UNSUPERVISED PART OF SPEECH TAGGING FOR PERSIAN

Tayebeh Mosavi Miangah[1] and Ali Delavar khalafi[2]

[1]English Language Department, Payame Noor University, Yazd, Iran

mosavit@pnu.ac.ir

[2]Mathematics Department, Yazd University, Yazd, Iran

delavarkh@mail.ipm.ir

Abstract

In this paper we present a rather novel unsupervised method for part of speech (below POS) disambiguation which has been applied to Persian. This method known as Iterative Improved Feedback (IIF) Model, which is a heuristic one, uses only a raw corpus of Persian as well as all possible tags for every word in that corpus as input. During the process of tagging, the algorithm passes through several iterations corresponding to n-gram levels of analysis to disambiguate each word based on a previously defined threshold. The total accuracy of the program applying in Persian texts has been calculated as 93 percent, which seems very encouraging for POS tagging in this language.

Key words

Iterative Improved Feedback model, machine translation, part of speech tagging, Persian language, unsupervised learning.

1. Introduction

Today, using corpora for solving different linguistic problems has gained a very high interest from the specialists in the field of computational linguistics. In computational linguistics the term 'corpus' refers to a collection of annotated or unannotated words in the body of text, which can be used for various problems such as word sense disambiguation [1]; phrase recognition [2]; morphological analysis and automatic lemmatization [3], [4]; language teaching [5]; machine translation [6]; information retrieval [7]; part of speech tagging [8] and many other problems.
In the field of POS tagging many works have also been done, applying different approaches. Some researchers tried to solve this very problem using rule-based approaches [9], [10] and some preferred to used statistical information to cope with the problem [11], [12].

In statistics-based approaches there are two alternative training methods based on which the related frequencies and subsequent calculations are to be carried out. The first method uses a tagged training corpus from which the probabilities for each word or each tag or a sequence of tags are extracted [4]. For these kinds of methods to be able to get to a reliable estimation, a large

DOI : 10.5121/ijaia.2012.3204

tagged corpus is required. Merialdo [13], for instance, uses such a corpus to disambiguate English texts using Hidden Markov Model.

To speak of Persian, it should be said that corpus-based approaches for text analysis have a rather short story in Persian language. The only serious attempt in this connection is constructing an interactive POS tagging system developed by Assi and Abdolhosseini [14]. In their project they followed the methods proposed in Schuetze [15]. It is based on the hypothesis that syntactic behavior is reflected in co-occurrence patterns. Therefore, the similarity between two words will be measured with respect to their syntactic behaviors to their left side by the degree to which they share the same neighbors on the left. So, the word types are recognized according to their distributional similarity (their similarity in terms of sharing the same neighbors), and then each category can be manually tagged. In this way a grammatically tagged corpus of Persian was created making up of 45 tags which have designed with reference to the categories normally introduced in dictionaries. Each tag is made up of one to five characters. In general, the accuracy of this kind of distributional POS tagging system proved to be 57.5 percent. Megerdoomian [16] also has reported some of the main challenges that arise in the development of a Persian part of speech tagger - such as encoding issues, long-distance dependencies in morphology, recognition of complex tokens, word and phrasal boundaries, and analysis of multiword expressions - and proposed approaches to resolving these issues.

The second method uses untagged or raw corpus for training, which is known as unsupervised method because it does not need human help in tagging the training data. For most unsupervised systems of tagging the only information is a lexicon assigning all possible tags to each word and a very large untagged corpus. In this respect it is comparable to stochastic taggers which were trained using Baum-Welch Algorithm [17] in which only an untagged corpus and a dictionary providing all possible POS tags for every word in the corpus are needed.
Though unsupervised training yields lower precision rates than supervised training, it has the following advantages: For most languages large annotated corpora are not available, while unannotated corpora in electronic form are abundant for most languages. Furthermore, the training of the tagger is domain-dependent. It means applying a trained tagger to a text of another domain usually results in a degradation of precision [18].

