2007

Statistical POS tagging experiments on Persian text

F. Raja
University of Tehran, Iran

S. Tasharofi
University of Tehran, Iran

Farhad Oroumchian
University of Wollongong in Dubai, farhado@uow.edu.au

Publication Details

Raja, F, Tasharofi, S and Oroumchian, F, Statistical POS tagging experiments on Persian text, Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages, Stanford, California, 21-22 July 2007. Original conference information available here

Statistical POS Tagging Experiments on Persian Text

Fahimeh Raja
Faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran
f.raja@ece.ut.ac.ir

Samira Tasharofi
Faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran
stasharofi@ut.ac.ir

Farhad Oroumchian
The College of IT, University of Wollongong in Dubai
FarhadO@uowdubai.ac.ae

Abstract

Part-Of-Speech (POS) tagging is the process of marking-up the words in a text with their corresponding parts of speech. It is an essential part of text and natural language processing. There are many models and software for POS tagging in English and other European languages. Little work has been done on POS tagging of Persian language which uses Arabic script for writing. In these experiments we want to see how effective would be if we just applied a POS tagger from a language such as English to Persian. Although English and Persian are both Indo-European languages but they have subtle differences. This paper presents creation of a POS tagged corpus for evaluation purposes and evaluation of a statistical tagging method on Persian text. The results show that an overall tagging accuracy between 96.4% and 96.9% is achievable without the need to add any Persian linguistic knowledge to the tagging process. In This study we also looked at the effect of the size of training and test corpora on the accuracy of POS tagging.

1 Introduction

Many natural language processing (NLP) tasks require the accurate assignment of Part-Of-Speech (POS) tags to previously unseen text for preprocessing. So, they use a software called POS tagger which assigns a (unique or ambiguous) POS tag to each token in the input and passes its output to the next processing level, usually a parser or indexer. Furthermore, there is an interest in POS tagging for corpus annotation projects, which create valuable linguistic resources by a combination of automatic processing and human correction.
For both applications, a tagger with the highest possible accuracy rate is required. The debate about which paradigm solves the POS tagging problem best is not finished. Due to the availability of large corpora which have been manually annotated with POS information, many taggers use annotated text to "learn" either probability distributions or rules and use them to automatically assign POS tags to unseen text.

Some studies (Halteren et al., 1998; Volk et al., 1998) (Cutting et al., 1992; Schmid, 1995; Ratnaparkhi, 1996) suggest the statistical approaches yield better results than finite-state, rule-based, or memory-based taggers (Brill, 1993; Daelemans et al., 1996). Among the statistical approaches, the Maximum Entropy framework has a very strong position. In (Zavrel et al., 1999) it is shown that the combining the Markov models with a good smoothing technique along with handling of unknown words improves the performance. The TnT (Brants, 2000) tagger which is proposed by Thorsten Brants is based on this approach. In literature, the TnT efficiency is reported to be as one of the best and fastest on diverse languages such as German (Brants, 2000), English (Brants, 2000; Mihalcea, 2003), Slovene (Dzeroski et al., 2000) and Spanish (Carrasco et al., 2003).

On the other hand, it is always interesting to see how a method which is used in one language works on another language. This helps in providing insight into the nature of different languages. Persian (Farsi) is one of the important languages in Middle East. It is spoken in Iran, Tajikistan and

parts of Afghanistan. Although some efforts have been made on parsing and processing of Persian language, NLP of Persian is still in its early stages. There is a debate on how much linguistic information is needed to be added to a tagger in order to have an acceptable performance. In some experiments, researchers have used post processing of statistical taggers output for correcting the tags based on simple linguistic rules. In this paper, we want to show that with better statistical approaches we can achieve similar results and there is no need for post processing. The main problem in training statistical taggers is creating an annotated or tagged corpus. We used BijanKhan's tagged corpus (BijanKhan, 2004) for creating different sizes of training and test sets. However this corpus is built for other purposes and has very fine grained tags which are not suitable for POS tagging experiments. Therefore, we had to modify and simplify the tag set and reprocess the corpus in order to create a reasonable test corpus for POS experiments.

In the rest of this paper, first the Markov model and smoothing is discussed in Section 2. In Section 3 the TnT tagger is introduced. Then in Section 4 the creation of the test corpus is explained. Section 5 presents the evaluation process. Section 6 depicts the analysis of the results and finally, Section 7 presents conclusion and future works.

