Word Prediction in a Running Text: A Statistical Language Modeling for the Persian Language

Masood Ghayoomi
Institute for Humanities and Cultural Studies, Tehran, 14374
masood29@yahoo.com

Seyyed Mostafa Assi
Institute for Humanities and Cultural Studies, Tehran, 14374
s_m_assi@ihcs.ac.ir

Abstract

Word prediction is the problem of guessing which words are likely to follow in a given segment of a text to help a user with disabilities. As the user enters each letters of the required word, the system displays a list of the most probable words that could appear in that position. In our research we designed and implemented a word predictor for the Persian language. Three standard performance metrics were used to evaluate the system including keystroke saving, the most important one. The system achieved 57.57% saving in keystrokes.

1. Introduction

A word prediction system facilitates typing of a text for a user with physical or cognitive disabilities. As the user enters each letter of the required word, the system displays a list of the most likely completions of the partially typed word. As the user continues typing more letters, the system updates the suggestion list accordingly based on the new context. If the required word is in the list, the user can select it with a single keystroke. Then, the system tries to predict the next word. It displays a list of suggestions to the user. If he finds the next intended word, he selects it; otherwise he enters the first letter of the next word to restrict the suggestions. The process continues to complete the text.

For someone with physical disabilities, each keystroke is an effort; as a result, the prediction system saves the user's energy by reducing his physical effort and also the system assists the user in the composition of the well-formed text qualitatively and quantitatively (Fazly, 2002). Moreover, the system increases user's concentration (Klund and Novak, 2001).

Traditionally, word predictors have been built based on statistical language modeling (SLM) (Gustavii and Pederssen, 2003). SLM is based on the probability of a sequence of *n* given words (*n*-gram). A number of word prediction systems are available today for English, Swedish, and other European languages. Most of these systems have used *n*-gram language modeling.

The current research deals with the design and implementation of a word prediction system based on SLM for the Persian language.

2. Related Works

By looking back, early prediction systems mostly were developed in the 1980s. They were used as a writing assistance system for the one with disabilities. In the early systems, they only suggested the high frequency words that matched the partially typed word and ignored the entire previous context (Swiffin et al, 1985). SoothSayer is such a system. To make suggestions more appropriate, some systems look at a larger context by exploiting word bigram language model beside the word unigram. WordQ (Nantais et al, 2001; Shein et al, 2001) is a system which is developed for English. Profet (Carlberger et al, 1997a; Carlberger et al, 1997b) is a system developed in four

Proceedings of the Australasian Language Technology Workshop 2005, pages 57–63,
Sydney, Australia, December 2005.

languages: English, Norwegian, Polish and French. PAL (Predicative Adaptive Lexicon) is one of the major projects at ACSD (Applied Computer Studies Division) at Dundee University, Scotland (Booth et al, 1990). These systems have used word unigrams and bigrams; also, the systems try being adapted to the user's typing behavior by employing information on the user's recency and frequency of use.
Since there are no previous works of any developed word prediction systems for Persian, what we have done is the first attempt to design and implement a word predictor for this language. We have used the experience of the developed systems for the English and Swedish languages in our research. Details are presented in Ghayoomi (2004).

3. Some Facts about the Persian Language

Persian is a member of the Indo-European languages and has many features in common with them in morphology, syntax, the sound system, and the lexicon. Arabic is from the Semitic family and differs from Persian in many respects.
The Persian alphabet is a modified version of the Arabic alphabet. Hence it is more appropriate to the Arabic sound system and less suitable for Persian. For instance 'ز', 'ذ', 'ض' and 'ظ' are four alphabets both in Persian and Arabic, but all pronounced the same /z/ in Persian and differently in Arabic. So there is a little correspondence between Persian letters and sounds. Although some alphabets are written differently and there is no difference in their pronunciations, they make differentiations in the meanings of words.
Letters have joined or disjoined forms; i.e. based on the position that the letters appear in a word, they have different forms.
Persian writing system is right to left, the same as Arabic; but quite contrary to the European languages that have left to right writing system.
The vocabularies have been greatly influenced by Arabic and to some extent by French, and a great amount of words are borrowed from these languages.
Talking about Persian syntax, only verbs are inflected in the language. The subjective mood is widely used in it. It is an SOV language, and also a free word order language. The language does not make use of gender; not even the third person of he or she distinctions that exists in English (Assi, 2004).