There are different approaches to unsupervised learning. Cutting, and his colleague [2] used a HMM and gained 96 percent accuracy on the Brown corpus. In their approach they used an idea originated by Kupiec [19] in which words are grouped in equivalence classes. Thus reducing the number of parameters that need to be adjusted. Brill [20] used transformation-based unsupervised tagging which begins with an unannotated text corpus and a dictionary listing words and the allowable POS tags for each word. Then a set of rules are applied and kept looking until all possible candidates obtain null score.

The accuracy reported for this approach is 95.1 percent testing on Penn Treebank Wall Street Journal corpus and 96.0 percent testing on Brown corpus. Tzoukermann and Radev [21] use word class for POS disambiguation. They investigate a direction for coming up with different kinds of probabilities based on paradigms of tags for given word. Their estimations are based not on the words, but on the sets of tags associated with a word. They claim that this approach gives a more efficient representation of the data in order to distinguish word POS. The accuracy of this method

reached about 95 percent for POS disambiguation of unrestricted French text. In fact, the term 'genotype' used in the present paper has been adopted from their work.
In this paper we present a rather different approach to unsupervised POS tagging which has been applied to Persian texts. The presented method uses only a lexicon including all possible tags of each word which is referred to as 'genotype' and a corpus containing a large amount of raw Persian texts. A mathematical heuristic model named as Iterative Improved Feedback (IIF) Model is applied to extract the relevant frequencies of the word genotypes from the raw corpus and use them for disambiguating unrestricted Persian texts from the standpoint of POS. When a word was disambiguated, only one tag would be suitable for the context to which the word belongs and we refer to this correct tag as 'genotype decision'.

Working with these kinds of input for building a tagger for a low density language[1] like Persian for which large annotated corpus are scarce seems to be highly advantageous. In the next section the proposed IIF Model is introduced. Discussion of the results from the experiment on this model is presented in Section 3. Sections 4 and 5 deal with comparing the presented model to the other similar models and concluding the study, respectively.

2. The Model

2.1. Corpus Preparation

As mentioned formerly, in our unsupervised tagging system there is no need to collect an annotated Persian corpus, but a raw corpus, a lexicon comprising all Persian words and their allowable POS tags or genotype. In light of this, we tried to collect as many Persian texts in different fields as possible (over 100 million words). These texts are mainly extracted from scientific articles, journals, books, interviews newspapers, etc. found in the Internet and preprocessed before entering to the corpus. That is, all tables, pictures, figures or diagrams are to be deleted from the texts to be ready for the corpus. Moreover, the texts should be converted to an XML format to be suitable for use on Internet sites. In this stage the texts can be entered into the corpus to be used for different types of linguistic research. When the texts are entered the corpus, they are automatically tokenized in terms of sentence.

The first stage of the work which is known as 'Initial State Annotator' [20] tags each word in the corpus with its genotype. Determining the genotype of every word can be easily extracted from a machine readable dictionary. Once the genotype of each word in the lexicon has been determined, the algorithm has been passed through the first stage, namely, 'Initial State Annotator'. In this phase every word has been assigned with a group of tags, all of which are possible syntactic categories of that word in different contexts.

The tagset we used for 'Initial State Annotator' stage comprises 68 tags each of which consists of from 2 to 5 characters. In linguistic analysis using one or the other tagset depends on its application in the related task. Specifiability of a tagset is preferable, which means that the linguistic task of the tagger has been carefully specified resulting in more precise evaluation of tagger performance [22] Our tagset selection is based on Persian morphological characteristics and requirements of the current experiment. For instance, in our tagset we include the tag [nsgz]

[1]- a non-widely spoken language for which resources are scarce

standing for singular common noun plus indefinite suffix 'I', since in Persian there are a relatively large number of words whose genotype contains several tags including [nsgz] and the genotype decision is an adjective, simple noun or verb.

As mentioned earlier, the corpus has been previously tagged with genotype belonging to each word. This work which is totally automatic is easily carried out using an electronic dictionary containing all possible tags for each Persian word. Table 1 shows the percent dedicated to each group of words in our corpus in terms of their number of possible tags.

