A Framework for Spelling Correction in Persian Language Using Noisy Channel Model

Mohammad Hoseyn Sheykholeslam[†], **Behrouz Minaei-Bidgoli**[‡], **Hossein Juzi**[†]

[†] Computer Research Center of Islamic Sciences,
Qom, Iran

[‡] Iran University of Science and Technology
Tehran, Iran

E-mail: sheykholeslam@noornet.net, b_minaei@iust.ac.ir, hjuzi@noornet.net

Abstract

There are several methods offered for spelling correction in Farsi (Persian) Language. Unfortunately no powerful framework has been implemented because of lack of a large training set in Farsi as an accurate model. A training set consisting of erroneous and related correction string pairs have been obtained from a large number of instances of the books each of which were typed two times in Computer Research Center of Islamic Sciences. We trained our error model using this huge set. In testing part after finding erroneous words in sample text, our program proposes some candidates for related correction. The paper focuses on describing the method of ranking related corrections. This method is customized version of Noisy Channel Spelling Correction for Farsi. This ranking method attempts to find intended correction c from a typo t, that maximizes P(c) P(t | c). In this paper different methods are described and analyzed to obtain a wide overview of the field. Our evaluation results show that Noisy Channel Model using our corpus and training set in this framework works more accurately and improves efficiently in comparison with other methods.

Keywords: Spelling Error Correction, Noisy Channel Model, Persian Language

1. Introduction

Spelling error correction is done in two levels: Isolated Word and Context Based. We discuss in this paper about one of the efficient Isolated Word methods. Commonly Spell Checking includes two main steps. The first step involves the utilization of a dictionary to detect erroneous strings and the second one includes a set of algorithms and techniques to be used for spell checking. These techniques are utilized within three steps: (1) Generating all of substitutions, (2) Validation of substitute strings in the dictionary, (3) Ranking the suggestions (Naseem, 2004).

In the past, most misspellings were codified. The common misspelling models were used to correct the errors. This technique was used by Damerau (1964), Angel (1983), and Zobel (1994). The erroneous data set is inputted for codifying the error patterns (Church & Gale, 1991; Brill & Moore, 2000; Toutanova & Moore, 2002). Today the probabilistic models are used to correct the misspellings.

2. Word-based Spell Checking techniques

In general, the word-based spell checking techniques are divided into these sub-groups: Edit Distance Techniques, Phonetics Based Techniques, Similarity Key Techniques, N-Gram Based Techniques, and Probabilistic Techniques. Of course, these techniques are not completely independent from each other; rather they may have some overlaps.

2.1 Edit Distance techniques

The term "Edit Distance" was first introduced by Wagner (1974). It really means the minimum changes needed for converting a string to another string. A similar concept was defined by Damerau (1964) (Kukich, 1992).

2.1.1. Damerau Single-Error technique

The most famous technique of this group of techniques is Damerau single-error technique. Damerau showed that 80% of single character errors belong to one of the following categories:

(1) Inserting a character, (2) Deleting a character, (3) Substituting a character, (4) Swapping a character by its neighbouring character (Damerau, 1964).

2.1.2. Levenshtein technique

The Levenshtein distance calculates the distance between the two characters using the insertion, deletion, and substitution operators. But it is more comprehensive than the Damerau's technique because it allows multiple error occurrences in a word (Erikson, 1997).

2.1.3. Weighted Edit Distance technique

In Damerau and Levenshtein's techniques, all the characters have equal probability for deletion and insertion, and they are substitutable with all alphabet letters. This is in fact wrong.

In a research by Kukich (1992), it was shown that 58% of the substitutions are due to pressing the neighbouring keys (on the keyboard).

To apply this technique, an n by n matrix is constructed, in which n is the number of the alphabet letters. Any ij^{th} element in the matrix is the probability of substitution of the i^{th} letter with j^{th} letter (Erikson, 1997).

2.1.4. Tapering technique

Tapering is another technique for Edit Distance Correction. It works like this: the word that is different with the correct word from the end, is more similar than the word that is different from the beginning (Zobel & Dart, 1996).

2.2 Phonetics based techniques

These techniques focus on the sound of the omitted characters in the erroneous words. The goal is to find a word in dictionary which is phonetically the closest to the erroneous word. This class of techniques contains famous methods like: Soundex Algorithm, Phonix Algorithm, and Editex Algorithm (Kukich, 1992).

2.3 Similarity key techniques

These techniques associate a code to letters, like in Soundex and Phonix algorithms. These techniques can be combined with the Edit Distance technique. Skeleton Key, Omission Key, and Plato Key techniques belong to Similarity Key Techniques (Kukich, 1992).