2 Markov and Smoothing

Often we are interested in finding patterns which appear over a period of time. These patterns occur in many areas such as sequences of words in sentences and the sequence of phonemes in spoken words. Frequently, patterns do not appear in isolation but as part of a series. Assumptions are usually made about the time based process; a common assumption is that the process's state is dependent only on the preceding N states, and then we have an order N Markov model (Thede et al., 1999).

A Markov Model (MM) is a probabilistic process over a finite set of states which can be used to solve classification problems that have an inherent state sequence representation. The model can be visualized with its states connected by a set of transition probabilities indicating the probability of traveling between two given states. A process begins in some state and "moves" to a new state as dictated by the transition probabilities. As the process enters each state, one of a set of output symbols is emitted by the process. Exactly which symbol is emitted is determined by a probability distribution that is specific to each state. The output of the MM is a sequence of output symbols (Thede et al., 1999).

When using an MM to perform POS tagging, the goal is to determine the most likely sequence of tags (states) that generates the words in the sentence (sequence of output symbols). In other words, given a sentence S, calculate the sequence U of tags that maximizes P(S|U). Therefore the transition probability is the probability that tag tj follows ti. This probability can be estimated using data from a training corpus. The Viterbi algorithm is a common method for calculating the most likely tag sequence when using an MM. This algorithm is explained in detail in (Lawrence et al., 1989).

While MM is a precise approximation of the underlying probabilities, these probabilities usually cannot directly be used because of the sparse data problem. This means that there are not enough instances to reliably estimate the probability. Moreover, setting a probability to zero causes the probability of the sequence to be set to zero. In an attempt to avoid sparse data estimation problems, the probability estimated for each distribution is smoothed. There are several methods of smoothing discussed in the literature. These methods include linear interpolation (Brants, 2000), the Good-Turing method (Good, 1953), and the Katz method (Katz, 1987). These methods are all useful smoothing algorithms for a variety of applications (Thede et al., 1999).

3 The TnT Tagger

Brants‘s TnT (Trigrams'n'Tags) tagger (Brants, 2000) is a statistical POS tagger, trainable on different languages and virtually any tag set. The component for parameter generation is trained on a tagged corpus. The system incorporates several methods of smoothing and of handling unknown words. TnT is not optimized for a particular language; instead, it is optimized for training on a large variety of corpora. The tagger is an implementation of the Viterbi algorithm for second orders Markov models. The main paradigm used for smoothing is linear interpolation; the respective weights are determined by deleted interpolation.

Unknown words are handled by a suffix trie and successive abstraction. Average POS tagging accuracy reported for various languages is between 96% and 97%, which is at least as good as the state of the art results found in the literature. The accuracy for known tokens is significantly higher than for unknown tokens. For example in experiments with German newspaper data, the result for seen words (the words in its lexicon) is 11% better than for the new words (97.7% vs. 86.6%). It should be mentioned that the accuracy for known tokens is high even with very small amounts of training data (Brants, 2000).

4 The Corpus

The corpus which was used in this work (Oroumchian, 2006) is a part of the BijanKhan's tagged corpus (BijanKhan, 2004), which is maintained at the Linguistics laboratory of the University of Tehran. The corpus is gathered from daily news and common texts. It contains 2598216 tokens and tagged with 550 different tags. The tags are organized in a tree structure. This vast amount of tags are used to achieve a fine grained POS tagging, i.e. a tagging that discriminates the subcategories in a general category. However, most of the tools for POS tagging do not work with a large set of tags. In order to make the tagging process more feasible, we decided to reduce the size of our tag set. We performed a statistical analysis of the corpus to see how many times each tag appears in the corpus. Then we decided to combine the infrequent tags in a meaningful way.

BijanKhan's corpus has a good representation for tags; each tag in the tag set follows a hierarchical structure. Each tag name includes the names of its parent tags. Each name starts with the name of the most general tag and follows by names of the subcategories until it reaches the name of the leaf tag. For example, the tag "N_PL_LOC" contains three levels; "N" at the beginning stands for noun; the second part, "PL" shows the plurality of the tag, and the last part, “LOC”, illustrates that the tag is about locations. For another example, the tag "N_PL_DAY" demonstrates a noun that is plural and describes a date.