Table 1. Distribution table of Persian words in the corpus based on their genotype

Members in genotype	1	2	3	4	5
percent of words	74.85	20	4.6	0.42	0.13

About 74.85 percent of the words in our corpus have only one member in their genotype and thus unambiguous, about 20 percent of the words have two members in their genotype, about 4.6 percent of the words have three members, about 0.42 percent of them have four members and about 0.13 percent of them have five members in their genotype. As it is shown in table 1, the majority of Persian words in our selected corpus are unambiguous, that is, having only one member in their genotype. So, it seems reasonable to base our calculations only on this majority and look for successive words which are all unambiguous. Although calculation based on only unambiguous successive words is more complex than the whole words and also in the former case we need a larger corpus than the latter one, we encounter no problem in this regard, because we have a sufficiently large unannotated corpus to be analyzed.

2.2. Iterative Improved Feedback Model

In this experiment our input is only a raw corpus whose words have been assigned with all possible tags called as genotype. The output of our program is , naturally, as any other tagger system, a test corpus which has been automatically tagged for part of speech based on information derived from the training corpus.

The program reads each word from the right to the left (according to Persian writing system). When a word whose genotype contains more than one tags is found, the program sets to calculate the frequencies of each member of the given word's genotype with the tag assigned for the next word among unambiguous successive words of the training corpus. If the next word to the given word is also ambiguous, then the number of calculations of the frequencies becomes θ_{n+1}, where θ_{n+1} is the number of tags belonging to the next word. In this regard the algorithm can be shown as follows:

If W_{n+1} is unambiguous, for all $W_n \in S_n$, compute:

$P(t_n \mid t_{n+1})$

If W_{n+1} = ambiguous, for all $W_{n+1} \in S_{n+1}$, compute:
$P(t_n, t_{n+1})$

where, S_n is a set of all possible tags for W_n.

This is only the first iteration of our program which is called bi-gram level of analysis. In the second iteration in addition to the probabilities obtained in the first iteration, we calculate the frequencies of three successive words tags, namely, t_{n-1}, t_n and t_{n+1} among the unambiguous successive words of the training corpus. The POS of the W_{n-1} is always unambiguous, since it has already been disambiguated. So, there is no problem with calculations of the related probabilities in this iteration and the subsequent ones. The second iteration or the tri-gram level of analysis is followed by the next level, namely, four-gram level in which the calculation of the frequencies is carried out using t_{n-2}, t_{n-1}, t_n and t_{n+1}, again among the unambiguous successive words of the training corpus. For each iteration the probabilities are all determined and the number of our calculations in each iteration depends on the number of the genotype of W_n and of W_{n+1}. We used the maximum likelihood probabilities $\hat{P}$ which are derived from the relative frequencies for each iteration as follows:

Bi-grams:

(1) $$\hat{P}\ (t_n \mid t_{n+1}) = \frac{f(t_n, t_{n+1})}{f(t_{n+1})}$$

Tri-grams:

(2) $$\hat{P}\ (t_n \mid t_{n+1}, t_{n-1}) = \frac{f(t_{n-1}, t_n, t_{n+1})}{f(t_{n-1}, t_{n+1})}$$

Four-grams:

(3) $$\hat{P}\ (t_n \mid t_{n+1}, t_{n-1}, t_{n-2}) = \frac{f(t_{n-2}, t_{n-1}, t_n, t_{n+1})}{f(t_{n-1}, t_{n-2}, t_{n+1})}$$

To decide whether we should pass over every iteration to the next one or not, we defined a threshold using following formula:

(4) $$T_k = \hat{P}(E_1^* \mid t_{n+1}, t_{n-1}, ..., t_{n-k}) - \hat{P}(E_2^* \mid t_{n+1}, t_{n-1}, ..., t_{n-k})$$

where, T_k = threshold in kth iteration, and

$$E_1^* = Arg \max_{E \in Sn} \left\{ \hat{P}(E \mid t_{n+1}, t_{n-1}, ..., t_{n-k}) \right\}$$
$$E_2^* = Arg \max_{E \in Sn - \{E_1^*\}} \left\{ \hat{P}(E \mid t_{n+1}, t_{n-1}, ..., t_{n-k}) \right\}$$

If the difference between the highest probability of each iteration and its next highest probability is more than our threshold which is here 0.07, then the algorithm stops in that iteration without trying the next iteration and the most probable tag is assigned for the given word as the correct tag. However, in case that the difference between the two highest probabilities is lower than the threshold, the calculation of frequencies and subsequent probabilities goes on to the next iteration up to the desirable threshold is satisfied. Then the algorithm stops in next stage. For some ambiguous words in Persian the algorithm stops in the first iteration, because as it is said in section 2, the W_n_1 has a high effect on determination of POS tag for W. For these words, the difference between the highest probability and the next one is large enough not to continue the calculations in the subsequent iterations.

