Contextual Analysis for Middle Eastern Languages with Hidden Markov Models

Kazem Taghva

Department of Computer Science
University of Nevada, Las Vegas
Las Vegas, NV

Abstract

Displaying a document in Middle Eastern languages requires contextual analysis due to different presentational forms for each character of the alphabet. The words of the document will be formed by the joining of the correct positional glyphs representing corresponding presentational forms of the characters. A set of rules defines the joining of the glyphs. As usual, these rules vary from language to language and are subject to interpretation by the software developers.

In this paper, we propose a machine learning approach for contextual analysis based on the first order Hidden Markov Model. We will design and build a model for the Farsi language to exhibit this technology. The Farsi model achieves 94% accuracy with the training based on a short list of 89 Farsi vocabularies consisting of 2780 Farsi characters.

The experiment can be easily extended to many languages including Arabic, Urdu, and Sindhi. Furthermore, the advantage of this approach is that the same software can be used to perform contextual analysis without coding complex rules for each specific language. Of particular interest is that the languages with fewer speakers can have greater representation on the web, since they are typically ignored by software developers due to lack of financial incentives.

Index Terms

Unicode, Contextual Analysis, Hidden Markov Models, Big Data, Middle Eastern Languages, Farsi, Arabic, data science, machine learning, artificial intelligence

1. Introduction

One of the main objectives of the Unicode is to provide a setting that non-English documents can be easily created and displayed on modern electronic devices such as laptops and cellular phones. Consequently, this encoding has led to development of many software tools for text editing, font design, storage, and management of data in foreign languages. For commercial reasons, the languages with high speaking populations and large economies have enjoyed much more rapid advancement in Unicode based technologies. On the other hand, less spoken languages such as Pushtu is barely given attention. According to [11], approximately 40 to 60 million people speak Pushtu worldwide.

Many Unicode based technologies are based on proprietary and patented methods and thus are not available to the general open source software developers' communities. For example, BIT [9] does not reveal its contextual analysis algorithm for Farsi[10]. Many software engineers need to redevelop new methods to implement tools to mimic these commercial technologies. The new contextual analysis for Farsi developed by Moshfeghi in Iran Telecommunication Research Center is an example of these kinds of efforts [10].

The Unicode also introduces a challenge for the internationalization of any software regardless of being commercial or open source. Tim Bray [2] writes:

Whether you're doing business or academic research or public service, you have to deal with people, and these days, it's quite likely that some of the people you want to deal with come from somewhere else, and you'll sometimes want to deal with them in their own language. And if your software is unable to collect, store, and display a name, an address, or a part description in Chinese, Bengali, or Greek, there's a good chance that this could become very painful very quickly.

There are a few organizations that as a matter of principle operate in one language only (The US Department of Defense, the Acadmie Franaise) but as a proportion of the world, they shrink every year."

This internationalization is a costly effort and subject to availability of resources. As mentioned above, languages with high speaking population such as Mandarin attract a lot of the efforts. The availability of data in Unicode represents an opportunity to employ machine learning techniques to advance software internationalization and foreign text manipulation. The language translation technologies heavily use Hidden Markov Models (HMM) to improve translation accuracy [3][1].

In this paper, we propose the use of HMM for contextual analysis. In particular, we design and build a generic HMM for Farsi that can be easily adapted to other Middle Eastern languages.

In section 2, we provide some background and related work on contextual analysis. Section 3 will provide a brief introduction to first order HMM. In section 4, we describe the design and implementation of our HMM for Farsi contextual analysis. The training and testing of HMM will be explained in section 5. Finally, section 6 describes our conclusion and proposes future work.

2. Background

In 2002, the Center for Intelligent Information Retrieval at the University of Massachusetts, Amherst, held a workshop on Challenges in Information Retrieval and Language Model [7]. The premise of this workshop was to promote the use of the Language Model technology for various natural languages. The aim is to use the same software for indexing and retrieval regardless of the language. It was pointed out that, by using training materials such as document collections, we can automatically build retrieval engines for all languages. This report was one of the reasons that we decided to start a couple of projects on Farsi and Arabic [16][19].

Consequently, these projects led to developments of the two widely used Farsi and Arabic Stemmers [15][17]. One of the difficulties we had was the lack of technologies for input and display of Farsi and Arabic documents [19]. For example, we needed an input/display method that would allow us to enter Farsi query words in a Latin-based operating system without any special software or hardware. It was further necessary to have a standard character encoding for text representation and searching. At the time, we developed a system that provides the following capabilities:

- a web-browser based keyboard applet for input
- if the web-browser has the ability to process and display Unicode content, it will be used
- if the browser cannot display Unicode content, an auxiliary process will be invoked to render the Unicode content into a portable bitmap image with associated HTML to display the image in the browser.