The tag set reduction was done according to the following four steps:

1. In the first step, we reduced the depth of the hierarchy because the tags were more specific than we needed and this specificity caused them to have few utterances. We reduced the tags with three or more levels in hierarchy to two-level ones. Hence, both of the above examples "N_PL_LOC" and "N_PL_DAY" will reduce to a two-level tag, namely “N_PL”. The new tag shows plural nouns. After rewriting all the tags in the corpus in this manner, the corpus contained only 81 different tags.
2. Among the remaining tags, there were a number of tags that described numerical entities. After close examination of these tags, it was realized that many of them are not correct and are product of the mistakes in the tagging process. In order to prevent decreasing the accuracy of our POS tagger, all these tags were renamed to “DEFAULT” tag. So, the number of tags in the tag set reduced to 72 tags.
3. In the third step, some of the two-level tags were also reduced to one-level tags. Those were tags that appeared in the corpus rarely but were unnecessarily too specific. Examples of these are conjunctions, morphemes, prepositions, pronouns, prepositional phrases, noun phrases, conditional prepositions, objective adjectives, adverbs that describe locations, repetitions and wishes, quantifiers and mathematical signatures. By this modification, the number of tags reduced to 42.

In this step we reduced the tags that appeared rarely in the corpus. These are noun (N) and short infinitive verbs (V_SNFL). We consider the semantic relationship between these tags and their corresponding words. For example, since the words with tag “N” are single words, we replace “N” with “N_SING”. Also because the meaning of the “V_SNFL” tag is not similar to any other tags in the corpus, we simply removed it from the corpus. After this stage, 40 tags remained in our final tag set

5 Experimental Process

In the majority of the POS tagging approaches, the sample is often subdivided into "training" and "test" sets. The training set is generally used for learning, i.e. fitting the parameters of the tagger. The test set is for assessing the performance of the tagger.

In our experiments, we used different proportions of training and test sets to see the effect of

size of training on the overall results of experiments. We used random samples of 90%, 80%, 70%, 60% and 50% of the corpus for training. In order to avoid accidental results, each experiment repeated five times. In each run, we selected the training and test sets randomly. For example five different samples of 70% for training and 30% for testing were taken and each sample was used for POS tagging with TnT software. Then the result of 5 runs was averaged and used as the result of that experiment.

6 Experimental Results

For the evaluation purpose, the tagged file was compared with the original manually tagged test file and the differences were recorded. Considering the tagging accuracy as the percentage of correctly assigned tags, we have evaluated the performance of the TnT tagger from two different aspects: (1) the overall accuracy (taking into account all tokens in the test corpus) and (2) the accuracy for known and unknown words, respectively. Since after training the tagger, it could be used on text other than the training text, it is interesting to know how it would cope with words that did not appear in its training.

Tables 1, 2, and 3 depict the results of the experiments. As previously mentioned, for each experiment we have five runs and the results in these tables are the average result of those five runs. Table 1 shows the percentage of seen words (words that exist in training set), number of tokens in the test set, the number of tokens correctly tagged and the accuracy for that experiment. Similarly, Table 2 shows the same for words that are new for the tagger. Table 3 shows the overall result for each experiment. To be consistent with other languages we focus on the usual size of 10% and 90% of the corpus for test and training sets, respectively:

1. The overall POS tagging accuracy is around 96.94%.
2. The accuracy for known tokens is significantly higher than that for unknown tokens (97.26% vs. 79.44%). It shows 17.82% points accuracy difference between the words seen before and those not seen before.
3. The overall accuracy depends on the size of test and training sets because with the increase in the size of training set (and consequently, the decrease in the size of test set) the percentage of the unknown tokens may decrease; therefore, the larger the size of training set, the higher the accuracy.

The difference between highest overall accuracy 96.94% achieved by 90% training and the lowest 96.47% using only 50% training is only 0.47% which is negligible. This shows even with small training sample, we can achieve very high accuracy.

Table 1. Known tokens results

Exp.	Percent	Tokens	Correct	Accuracy
1	98.21	226723	220519	97.26%
2	98.07	546870	531284	97.15%
3	98.02	771064	748567	97.08%
4	97.65	1025406	994536	96.99%
5	97.40	1263920	1225708	96.98%

Table 2. Unknown tokens results

Exp.	Percent	Tokens	Correct	Accuracy
1	1.79	4134	3284	79.44%
2	1.93	10761	8503	79.01%
3	2.17	17133	13316	77.72%
4	2.35	24661	19087	77.40%
5	2.60	33688	26056	77.35%

Table 3. Overall results

Exp.	Tokens	Correct	Accuracy
1	230857	223803	96.94%
2	557631	539787	96.80%
3	788197	761883	96.66%
4	1050067	1013623	96.53%
5	1297608	1251764	96.47%

In Table 4, the overall accuracy in experiment 1 is compared to the performance of TnT tagger for English, German and Spanish as reported in the literature. In this table the proportion of test and training sets for all languages are 10% and 90% respectively. As seen in the table, without a need to post processing and adding extra linguistic information, we have achieved similar results in line with other languages such as English and German.