Using the gained results from the corpus, it is expected that the claim that disambiguation of a word at most up to the third iteration in significance level α will be possible is not rejected. For this purpose, we use Hypothesis Testing. We define:

$$X_n = \begin{cases} 1: \text{The n}^{\text{th}} \text{ word is disambiguated at most in third iteration} \\ \\ 0: \text{Otherwise} \end{cases}$$

Let: $Y = X_1 + ... + X_n$

Since $Var(X_n) = pq$ and $E(X_n) = p$, so according to Central limit theorem [23], the random variable

$$Z_n = \frac{Y_n - n\mu}{\sigma\sqrt{n}}$$

is asymptotically normal with the mean 0 and the variance 1. Now, based on the random sample (with the size of 1000 words) extracted from the corpus we want to verify the claim that disambiguation of a word at most up to the third iteration with the probability of 93% is possible, equivalently, $p = 0.93$, $q = 1 - p = 0.07$. In this case we will have:

$$\begin{cases} H_0 : \mu = 930 \\ H_1 : \mu \neq 930 \end{cases}$$

In our sample it is observed that 925 words have been correctly disambiguated in the third iteration. So, since $P(-z_{\frac{\alpha}{2}} \leq Z \leq z_{\frac{\alpha}{2}}) = 1 - \alpha$, then:

$$-z_{\frac{0.5}{2}} = -1.96 \leq Z = \frac{925 - 930}{0.25515\sqrt{1000}} = -0.61969 \leq z_{\frac{0.5}{2}} = 1.96 \ .$$

Thus, according to this sample, there are not sufficient evidences for rejecting H_0 in the significance level $\alpha = 0.5$. In other words, it can inexactly be said that disambiguation of a word at most up to the third iteration with the probability of 93% will be possible.

From the third iteration onward the algorithm does not go forward because in Persian the third word from the right has a little impact on the given word' tag. Figure 1 describes the complete algorithm. Here are the abbreviations:
ISA= Initial State Annotator
D= Difference between the two highest probabilities
T= Threshold (0.07)