2.4 Probabilistic techniques

All the considered techniques use the degree of similarity and measurement of different distances as their criteria for finding a substitute word. The problem with these methods is that they ignore important factors affecting the error patterns. By identifying these factors, we should devise a new technique. A comprehensive and suitable technique for this is a model based on probable error (Kukich, 1992).

These methods provide excellent ranking by using a vast corpus and a language related training set. Unfortunately, for languages like Farsi such corpuses are not available, so these techniques cannot be used. Production of huge language error corpora in Noor Computer Research Center of Islamic Sciences (CRCIS) made it possible to use these techniques.

2.2.1. Noisy Channel Model

A model of probable error that can be used for different languages and educational fields and is able to adjust its parameters is called "Noisy Channel".

If Noisy Channel is modelled correctly, it can make a sound guess about the erroneous word (Brill & Moore, 2000; Kukich, 1992; Jurafsky & Martin, 2000).

This method has successfully been used in many different speech processing and text processing programs in which various identification and classification of faulty and vague data are used (Jurafsky & Martin, 2000):

$$w = \arg\max_i w_i \, P(w_i \mid s) \tag{1}$$

Where, *w* is a correct word and *s* is an incorrect word an $\arg max$ returns the highest value for an expression and w_i is the possible suggestions to be substituted for *s*.

Since these probabilities show the behavior of the origin on the error, they are called Channel Probabilities (Kukich, 1992).

The Noisy Channel Technique was first employed in 1990 for spell checking (Kernighan et al, 1990). The probability method was only used for ranking the substitution options.

The technique for a "learning model" is an example of an EM algorithm in which the models' parameters are repeatedly estimated until we reach a stabilizing state (Jurafsky & Martin, 2000).

2.2.2. An improved model for Noisy Channel

Brill and Moore (2000) introduced a much more complex and comprehensive technique by using the Noisy Channel method. In this model, instead of using the character by character corrections used by Church and Gale (1990), a number of string by string correction techniques are used. A string is a chain of letters with a length of zero or longer. The application of more comprehensive operations helps one to cover both multiple and single errors.

Assume that Σ is set of alphabet, then:

$$\alpha \to \beta \; or \; P(\alpha \mid \beta)$$
$$if \; \alpha, \beta \; \in \sum \tag{2}$$

To measure $P(s \mid w)$, *s* and w are divided into r_1, r_2 ... sections.

$$P(s \mid w) = P(r_{1s} \mid r_{1w}) * P(r_{2s} \mid r_{2w}) * \ldots \tag{3}$$

When all substitutions for $\alpha \to \beta$ in the learning data are calculated, we can calculate $P(\alpha \mid \beta)$.

As it was seen, the calculation of $\alpha \to \beta$ is done through the learning calculations; but measuring $Count(\alpha)$ is a little problematic. If we populate the model by some sets of learning, we can easily set the number of string occurrences equal to the result of $Count(\alpha)$. But if the number of $\alpha \to \beta$ is calculated by set of learning data then an independent set must be used to estimate the $Count(\alpha)$, so that we can calculate the number of substitution occurrences in that set and then normalize it by a human error factor.

In the original technique by Brill and Moore, a dictionary (lexicon) was inserted into a Trie[1].

A Trie is a special type of an ordered-tree for saving associative arrays keys of which are usually alphabet strings.

Church and Gale reported a precision of 98.8% for position confusion parameters and a triple context window (Kernighan et al, 1990).

2.2.3. Pronunciation modelling for Improved Spelling Correction

In the corrected Noisy Channel model, introduced by Toutanova and Moore, in 2002, α and β are trails of phonics. In this technique all the words in the dictionary and the erroneous terms must be converted from a trail of

[1] http://en.wikipedia.org/wiki/Trie

letters to a trail of phonic items. This is easily done for the words in the dictionary because they have a specific pronunciation. But in erroneous terms, a model for turning characters into phonic items is needed. The most common model for this is the N-Grams model which belongs to Fisher (Toutanova & Moore, 2002).
Toutanova and Moore showed that the combination of the Noisy Channel model based on phonic items and the Noisy Channel model based on characters has a higher efficiency when compared with either of the models separately. The composite model performed the spell checking with 95.58% precision, and other best three models didn't give a result better than 99.5%.

3. Methodology

As the initial step of implementation of a system for spellchecking in Farsi using Noisy Channel Model, the error model of huge data of the CRCIS was extracted. This information was about replication of each word and each character, statistics of four single edit operations, etc. According to results the system ranks the retrieved suggestions.