Another area of difficulty that we encountered is that the presence of white space used to separate words in the document is dependent on the display geometry of the glyphs. Since Farsi and Arabic are written using a cursive form, each character can have up to four different display glyphs. These glyphs represent the four different presentation forms:

isolated:	the standalone character
initial:	the character at the beginning of a word
medial:	the character in the middle of a word
final:	the character at the end of a word

We found that depending on the amount of trailing white space following a final form glyph, a space character may or may not be found in the text. This situation came to light when our subject matter experts were developing our test queries. We found that since the glyphs used to display the final form of characters had very little trailing space, they were manually adding space characters to improve the look of the displayed queries.

2.1 Keyboard Applet

The keyboard applet was written in java script. The applet displays a Farsi keyboard image with the ability to enter characters from both the keyboard and mouse. The applet also handles character display conversion and joining of the input data.

The keyboard layout is based on the ISIRI 2901:1994 standard layout as documented in an email by Pournader [12]. Figure 1 shows the keyboard applet being used to define our test queries for search and retrieval.

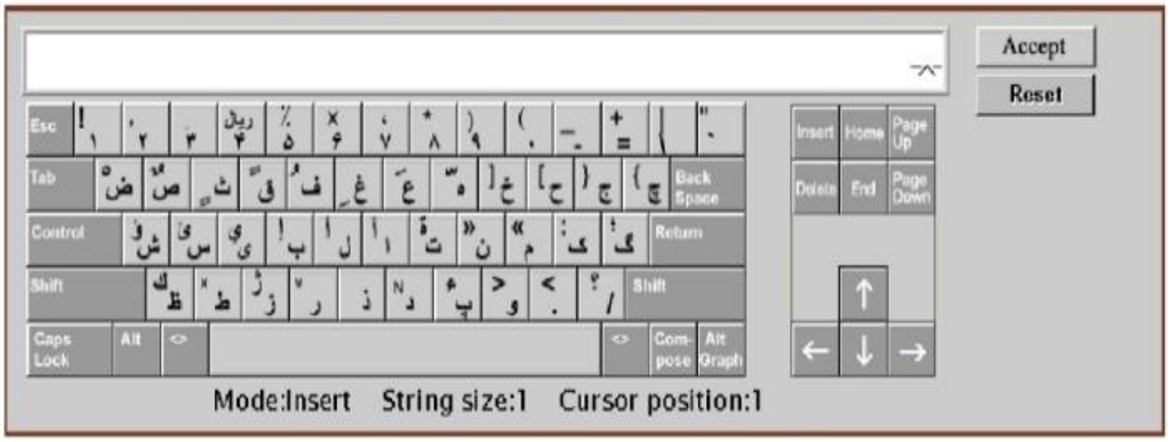
Display of the input data is normally performed by using the preloaded glyph images. However, if a character has not been preloaded, it can be generated on the fly. Most of the time, these generated characters are ``compound" characters. Farsi (and other Arabic script languages) may use "compound" characters which are a combination of two or more separate characters. For example, the rightmost character of خرما
, the Farsi word for ``date" (that is,the fruit), is a combination of a خ with a damma.

The complications associated with our work on Farsi and Arabic convinced us that we need to develop generic machine learning tools if we want to develop display and search technologies for most of the Middle Eastern languages. In the next few sections, we will offer a solution to contextual analysis to display the correct presentational forms of characters.

3. Hidden Markov Model

An HMM is a finite state automaton with probabilistic transitions and symbol emissions [13][14]. An HMM
consists of:

A set of states $S = \{s_1, s_2, \ldots, s_n\}$.
An emission vocabulary $V = \{w_1, w_2, \ldots, w_m\}$.
Probability distributions over emission symbols where the probability that a state s emits symbol w is given by $P(w|s)$. This is denoted by matrix B.
Probability distributions over the set of possible outgoing transitions. The probability of moving



from state s_i to s_j is given by $P(s_j|s_i)$. This is denoted by matrix A. A subset of the states that are considered start states, and to each of these is associated an initial probability that the state will be a start state. This is denoted by Π.