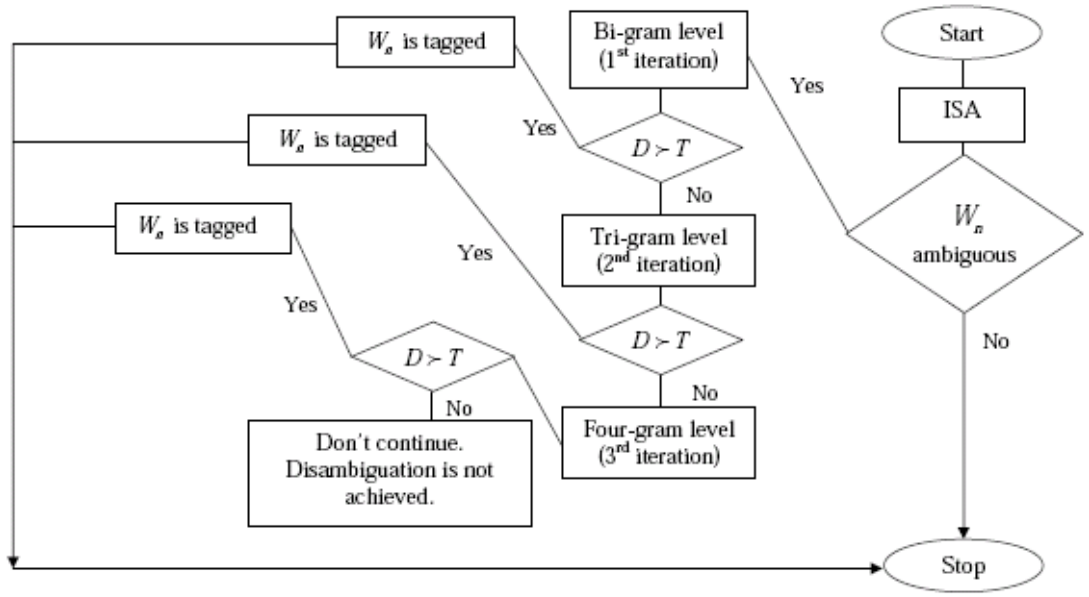

Figure 1: The algorithm of IIF Model for POS tagging

2.3. An Example.

Let us have a look at an example. Consider the following sentence in which the given word is *?mdh* , having three possible tags:

a:qa:i Ahmdi gft: ?mdh mshkla:t ma: prda:xt hzinhha: ast.
(Mr. Ahmadi said: the main part of our problems is the payment of the expenses.)

a:qa:i/[nsg] Ah*mdi*/[nsg] *gft*/[nsg][pas] :/[co] *?mdh*/[aj][av][nsg] *mshkla:t*/[npl]
ma:/[pnp] *prda:xt*/[nsg][pas] *hzinhha:*/[npl] ast/[vl] ./[fs]
In the first iteration the following probabilities are calculated as follows:

$\hat{P}\ (t_n = [\text{aj}]\ ,\ t_{n+1} = [\text{npl}]) = 0.15129$

$\hat{P}\ (t_n = [\text{av}]\ ,\ t_{n+1} = [\text{npl}]) = 0.1008$

$\hat{P}\ (t_n = [\text{nsg}]\ ,\ t_{n+1} = [\text{npl}]) = 0.21435$

After calculating the probabilities, the difference between the two highest probabilities is calculated as follows:

0.21435 - 0.15129 = 0.06306
As it is clear, this difference is lower than our threshold: 0.06306 < 0.07. So, we go on the algorithm and calculate the tri-gram probabilities of the second iteration as follows:

$\hat{P}$ (t_n = [aj] , t_{n+1} = [npl]| t_{n-1} = [co]) = 0

$\hat{P}$ (t_n = [av] , t_{n+1} = [npl]| t_{n-1} = [co]) = 0.00639

$\hat{P}$ (t_n = [nsg] , t_{n+1} = [npl]| t_{n-1} = [co]) = 0.079228

We compare the difference between the two highest probabilities and our threshold. Considering the following inequation:

0.072836 > 0.07

The algorithm stops at this stage without going to the subsequent iteration, and the highest probability in this iteration is considered as the one in which the tag of W_n is the correct tag. We do not go on the algorithm any further and take [nsg] as the correct tag for W_n .

2.4. Smoothing

When calculating the probabilities a problem may arise from sparseness of data especially for tri-gram and four-gram levels of iteration. Smoothing the probability estimation for each distribution is usually used in order to avoid sparse data estimation problems. As we know, there are possible but unseen subsequent tags for each tri-gram and onward that naturally may get the probability of zero because of the corresponding subsequent tags has not occurred in the corpus. So smoothing has been here used to reduce this undesirable effect of the zero probabilities. Different methods of smoothing can be applied according to different applications. Some of these methods are the additive method [24], the Good-Turing method [25], the Jelinek-Mercer method [26], Katz method [27], and Linear Interpolation method [28]. Among these methods, we preferred to use the last one, namely the Linear Interpolation method, because in addition to being a simple method, it has the advantage of being more scientific than the additive method. Moreover, it can best be adopted with our needs (handling three-gram and onwards) in this experiment. The first iteration or bi-gram level of analysis needs no smoothing since the sufficiently large examples for each two successive tags can be found in our corpus. In this respect we estimate the tri-gram and four-gram probabilities in (3.5) and (3.6), respectively as follows. (Smoothing the higher levels of analysis (four-gram onwards) will be similar to the two given formulas.):