3.1 Preparing huge corpora

The CRCIS implements encyclopedic applications about different topics and cultural or religious individuals. These applications utilize the printed books available in the world. The procedure is that each book is typed two times by two expert typists. Then the third expert typist compares the two typescripts using specific application and dictionary. Then he resolves the discrepancies. At last, he prepares a copy of the book's typescript that is free of even a single error. In The CRCIS, hundreds of book is typed yearly.
For generating a corpus of erroneous and correction word we have extracted all of conflict words between each of two typed versions of specific book and third version of it. This task was done on some 2000 books that contain nearly 170 million words. Some 3 million pairs of misspellings and corrections were obtained.

3.2 Suggesting correction

In the first stage, this program uses the Damerau's Single Edit Distance technique, so that the words that have the edit distance of typing insertion, deletion, substitution, or swapping equal to 1 with the correct word. In example for erroneous string like "پادشا" /pɑdeʃɑ/ (a misspelling for "پاداش" /pɑdɑʃ/ (means prize), "پاشا" /pɑʃɑ/ (means great rank in political system), and "پادشاه" (means king) /pɑdeʃɑh/.) the system generate the following:
Single letter insertion:
"پادشا" → "اپادشا" by insertion of 'ا' before 'پـ'.
Single letter deletion:
"پادشا" → "ادشا" by deletion of 'پـ'.
Single letter substitution:
"پادشا" → "بادشا" by substitution of 'پـ' with 'بـ'.
Two adjacent letters transposition:
"پادشا" → "اپدشا" by transposition of 'پـ' with for 'ـا'.
From the all of suggested strings, the system detects the following valid suggestions:

Incorrect	Correct	Edit Operation
پادشا	پادشاه	insertion of 'ه' after last 'ـا'
پادشا	پاشا	deletion of 'د'
پادشا	پاداش	swaping 'د' for 'ا'

Table 1: Valid suggestions for an erroneous string

3.3 Ranking

To score each proposed word its probability is calculated at first by the following formula:

$$P(c) = \frac{freq(c) + 0.5}{N} \quad (4)$$

Where, freq(c) is the number of occurrence of the letter "c" in the typescript of 2000 books typed in Computer Research Center of Islamic Sciences. And N is the number of words in all of these 2000 books which amounted to some 170 million words.
According to the Box and Tiao's technique we can achieve a Posterior Distribution of P by using a probability of an unlearned precondition. The value used for this is r + 0.5 instead of r. this probability is called "Expected Likelihood Estimate (ELE)" (Box & Tiao, 1973).
The shortcomings of this technique were studied in a research by Church and Gale [1]. After the calculation of P(c), the conditional probabilities of P (t | c) which is calculated by formula (2) is obtained by using the 4 confusion matrices below:

$$P(t \mid c) = \begin{cases} \frac{Add[c_{p-1}, t_p]}{chars[c_{p-1}, t_p]}, Insertion \\ \frac{Del[c_{p-1}, c_p]}{chars[c_{p-1}]}, Deletion \\ \frac{Sub[t_p, c_p]}{chars[c_p]}, Substitiution \\ \frac{Rev[t_p, t_{p+1}]}{chars[t_p, t_{p+1}]}, Reversion \end{cases} \quad (5)$$

Where:
Del[x, y] is the number of x's that are typed as xy.
Add[x, y] is the number of xy's that are typed as x.
Sub[x, y] is the number of x's that are typed as y.
Rev[x, y] is the number of yx's that are typed as xy.

The probabilities are obtained by dividing the confusion matrices by chars[x, y] or chars[x]. These matrices show the number of "xy" or "x" characters in the typescript of 2000 books.
Each proposed word (c) is scored by using the formula (6):

$$Score(c) = P(c).P(T \mid c) \quad (6)$$

And then it is normalized by adding the scores obtained from all the proposed words in formula (7):

$$Normal(c) = \frac{Score(c)}{\sum Score(c_i)}, c_i \in S \quad (7)$$

Where, c_i is the correct word proposed and S is the set of all proposals.
The raw and normalized probabilities for the proposed words for the term "پادشا" are calculated as shown in Table 2 below:

Correct	P(c)	P(t | c)	Score(c)	Normal(c)
پاداش	$1.45\ e^{-8}$	$1.8\ e^{-5}$	$2.7\ e^{-13}$	14.57
پادشاه	$4.36\ e^{-8}$	$3.1\ e^{-5}$	$1.3\ e^{-12}$	72.89
پاشا	$33.49\ e^{-8}$	$6.9\ e^{-7}$	$2.3\ e^{-13}$	12.52

Table 2: Sample calculation for Score(c) formula

4. Evaluation

Noisy Channel Model (NCM) technique is tested and compared with famous techniques such as Jaro-Winkler (JW) with Frequency-based ranking and Damerau-Levenshtein (DL) with Frequency-based ranking on a test set with exact size of 18,214 pairs of erroneous and related correct words.