4. N-gram Word Model

The task of predicting the next word can be stated as attempting to estimate the probability function P:

$$P(W_n|W_1,\ldots, W_{n-1})$$

In such a stochastic problem, we use the previous word(s), the history, to predict the next word. To give reasonable prediction to the words which appear together, we try to use Markov assumption that only the last few words affect the next word. So if we construct a model where all histories restrict the word that would appear in the next position, we have then an $(n-1)^{th}$ order Markov model or an *n*-gram word model. (Manning and Schüdze, 1999; Jurafsky and Martin, 2000)
The aim of our study is to design a word predictor that uses a unigram (n=1), bigram (n=2), and trigram (n=3) word model for Persian.

4.1. Word Prediction Algorithm

Suppose the user is typing a sentence and the following sequence has been entered so far from right to left based on Persian writing system:

$$CW_i \quad W_{i-1} \quad W_{i-2} \quad \ldots$$

where W_{i-2} and W_{i-1} are the most recently completed words and CW_i is the current word that is going to be predicted or completed. Let W be the set of all words in the lexicon that likely would appear in that

position. A statistical word prediction algorithm attempts to select the N most appropriate words from W that are likely to be the user's intended words, where N is usually between 1, 5, 9 or 10 based on the experiment done by Soede and Foulds (1986). The general approach is to estimate the probability of each candidate word, $w_i \in W$, being the user's required word in that context.

5. Methodology

5.1.Corpus

To do our research, we made a balanced corpus in different genres from 8 months of the on-line Hamshahri newspaper archive on the web. Although the corpus was small, it was a good representative for the Persian language. The corpus contained approximately 8 million tokens. After downloading the web pages, HTML pages were converted to their plain text equivalents.

5.2. Annotation

The plain text corpus was annotated. One of the annotations was replacing various spellings of a word by a selected spelling. In Persian, some words have various spellings without any changes in the meaning. To choose one spelling among various ones, the highest frequency of use was used to consider the word as the default spelling, and the various spellings were replaced by the selected one. Replacing was done manually. By doing so, the distribution of frequencies of a word with different spellings would be gathered together to assign a single frequency to the selected spelling; because of the smallness of the corpus. For example, these four words were found in the corpus: "امریکایی /?emrikāyi/", "امریکائی /?emrikā?i/", "آمریکایی /?āmrikāyi/", "آمریکائی /?āmrikā?i/". All the words mean "American". Between them, only the spelling "آمریکایی" with the highest frequency of use was selected as default and the other spellings were replaced to that.

The other annotation was removing words or phrases in the corpus from other languages or other Persian dialects comparing to the standard language that do not belong to Persian at all and not be used by native speakers of the language. Email or internet addresses were removed from the corpus. Headlines, footnotes and references in the articles were also removed.

5.3.Tokenization

After annotation, the corpus was divided into three sections: one was the training corpus that contained 6258000 tokens, and 72494 types; the other section was used as the developing corpus which contained 872450 tokens, and the last section was used as the test corpus which contained 11960 tokens.
To do the tokenization process, the training corpus was ran on NSP (N-gram Statistic Package), a program which was written in Perl in Linux (Banerjee and Pedersen, 2003), and uni-, bi-, and trigram statistics were extracted. Words with frequency of one and two regarded as Out-Of-Vocabulary (OOV) and only the most common sequence of words with the frequency of three and more were taken into account and the statistics of word uni-, bi-, and trigrams were extracted. In NSP a token is defined as a continuance sequence of characters to be space delimited alphanumeric strings or individual characters.

5.4. Solving Sparseness

Since a big corpus includes only a fraction of n-grams, increasing n makes the distribution of the events rarer. We have used the Simple Linear Interpolation (SLI) method (Manning and Schüdze, 1999) to smooth the probability distribution.