(5) $$P(t_n \mid t_{n+1}, t_{n-1}) = \lambda_1 \hat{P}(t_n) + \lambda_2 \hat{P}(t_n \mid t_{n+1}) + \lambda_3 \hat{P}(t_n \mid t_{n+1}, t_{n-1})$$

(6) $$P(t_n \mid t_{n+1}, t_{n-1}, t_{n-2}) = \lambda_1 \hat{P}(t_n) + \lambda_2 \hat{P}(t_n \mid t_{n+1} + \lambda_3 \hat{P}(t_n \mid t_{n+1}, t_{n-1}) + \lambda_4 \hat{P}(t_n \mid t_{n+1}, t_{n-1}, t_{n-2})$$

$\hat{P}$ s are maximum likelihood estimates of the probabilities. In formula (3.5), $\lambda_1 + \lambda_2 + \lambda_3 = 1$, and in the same way, in formula (3.6), $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$.

3. Experimental Results

We tested our algorithm on a corpus containing 1000 Persian words (the test corpus) extracted from Ettela'at Newspaper. We separated disambiguation process of the ambiguous words in the test corpus according to the criterion of K^{th} iteration at which the algorithm should stop. Table 2 demonstrates different levels of iteration which correspond to the n-gram levels of analysis along with the percent of the ambiguous words which have been disambiguated in each level.

Table 2. Accuracy gained in different iteration levels

iteration	first	second	third	onward
percent of ambiguous words disambiguated	21.9	44.5	26.6	2.3
accuracy	89.2	92.5	96.2	93.3

As the table shows a high percent of ambiguous words are disambiguated in the second iteration or tri-gram level of analysis and then algorithm stops in that level. However, when we come to the matter of accuracy of disambiguation, the third iteration or four-gram level of analysis gets the highest accuracy in this respect. The total accuracy of the program calculated as 92.8.

4. Related Work

Although the fundamentals of the method presented in this paper have been inspired from the unsupervised method originally proposed by Brill [29], [20], it has some considerable advantages over Brill's. In unsupervised learning method of Brill which uses some transformation-based rules, the method cannot be referred to as a completely unsupervised one but partially supervised, because at least for extracting the required rules a tagged training corpus, even if small, is needed. But our method toward POS tagging can be regarded as totally unsupervised without requiring any tagged corpus. The Model presented by this paper is better suited for words' properties of a language like Persian than any other proposed model.

5. Conclusion and Further Developments

In this experiment we present a rather novel method for automatic POS tagging of Persian texts which has not already be applied to any other language. The results gained from the experiment is very encouraging especially for tagging a language such as Persian for which there is no large annotated corpus in any form.

In this method we considered only tags not individual words, so it has the advantage of being domain-independent one and can be used for tagging texts from various fields. As far as the method requirement is only having a large unannotated corpus, it can be applied to any other language for which large online texts and a machine-readable dictionary are available. This

method cannot cope with determining POS of the unknown words. However, these kinds of words will also be handled in future work inspiring of some related works [30] and using the iterative improved feedback method. Another future plan of the authors is to extract some transformation rules from the small available annotated corpus of Persian and carry out the tagging process using the method proposed by [28] for Persian. Comparing the results of the two methods, namely, IIF method and transformation-based method, we can decide on choosing the best unsupervised method for automatic tagging of Persian texts in large scales.

References

[1] Mosavi Miangah, T. and A. Delavar Khalafi, (2005). Word sense disambiguation using target language corpus in a machine translation system. Literary and Linguistic Computing, 20(2): 237-249.

[2] Cutting, D., J. Kupiec, J. Pederson, P. and Sibun, (1992). A practical part of speech tagger. In: Proceeding of the Third Conference on Applied Natural Language Processing, ACL: 133- 140, Trento, Italy.

[3] Masayuki, A. (2003). Corpus-based Japanese morphological analysis. Doctor's Thesis, Nara Institute of Science and Technology, NAIST-IS-DT0161001.

[4] Mosavi Miangah, T. (2006). Automatic lemmatization of Persian words. Journal of Quantitative Linguistics, 13 (1): 1-15.

[5] Conrad, S. M. (1999). The importance of corpus-based research for language teachers. System, 27:1-18.