Model	1-Best	5-Best	10-Best
JW	55.3	83.1	90.5
DL	58.1	87.4	90.5
NCM	75.3	90.5	90.5

Table 3: Results of different ranking approaches

The first column in Table 3 shows the percentage of erroneous strings in which the related correct words produced the first suggested word by the system. The second and third columns show that the corresponding correction belong to top five and top ten items from the list of suggestions.

5. Conclusion and Future Works

This paper proposed an expert system for spelling correction using Noisy Channel Model in Persian Language. The results show the effect of using massive corpus of data and statistical method like Noisy Channel Model in generating a more efficient error model for Farsi Language and significant improvement of the first suggestion accuracy. In spite of improvement of suggestions there are many tasks to obtain better results. The closest step for perfection of accuracy is implementing of improved model for Noisy Channel in Persian Language using our huge corpora.

6. Acknowledgements

This project was supported in part by the Noor Computer Research Center of Islamic Sciences, Qom, Iran.
The authors would also like to thank for special supports and collaborations of staffs of Noor Computer Research Center for typing and correcting of the huge data.

7. References

Alberga, C.N., "String Similarity and Misspelling", *In Communications of ACM*, Vol. 10, No. 5, pp. 302--313, 1967.

Box, G. E. P., Tiao, G. C., "Bayesian Inference in Statistical Analysis", *Addison-Wesley, Reading, Massachusetts*, 1973.

Brill, E., Moore, R. C., "An Improved Error Model for Noisy Channel Spelling Correction. *In proceedings of 38th Annual meeting of Association for Computational Linguistics*, pp. 286--293, 2000.

Christian, P., "Soundex - can it be improved?", *Computers in Genealogy*, Vol. 6, No. 5, 1998.

Church, K., Gale, W., "Probability Scoring for Spelling Correction", *Statistics and Computing*, Vol. 1, pp. 93--103, 1991.

Damerau, F.J., "A Technique for Computer Detection and Correction of Spelling Errors", *In Communications of ACM*, Vol. 7, No. 3, pp. 171--177, 1964.

Erikson, K., "Approximate Swedish Name Matching - Survey and Test of Different Algorithms", 1997.

Holmes, D., McCabe, C. M., "Improving precision and recall for soundex retrieval", *In Proceedings of the IEEE International Conference on Information Technology -Codingand Computing (ITCC)*, Las Vegas, 2002.

Jurafsky, D., Martin, J. H., "Speech and Language Processing.: An Introduction to Natural Language Processing", *Computational Linguistics and Speech Recognition Prentice Hall*, 1st edition, 2000.

Kann, V., Domeij, R., Hollman, J., Tillenius, M., "Implementation Aspects and Applications of A Spelling Correction Algorithm", 1998.

Kernighan, M. D., Church, K. W., Gale, W. A., "A Spelling Correction Program Based on a Noisy Channel Model", *In Proceedings of COLING-90, The*

13th International Conference On Computational Linguistics, Vol. 2, 1990.

Kukich, K., "Techniques for Automatically Correcting Words in Text", *ACM Computing Survey*, Vol. 14, No. 4, pp. 377--439, 1992.

Naseem, T. (2004). A Hybrid Approach for Urdu Spell Checking. Master of Science (Computer Science) thesis at the National University of Computer & Emerging Sciences.

Pfeifer, U., Poersch, T., Fuhr, N., "Searching Proper Names in Databases", *Proceedings of the Conference on Hypertext Information Retrieval Multimedia, Germany*, pp. 259--275, 1995.

Pfeifer, U., Poersch, T., Fuhr, N., "Retrieval Effectiveness of Name Search Methods", *Information Processing and Management*, Vol. 32, No. 6, pp. 667--679, 1996.

Pollock, J.J., Zamora, A., "Automatic spelling correction in scientific and scholarly text", *CACM*, Vol. 27, pp. 358--368, 1984.

Pollock, J.,Zamora, A. "Automatic spelling correction in scientific and scholarly text", *CACM*, pp. 27, 358--368, 1984.

Toutanova, K., Moore, R. C., "Pronunciation Modeling for Improved Spelling Correction "*In proceedings of 40th Annual meeting of Association for Computational Linguistics*, pp. 144--151, 2002.

Zobel, J., Dart, P. "Phonetic String Matching: Lessons from Information Retrieval", 1996.

Zobel, J., Dart, P. "Finding Approximate Matches in Large Lexicons", *Software-Practice and Experience*, Vol. 25, No. 3, pp. 331--345, 1995.