*connected* or *small*" form when located in the middle of a word and attached to the next letter, and a "*disconnected* or *capital*" form when positioned at the end of a word (whether followed by a space or not). Examples are Farsi equivalents of letters B, P, S, and T. 3) 2 letters have only one capital form, whether located in the middle of a word and attached to the next letter, or positioned at the end of a word. 4) One letter (Farsi equivalent of letter H) can appear in three forms of capital (at the end of a word and following a group-2 letter) or small (in the middle of a word) or discrete (at the end of a word and following a discrete letter) according to its position in the word. To exploit the above rules in the word segmentation algorithm, it is noticed that obviously a capital form is most probably located at a word boundary, unless an additional space is (falsely) typed after it. On the other hand, a small form can never be at a word boundary unless a space character is falsely omitted. Using the above rules helps to reduce segmentation errors due to false punctuation, and specially omission of spaces. As a result, a simple and fast word segmentation algorithm is implemented. However, if space omissions (specially after *discrete* letters or group-3 letters) or space insertions happen frequently, then the approach will fail. Since such situations are actually observed in typed Farsi texts, we present an off-line and more complicated solution to the problem.

4.2. OFF-LINE SEGMENTATION

In this approach, it is tried to imitate what a human reader does to segment and read a Farsi sentence with lots of extra spaces and spaces deletions. As a reader mostly relies on his/her knowledge of the language words, the algorithm also tries to recognize valid words using the lexicon and the word formation rules. The algorithm presented here initially assumes that (extra) space insertion is allowed but no space deletion happens. This is a reasonable assumption regarding the actual Farsi typed texts. The issue of space deletion will be dealt with at the end of this section. Since extra spaces are allowed, a word might have been broken into a few parts. The complication occurs when segments of a word make a valid and meaningful word with segments of the next word. This can easily occur in Farsi and as a result, a sentence can have multiple word segmentations [2]. Human readers obviously rely on their syntax and semantic knowledge of the language to avoid ambiguities, however a TTS system should handle such situations with proper algorithms.

In the algorithm described here, a sentence is first segmented into a few "blocks". Each block is a collection of letters separated from the neighboring block by spaces and punctuation marks. Since no space deletion happens, a block then would be either a word or a part of it, and no block contains two or more words or their segments. A recursive routine augments a block with neighboring blocks and tests the combination against the words in the lexicon, using all the word formation rules. The augmentation stops once a valid word is found. A pseudo-code representation of the recursive segmentation algorithm follows.

Initialization:
Given a sentence with N blocks,
Find_Valid_Segments([1,N])
End

Recursive Routine:
```
Find_Valid_Segments([P,N])
  If P >  N then
    Print segmented words, End.
  Else
    For I =  P  to  N
      If  Is_A_Valid_Word([P,I]) then
        Declare the [P,I] blocks as a valid word.
        Find_Valid_Segments([I+1,N])
      End If
    End For
  End If
```

The routine Find_Valid_Segments([P,N]) finds all the valid word segmentations for a part of the sentence starting from block P to the end of the sentence (blocks [P,N]). As shown, for each block I

(I >= P) it checks whether the blocks [P,I] consist a valid word or not. This is accomplished by Is_A_Valid_Word([P,I]), that employs the lexicon and all the word formation rules. Once a valid word is found, it is declared as a segmented word, and the routine is called recursively for the next block, I+1.

The algorithm stated above finds all possible valid word segmentations of a sentence. To choose the best segmentation however, the algorithm selects the one that is more consistent with existing punctuation marks and the capital letter patterns. We are now developing and testing reasonable cost function for this task. As a result, the algorithm also handles the complicated problem of compound words in Farsi.

Algorithm Extension

Up to now, the algorithm assumed that there might be space insertions but no space deletions are allowed. If space deletions are allowed, the block might contain a part of a word, a word, or multiple words. In such cases, the same algorithm can be used if the block definition is changed. The block then is defined as a single letter of the sentence. Obviously the algorithm would then become computationally heavy.

Another issue is that the segmentation algorithm relies on the lexicon (together with word formation rules) to find a word. Problem then arises when a word is encountered that is not included in the lexicon (an "unseen" word). There are a few methods for handling this problem. One is to introduce a cost function for the word segmentation process. Each unseen word would increase the cost function, and among all possible segmentations, the one with the least cost would be selected.

5. ADDITIONAL PHONEME /e/ PROBLEM

Regarding the additional phoneme /e/ issue (that is added between adjectives and nouns, and between nouns in possessive form), Farsi word are classified into three categories:

1. Words that are always followed by the additional /e/ phoneme (like Farsi word /b@ray/)
2. Words that are never followed by an additional /e/ phoneme. Examples are words ending in phonemes /a/ or /u/.
3. Words that might be followed by an additional /e/ phoneme, depending on their grammatical role in .the sentence.

Deciding about groups 1 and 2 is simple, however for group 3 we added an additional /e/ between more than ten types of word sequences like noun-noun, noun-adjective, adjective-adjective, noun-pronoun, noun-number, and a few more. Experiments using more than 200 Farsi sentences showed that the algorithm performed correctly at 85% of cases. Thus syntax and semantic analyses are necessary to correctly add the additional phoneme /e/ between Farsi words.

4. CONCLOSION AND FUTURE WORK

In this research, a rule-based approach to Farsi language TTP conversion problem was introduced. As discussed before, the TTP conversion consists a crucial and non-trivial part of a TTS system. There are still a few important issues that need to be carefully considered. The first issue is the additional phoneme /e/ problem. As discussed before, a proper solution of the issue needs syntax and semantic analyses. The second issue is the problems related to the words that are not included in the lexicon. Both the segmentation and TTP conversion processes fail when such words are encountered. As suggested at the end of Section 4.2, proper solutions to the problem have to be investigated both for segmentation and TTP conversion. Another future direction of this research is to work towards a more efficient implementation of the segmentation algorithm.

Acknowledgement

This work was done in a national research project on Farsi speech processing, sponsored by the National Research Council of Iran

5. REFERENCES

1. H. Sheikhzadeh, A. Eshkevari, M. Khayatian, R. Sadigh and S. M. Ahadi, "Farsi Language Prosodic Structure, Research and Implementation Using a Speech Synthesizer", *Proceedings of Eurospeech*, Budapest, Sept. 1999.
2. A. Soltani Gord Faramarzi, From Word to Text (in Farsi), Forough Danesh Publishing, Tehran, 1981.