[6] Sutsumi, J. et al., (1994). Multilingual system of machine translation based on statistical information. (in Russian). In: Proceeding of QUALICO-94.

[7] Braschler, M. and P. Schauble, (2000). Using corpus-based approaches in a system for multilingual information retrieval. Information Retrieval, 3:273-284.

[8] Kempe, A. (2000). Part of speech tagging with two sequential transducers. In: Proceeding of CLIN-2000: 88-96, Tliburg, The Netherland.

[9] Greene, B. B. and G. M. Rubin, (1971). Automatic grammatical tagging of English. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island. Tenth European Summer School in Logic, Language and Information (ESSLL-98), Student Session: 43-50.

[10] Koskenniemi, K. (1990). Finite-state parsing and disambiguation. In: Karlgren, H. editor, COLING-90: 229-232, Helsinki University.

[11] DeRose, S. (1988). Grammatical category disambiguation by statistical optimization. Computational Linguistics, 14:31-39

[12] Brants, T. (2000). TnT - A statistical part of speech tagger. ANLP-2000: 224-231.

[13] Merialdo, B. (1994). Tagging English text with a probabilistic model. Computational Linguistics, 20(2): 155-172.

[14] Assi, S. M. and M. Haji Abdolhosseini, (2000). Grammatical tagging of a Persian Corpus. International Journal of Corpus Linguistics, 5 (1): 69-81.

[15] Schuetze, H. (1995). Distributional part-of-speech tagging. From texts to tags: Issues in Multilingual Language Analysis. Online *proceedings of the ACL SIDGAT Workshop.* On the Internet at http:// xxx.lanl.gov/find/cmp-lg.

[16] Megerdoomian, K. (2004). Developing a Persian part-of-speech tagger. In: Proceedings of First Workshop on Persian Language and Computers. Invited Talk. Tehran University, Iran. May 25-26, 2004.

[17] Baum, L. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. Inequalities 3:1-8.

[18] Becker, M. (1998). Unsupervised part of speech tagging with extended templates. In:

[19] Kupiec, J. (1992). Robust part of speech tagging using a hidden Markov model. Computer speech and Language, 6.
[20] Brill, E. (1997). Unsupervised learning of disambiguation rules for part of speech tagging. In: Natural Language Processing Using Very Large Corpora, Kluwer Academic Press.
[21] Tzoukermann, E. and D. R. Radev, (1996). Using word class for part of speech disambiguation. In: 4th Workshop on Very Large Corpora, Special Interest Group for Linguistic Data and Corpus-based Approaches (SIGDAT) of the ACL , Copenhagen, Denmark: 1-13.
[22] Marcus, M., B. Santorini, and M. A. Marcinkiewicz, (1993). Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, 19(2): 313-30.
[23] Kreyszig, E. (1979). Advanced engineering mathematics, JOHN WILEY AND SONS, Forth Edition, USA.
[24] Gale, W. A. and K. W. Church, (1994). What's wrong with adding one? In Oostdijk, N. and de Haan, P. editors, Corpus-based research into language. Rodolpi, Amsterdam.
[25] Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. Biometrika, 40:237-264.
[26] Jelinek F. and R. L. Mercer, (1980). Interpolated estimation of markov source parameters from sparce data. In: Proceeding of the Workshop Pattern Recognition in Practice: 381-397, Amsterdam, North-Holland.
[27] Katz, S. M. (1987). Estimation of probabilities from aparse data for the language model component of a speech recognizer. IEEE Transactiona on Acoustics, Speech and Signal Processing, ASSP-35(3): 400-401, March.
[28] Gerald, C. F. and P. O. Wheathley, (1984). Applied numerical analysis. Addison-Wesley Publishing Company, USA.
[29] Brill, E. (1992). A simple rule-based part of speech tagger. In: Proceeding of the Third Conference on Applied Natural Language Processing, ACL, Trento, Italy.
[30] weischedel, R., M. Meteer R. Schwartz, L. Ramshaw and J. Palmuzzi, (1993). Coping with ambiguity and unknown words through probabilistic models. Computational Linguistics, 19(2): 359-82.