6. Implementation

6.1. The Algorithm

The architecture of our algorithm is shown in figure 1. The system we developed has four

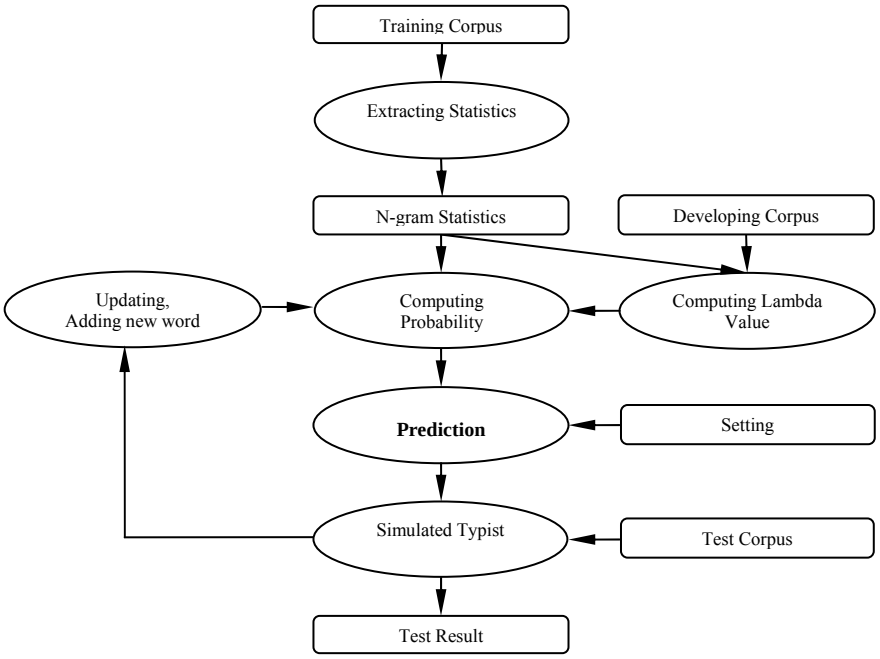

Figure 1: The architecture of our algorithm

major components: a) the statistical information extracted from the training corpus for the prediction algorithm; b) the predictive program that tries to suggest words to the simulated user. This component has two parts: one is word completion and the other one is word prediction. The prediction algorithm first completes the partially spelled word and then it predicts the probable words and present them in the suggestion list; c) a simulate user that types the test text. The simulated typist is a perfect user who always chooses the desired word when it is available in the prediction list and does not miss it; d) the component of updating the statistics of the words' recency of use and adding new words along with their frequency of use. To get the system adaptive to the user, two processes will be done. One is extracting word uni-, bi-, and trigrams from the current text that is being entered. The other process is saving and updating the recent extracted statistical information in a dynamic file. The recent information is related to the static file which keeps the statistical information resulted from the training corpus. When the predictor tries to predict words, first it searches the dynamic file and gives more weight to the words that are recently used; then, it uses the statistical information of the static file. Gradually as the user enters more texts, the system saves and updates the information and gets adapted to the user's style of writing and brings up more appropriate suggestions in the prediction list.

6.2. Conditions

In addition to the word prediction algorithm themselves, the parameter that varied in our experiments was the number of suggestions in the prediction list. It is assumed that the higher number of words in the suggestion list, the greater the chance of having the intended word among the suggestions; but it imposes a cognitive load on the user, because it takes the search time for the desired word longer and it is more likely that the user would miss the word they are looking for. Different users of word prediction systems may prefer different values for this parameter according to their type and level of disabilities. As it has been stated in section 4.1, Soede and Foulds (1986) experimentally identified the number of suggestions. In our work, we selected the values 1, 5 and 9 for the number of

suggestions.
In our system, the sorting order of words in the list is based on the frequency of use in which the most probable words would appear on the top of the list.
Also, in our research we designed a word processor to be compatible with the Persian specifications such as having a right to left writing system to have the cursor in its right direction.