Figure 1. Example Use Of The Keyboard Applet

As an example, consider the widely used HMM [21] that decodes weather states based on a friend's activities. Assume there are only two states of weather: ***Sunny***, ***Rainy***. Also assume there are only three activities: ***Walking***, ***Shopping***, ***Cleaning***.

You regularly call your friend who lives in another city to find out about his activity and the weather status. He may respond by saying ``I am cleaning and it is rainy", or ``I am shopping and it is sunny". If you collect a good number of these weather states and activities, you then can summarize your data as the HMM shown in Figure 3.

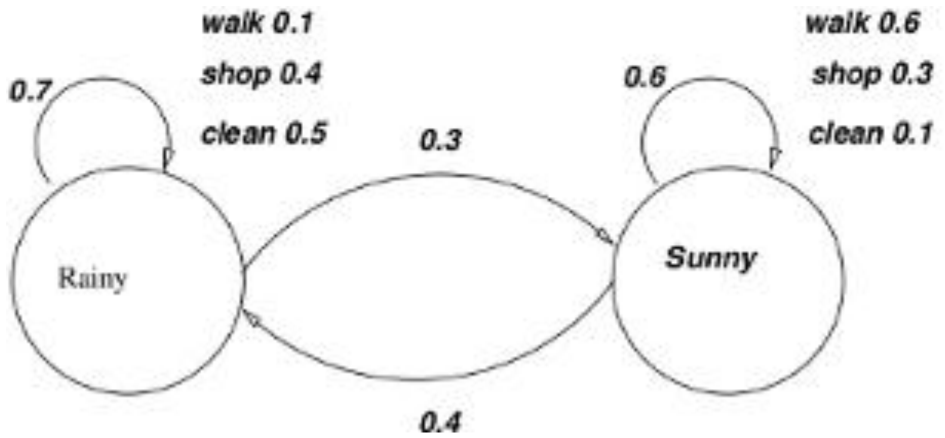


Fig:Figure 1 An Hmm For Activities And Weather

This HMM states that on rainy days, your friend walks 10% of the days while on sunny days, he walks 60%. The statistics associated with this HMM is obtained by simply counting the activities on rainy and sunny days.

You also notice arrows from states to states that keeps track of weather changes. For example, our HMM reflects the fact that on a rainy day, there is a 70% chance of rain next day while 30% chance of sunshine.

In addition, one can keep track of how many days in the data are sunny or rainy. This will be the initial probabilities. Formally these statistics are calculated by Maximum Likelihood Estimates (MLE). Formally, transition probabilities are estimated as:

$$P(s_i, s_j) = \frac{Number\ of\ transitions\ from\ s_i to\ s_j}{Total\ number\ of\ transiotions\ out\ of\ s_i}$$

The emission probabilities are estimated with Maximum Likelihood supplemented by smoothing. Smoothing is required because Maximum Likelihood Estimation will sometimes assign a zero probability to unseen emission-state combinations.

Prior to smoothing, emission probabilities are estimated by:

$$\mathrm{P(w|s)_{ml}} = \frac{\text{Number of times w is emitted by s}}{\text{Total number of symbols emitted by s}}$$

The most interesting part of an HMM is the decoding aspect. We may be told that our friend's activities for the last four days were cleaning, cleaning, shopping, cleaning and we want to know what the weather patterns were for those four days. This essentially translate to finding a sequence of four states $s_1 s_2 s_3 s_4$ that maximizes probability:

$$P(s_1 s_2 s_3 s_4 \mid \textbf{cleaning cleaning shopping cleaning})$$

This amounts to choosing the highest probability among 16 choices for $s_1 s_2 s_3 s_4$. This is computationally very expensive as the number of states and symbols increases. The solution is given by the Viterbi algorithm that finds an optimal path using dynamic programming [14]. The Algorithm 1 is a modification of the pseudo code from [21].

In the next section we will describe the design and implementation of an HMM for Farsi contextual analysis.

4. Farsi Hidden Markov Model

The Farsi HMM is very similar to the example of HMM described in the previous section. The HMM has a state for each presentation form of Farsi alphabet. Also the HMM has a vocabulary of size 32, one for each character in Farsi alphabet. A simple calculation reveals that the Farsi HMM should have 128 states and 32 vocabulary. The HMM has fewer than 128 states since some of the characters do not have four presentational form. For example, there are only two states for the character ا (alef), as there are no medial or initial form for this character.