6.3. Performance Measures

To evaluate our system, three standard performance metrics have been used in our research (Woods, 1996; Fazly, 2002):
Keystroke Saving (KSS): The percentage of keystrokes that the user saves by using the word prediction system. A higher value for keystroke saving implies a better performance.
Hit Rate (HR): The percentage of correct words that appear in the suggestion list without entering any letters of the next word. A higher hit rate implies a better performance.
Keystroke until Prediction (KuP): The average number of keystrokes that the user enters for each word before it appears in the prediction list. A lower value for this measure implies a better performance.

7. Results

To test our system, test corpus was given to the simulated typist. The length of the test corpus was 11960 words and contained 46637 characters without considering space as a character. The reason of not considering space is that after selecting any words a space will be entered automatically and the result is having a keystroke saving. On the other hand, to select a word from the list one of the Function Keys, F1 to F9, are required to be pressed to drag and drop the intended word to the text being typed. The result is that the keystroke which is saved by entering the automatic space would be lost.
The virtual typist is a Visual C^{++} program that reads in each text letter by letter. After reading each letters, it determines what the correct prediction for the current position is. The prediction program then is called and a list of suggestions is returned to the user. The user searches the prediction list for the correct prediction. If it is found in the list, the user increases the amount of correct predictions by the predictor. The correctly predicted word is then completed and the user continues to read the rest of the text.
The gained results are presented in table 1 for 1, 5 and 9 numbers of suggestions:

	KSS%	HR%	KuP
1 suggestion	31.67	5.56	2.66
5 suggestions	52.28	18.69	1.86
9 suggestions	57.57	24.42	1.65

Table 1: The summary of the gained results based on the test corpus

Based on table 1, clearly increasing the number of suggestions would increase the percentage of KSS, and HR; and decreasing KuP. The highest KSS is achieved when the numbers of suggestions are 9. The 57.57% KSS means for each 100 characters that the user is required to type to enter a text segment, more than half of the text is entered by the system, and the rest by the user. 24.42% of words appeared in the prediction list before entering any letters of the next word. On average 1.65 keystrokes were needed to be pressed by the user to type any words on the system. There is no valid average word length for Persian, but based on a sampling method from our Persian corpus, the average length is 3.91.

8. Discussion

We conducted another experiment by dividing the test corpus into 23 parts based on their subjects (genres) in the newspaper. Each text segment equally contained 1000 characters, without considering space. Then each text was given to the virtual typist one by one. The results are available in table 2. Using a development set, we found that by using 9 numbers of suggestions we gained the highest KSS. Therefore our final setup

uses the same 9 numbers of suggestions as its default.

Subjects of News	KSS %	HR (9) %	KuP
Arts	67.21	29.21	1.32
Arts and Literature	55.98	24.90	1.55
Cinema	62.50	29.57	1.42
City	62.48	32.22	1.36
Council	66.76	29.88	1.30
Disables	64.53	25.00	1.40
Economics	68.22	35.24	1.19
Education	73.57	40.84	1.03
Environment	61.83	29.85	1.39
Foreign News	69.83	34.76	1.16
Literature	51.54	29.59	1.46
Media	69.00	31.47	1.21
Music	51.53	22.67	1.76
Political News	72.22	38.33	1.07
Rights of Citizens	60.34	31.36	1.44
Science	65.21	30.48	1.27
Science and Culture	56.92	27.43	1.45
Social News	59.03	30.38	1.54
Society	64.08	30.51	1.29
Sports News	70.84	34.63	1.11
Tehran News	68.55	34.00	1.25
Thought	70.78	39.03	1.06
World Sports	63.41	29.14	1.45

Table 2: KSS, HR and KuP performance measures for different genres of test corpora with 1000 characterS

By comparing the results, we observed when KSS increases, HR increases, and KuP decreases; and vice versa. This observation shows that there is a one-to-one correspondence between KSS and HR but they are quite contrary to KuP. Some subjects (genres) such as *Education* achieved the highest KSS, the highest HR and the lowest KuP. But *Music* achieved the lowest KSS, the lowest HR, and the highest KuP.
In general, we saw based on the sequence of words in different genres, it has different effects on the gained results. It seems that the texts on the subjects of *Thought*, *Sports News*, *Political News*, and *Education* which gained the keystroke saving of more than 70%, have more words and sequences of words in common, and the words are more predictable as a result. It means the dependency of words with each other being collocated is high. But the texts on the genres of *Music*, *Literature*, *Arts and Literature*, *Science and Culture*, and *Social News* which gained the keystroke saving of less than 60% have some words that are not available in the lexicon of the program and/or the sequence of the words that come together on these genres are rare and less predictable consequently.
Of course by adapting the system for a special purpose, a better result would be gained as it was described in section 6.1.