Algorithm 1: Viterbi Algorithm

```
Algorithm 1: Viterbi Algorithm
        Data:   Given K states and M vocabularies, and a
        sequence of vocabularies $Y = w_1 w_2 \dots w\_n$
Result:   The most likely state sequence
      $R = r_1 r_2 \dots r_n$ that maximizes the above probability
Function Viterbi ( V,S, Π, Y, A,B) : X
for each state $s_i$ do
    $T_1[i,1] = \Pi_i * B_{iw_1}$;
    $T_2[i,1] = 0$;
end

for i = 2,3, ... , n do
    for each state $s_j$ do
        $T_1[j,i] = \max_k \left(T_1[k,i-1] * A_{kj} * B_{jw_i}\right)$;
        $T_2[j,i] = argmax_k \left(T_1[k,i-1] * A_{kj} * B_{jw_i}\right)$;
    end
end
```

```
$z_n = argmax_k(T_1[k,n])$
$r_n = sz_n$
for i = n , n-1 , ... , 2 do
    $z_{i-1} = T_2[z_i, 1];$
    $r_{i-1} = s_{z_i} - 1;$
end
Return R
```

As an example, suppose we want to type the word **شغال** , in English jackal. On the keyboard, we type four isolated characters ش , غ , ا , and ل . The HMM should decode these four characters as initial, medial, final, and isolated, respectively. In other words, the sequence of the four isolated characters (or vocabulary in HMM terminology) should be decoded in the four states as shown in Figure 3.

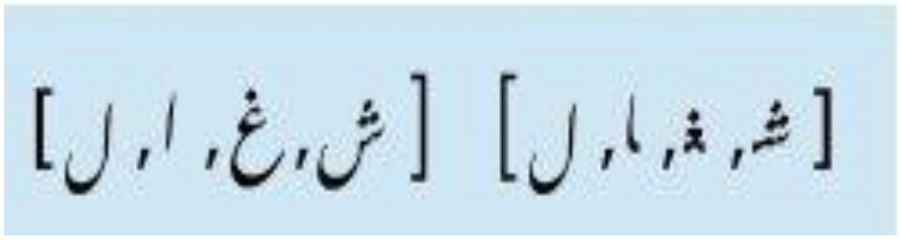


Figure 2 The Four Isolated Characters On The Left Are Vocabularies While The Four Characters On The Right Are The States Of Hmm

The part of the HMM as displayed in Figure 4. Shows how Viterbi algorithm takes the path to decode the correct form of the characters by choosing the appropriate states. As we observe, there are four states for the character غrepresenting the four shapes of this character. We also observe that there are only two states for the character ا , as there are no medial or initial form for this character.

A typical implementation of HMM adds states and vocabularies as being trained [20]. The training is done by providing pairs of the form $([w_1 w_2 \ldots . w_n], [s_1 s_2 \ldots s_n])$ similar to the [vocabularies, states] sequences as shown in Figure 3.

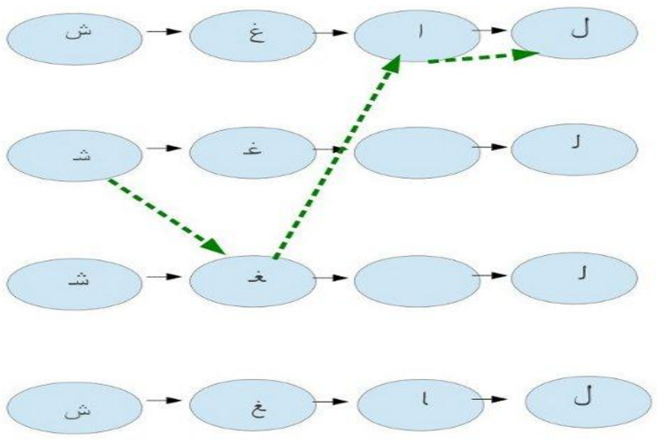


Figure 3 Glyphs Chosen By HMM

5. Training and Testing of HMM

We trained the HMM with 89 words (2780 characters) chosen from the list of the frequent words from Kayhan newspaper published in 2005 [4]. There are over 10,000 words in this collection. We limited the training to this short list to save time. The list of these words are shown in Figure 5.

The test data is a small number of words selected randomly from a small dictionary and shown in Figure 6. This list
contains 32 words (350 characters). The training file contains pairs of words separated by a vertical bar. The first word is the isolated form and the second word is the correct presentational form of the word. We read the file one line at a time and submit the two words for trainingas seen in the following Ruby code:

```
f = File.open("./training-data")
farsi.train([" "],[" "])\\
f.each do | line |
seq1,seq2 = line.chomp.split(/s*|s*/)
        farsi.train(seq1.split(" "),
        seq2.split(" "))
end
```

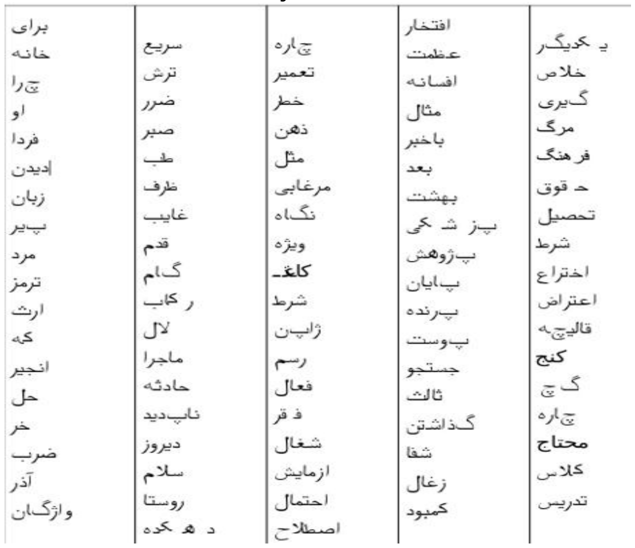
As it is seen, we have added a blank vocabulary and state to our HMM. The HMM adds vocabulary and states as a part of the training. The HMM has 32 vocabularies and 74 states. It is anticipated that the HMM will have more states as the size of the training data increases.

The test correctly decoded 94% of all the characters. Most of the mistakes are due to the fact that the HMM has not seen enough combination examples of characters. For example, in the word آتش , the initial form of ت was not decoded correctly. A closer examination of the training data reveals that there are no occurence of تش in the set. Similarly, there are other errors of this form such as the initial form of ن in the word ترانه . There are also a few errors attributed to the double combination of the character ی as in طلایی . We believe most of these errors will be corrected with a larger training sets.

6. Conclusion and Future Work

In this paper, We have presented a machine learning approach to the contextual analysis of script languages. It is shown that an ergodic HMM can be easily trained to automatically decode presentational forms of the script languages.

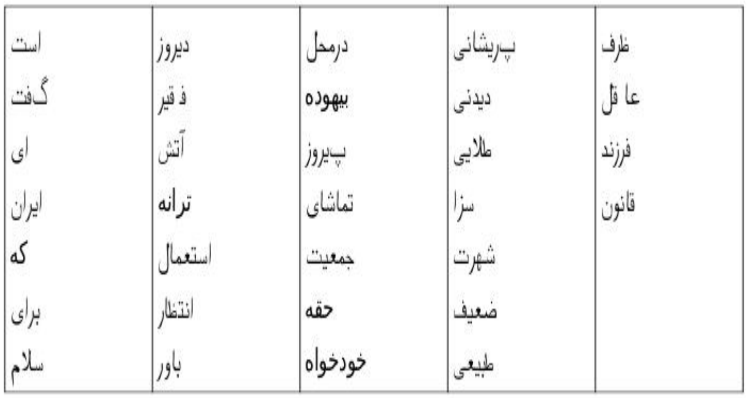
Although the paper is developed based on Farsi, it can be easily extended to other middle eastern languages. Further training and research in this area can improve the character accuracy.
A successful program for contextual analysis may have to include a list of exceptional words that do not fall into the normal combination of the characters. It is also important to notice that most of the Arabic and Farsi type setting technologies such as ArabTex [8] or FarsiTex [5] have problems with contextual analysis. This is mainly due to the fact that it is practically impossible to devise an algorithm that has 100% accuracy for tasks associated with natura languages.



برای			افتخار	
خانه	سریع	چاره	عظمت	ی کدیگر
چرا	ترش	تعمیر	افسانه	خلاص
او	ضرر	خطر	مثال	گیری
فردا	صبر	ذهن	باخبر	مرگ
ادیدن	طب	مثل	بعد	فرهنگ
زبان	ظرف	مرغابی	بهشت	ح قوق
پیر	غایب	نگاه	پز ش کی	تحصیل
مرد	قدم	ویژه	پژوهش	شرط
ترمز	گام	کاغذ	پایان	اختراع
ارث	ر کاب	شرط	پرنده	اعتراض
که	لال	ژاپن	پوست	قالیچه
انجیر	ماجرا	رسم	جستجو	کنج
حل	حادثه	فعال	ثالث	گ چ
خر	ناپدید	ف قر	گذاشتن	چاره
ضرب	دیروز	شغال	شفا	محتاج
آذر	سلام	ازمایش	زغال	کلاس
واژگان	روستا	احتمال	کمبود	تدریس
	د هکده	اصطلاح		