9. Conclusion

We have designed, implemented and tested a word predictor for Persian. To the best of our knowledge this is the first attempt for the language. Using such a system saved a great number of keystrokes; and it led to reduction of user's effort.

10. Further Work

Our future work is adding a spell-checker to the system to replace various spellings of a word to the available word in the lexicon of the system, adding syntactic and later semantic information of the Persian language to the system to make predictions more appropriate syntactically and semantically.

Bibliography

Assi, S.M. (2004) "Persian language and IT" In *Proceedings of the 2nd Workshop on Information Technology and Its Disciplines (WITID), Kish Island, Iran*, Feb. 24-26, 2004, pp. 85-94.

Banerjee, S. and T. Pedersen. (2003) "The design, implementation and use of the Ngram Statistics Package (NSP)." In *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City*, pp. 370-381.

Booth, L. and W. Beattie and A. Newell (1990) "I know what you mean". *Special Children*, pp. 26-27.

Carlbeger, A. and T. Magnuson and J. Carlberger and H. Wachtmeister and S. Hunnicutt. (1997a)

"Probability-based word prediction for writing support in dyslexia." In Barner, R., Heldner, M., Sullivan, K., and Wretling, P., editors, *Proceedings of Fonetik '97 Conference*, Volume 4, pp. 17-20.

Carlberger, A. and J. Carlberger and T. Magnuson and M.S. Hunnicutt and S.E. Palazuelos-Cagigas and S.A. Navarro. (1997b) "Profet, a new generation of word prediction: An evaluation study." Copestake, A., Langer, S. and Palazuelos-Cagigas S., editors, Natural Language Processing for Communication aids, In *Proceedings of a workshop sponsored by ACL, Madrid, Spain*, pp 23-28.

Fazly, A. (2002) *The Use of Syntax in Word Completion Utilities*. Master dissertation. Canada: University of Toronto.

Ghayoomi, M. (2004) *Word Prediction in Computational Processing of the Persian Language*. Master dissertation. Iran: Islamic Azad University, Tehran Central Branch.

Gustavii, E. and E Pettersson (2003) *A Swedish Grammar for Word Prediction*. Stockholm: Uppsala University

Jurafsky, D. and J.H. Martin. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. New Jersey: Prentice-Hall.

Klund, J. and M. Novak (2001) "If word prediction can help, which program do you choose?"
http://trace.wisc.edu/docs/wordprediction2001/index.htm

Manning, C.D., and H. Schütze. (1999) *Foundations of Statistical Natural Language Processing*. The MIT Press.

Nantais, T. and F. Shein and M. Johansson. (2001) "Efficacy of the word prediction algorithm in WordQ™." In *Proceedings of the 24th Annual Conference on Technology and Disability, RESNA.*

Shein, F. and T. Nantais and R. Nishiyama and C. Tam and P. Marshall. (2001) "Word cueing for persons with writing difficulties: WordQ." *The16th Annual International Conference on Technology and Persons with Disabilities, California State University at Northridge, Los Angeles, CA*, March.

Soede, M. and R.A. Foulds (1986) "Dilemma of prediction in communication aids and mental load." *In Proceedings of the 9th Annual Conference on Rehabilitation Technology*, 357-359.

Swiffin, A.L. and J.A. Pickering and J. L. Arnott, and A. F. Newell (1985) "PAL: An effort efficient portable communication aid and keyboard emulator." In *Proceedings of the 8th Annual Conference on Rehabilitation Technology*, pp. 197-199.

Wood, M.E.J. (1996) *Syntactic Pre-Processing in Single-Word Prediction for Disabled People*. Ph.D. dissertation. University of Bristol, Bristol.