Figure 4 Top 89 Frequent Words From Kayhan 2005

Finally, a higher order HMM may also improve the contextual analysis. For example, it is shown that the second order HMM improves the hand written character recognition [6]. It may also worth mentioning that the second order HMM does not improve error detection and correction for post processing of printed documents [18]



است	دیروز	درمحل	پریشانی	ظرف
گفت	ذ قیر	بیهوده	دیدنی	عا قل
ای	آتش	پیروز	طلایی	فرزند
ایران	ترانه	تماشای	سزا	قانون
که	استعمال	جمعیت	شهرت	
برای	انتظار	حقه	ضعیف	
سلام	باور	خودخواه	طبیعی	

Figure 5. Test Data Chosen Randomly

REFERENCES

[1] Jan A. Botha and Phil Blunsom. Compositional Morphology for Word Representations and Language Modeling. In Proceedings of the 31st International Conference on Machine Learning (ICML) , Beijing, China, june 2014. *Award for best application paper*.

[2] Tim Bray. Element sets: A minimal basis for an XML query engine. In QL , 1998.

[3] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist. , 19(2):263 - 311, June 1993.

[4] Jon Dehdari. Top frequent words in Farsi, 2005. http://www.ling.ohio-state.edu/~jonsafari/.

[5] Mohammad Ghodsi and Behdad Esfahbod, 1992. http://www.farsitex.org/

[6] Y. H. Y. He. Extended Viterbi algorithm for second order hidden Markov process. In Proceedings 9th International Conference on Pattern Recognition, pages 718{720, 1988.

[7] Allan James. Challenges in information retrieval and language modeling: Report of a workshop held at the center for intelligent information retrieval, university of Massachusetts Amherst, September 2002. SIGIR Forum, 37(1):31{47, April 2003.

[8] Klaus Lagally, 2006. http://www2.informatik. uni-stuttgart.de/ivi/bs/research/Arabic.htm.

[9] Fallah Moshfeghi and Kourosh Shadsari. Design and implementation of bilingual information entrance and edit environment. technical report, Iran Telecommunication Research Center , winter, 1999.

[10] Kourosh Fallah Moshfeghi. A new algorithm for contextual analysis of Farsi characters and its implementation in java. In 17th International Unicode Conference , 2000.

[11] Herbert Penzl and Ismail Sloan. A Grammar of Pashto a Descriptive Study of the Dialect of Kandahar, Afghanistan . Ishi Press International, 2009.

[12] Roozbeh Pournader. National Iranian standard isiri 6219, information technology Persian information interchange and display mechanism, using Unicode. In Technical Report , 2005.

[13] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In PROCEEDINGS OF THE IEEE , pages 257{286, 1989.

[14] Lawrence R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267{296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[15] Kazem Taghva, Russell Beckley, and Mohammad Sadeh. A stemming algorithm for the Farsi language. In International Symposium on Information Technology: Coding and Computing (ITCC 2005), Volume 1, 4-6 April 2005, Las Vegas, Nevada, USA , pages 158{162, 2005.

[16] Kazem Taghva, Jeffrey S. Coombs, Ray Pereda, and Thomas A. Nartker. Language model-based retrieval for Farsi documents. In International Conference on Information Technology: Coding and Computing (ITCC'04), Volume 2, April 5-7, 2004, Las Vegas, Nevada, USA , pages 13{17, 2004.

[17] Kazem Taghva, Rania Elkhoury, and Jeffrey S. Coombs. Arabic stemming without A root dictionary. In International Symposium on Information Technology: Coding and Computing (ITCC 2005), Volume 1, 4-6 April 2005, Las Vegas, Nevada, USA , pages 152{157, 2005.

[18] Kazem Taghva, Srijana Poudel, and Spandana Malreddy. Post processing with first- and second-order hidden Markov models. In DRR, 2013.

[19] Kazem Taghva, Ron Young, Jeffrey Coombs, Russell Beckley, and Mohammad Sadeh. Farsi searching and display technologies. In SDIUT, 2003.

[20] David Tresner-Kirsch. Hmm ruby gem, 2009. https://github.com/dtkirsch/hmm/.

[21] Xing M. Wang. Probability bracket notation: Markov state chain projector, hidden Markov models and dynamic Bayesian networks. CoRR , abs/1212.3817, 2012.