Table 4. Compared results for different languages

Language	Unknown Tokens Percent	Known accuracy	Unknown accuracy	Overall accuracy
English	2.9%	97.0%	85.5%	96.7%
Germany	11.9%	97.7%	89.0%	96.7%
Spanish	14.4%	96.5%	79.8%	94.15%
Persian	1.79%	97.26%	79.44%	96.94%

7 Conclusion and Future Works

An evaluation of a statistical POS tagger known as TnT which implements Markov model and linear smoothing on Persian language has been presented.

In this work, a test collection for POS tagging was produced by reducing the tag set of a manually tagged corpus. The experiments were repeated several times for different sizes of test and training sets selected randomly from the collection.

The results of using TnT tagger on Persian text show that the highest overall accuracy of the taager is 96.94% when the size of test and training sets are 10% and 90% of the corpus respectively. Moreover, these results reveal that the accuracy for known words is higher than unknown words (about 18%). It also shows that the accuracy of TnT depends on the size of test and training sets. The smaller the training set, the lower the accuracy. However, it also suggests that with a small size of training text (only 50%), still we can achieve a respectable performance of overall 96.47%. That is only 0.50% less than the best overall result.

The results of using TnT on different languages indicates that the decisions made in TnT yield good results on a large variety of corpora.

In future, we will compare the Markov model with other taggers on Persian texts. Also, we would like to investigate how much improvement we can achieve if we add post processing of "unknown" words. For this purpose we intend to use simple linguistic heuristics of Persian language.

Acknowledgement Many thanks go to Thorsten Brants for his attention to our e-mails and giving us his very efficient and user friendly tool. We also would like to thank Dr. Faili for his helps in gathering and preparing the tagged corpus and Dr. BijanKhan for his valuable work in tagging the Persian texts and providing us with his corpus.

References

Mahmood BijanKhan. 2004. The Role of the Corpus in Writing a Grammar: An Introduction to a Software. *Iranian Journal of Linguistics, 19*.

Thorsten Brants. 2000. TnT – a Statistical Part-of-Speech Tagger. *Proc. 6th Conf. on applied natural language processing (ANLP)*, Seattle, WA.

Eric Brill. 1993. A corpus-based approach to language learning, *Ph.D. Dissertation*, Department of Computer and Information Science, University of Pennsylvania.

Raúl M. Carrasco and Alexander Gelbukh. 2003. Evaluation of TnT Tagger for Spanish. *Proc. 4th Mexican International Conf. on Computer Science (ENC'03)*.

Doug Cutting, Julian Kupiec, Jan Pealersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. *Proc. 3rd Conf. on Applied Natural Language Processing (ACL)*, 133–140.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. A memory-based part of speech tagger-generator. *Proc. Workshop on Very Large Corpora*, Denmark.

Saso Dzeroski, Tomaz Erjavec, and Jakub Zavrel. 2000. Morphosyntactic Tagging of Slovene: Evaluating Taggers and Tagsets. *Proc. LREC*, Athens.

I.J. Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika, 40*:237-264.

Hans V. Halteren, Jakub Zavrel, and Walter Daelemans. 1998. Improving data driven word class tagging by system combination. *Proc. International Conf. on Computational Linguistics COLING*, Montreal, Canada, 491-497.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing, 35*:400-401.

Rada Mihalcea. 2003. Performance Analysis of a Part of Speech Tagging Task. *Proc. Computational Linguistics and Intelligent Text Processing*, México.

Farhad Oroumchian, Samira Tasharofi, Hadi Amiri, Hossein Hojjat, and Fahimeh Raja. 2006. Creating a Feasible Corpus for Persian POS Tagging. *Technical Report, no. TR3/06*, University of Wollongong (Dubai Campus).

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 257-286.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. *Proc. Conf. on Empirical Methods in Natural Language Processing EMNLP*, USA.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. *Helmut Feldweg and Erhard Hinrichts*, Tubingen.

Scott M. Thede and Mary P. Harper. 1999. A second-order Hidden Markov Model for part-of-specch tagging. *Association for Computational Linguistics Morristown*, USA, 175-182.

Martin Volk and Gerold Scheider. 1998. Comparing a statistical and a rule-based tagger for german. *Proc. KONVENS*, Bonn, 125–137.

Jakub Zavrel and Walter Daelemans. 1999. Evaluatie van part-of-speech taggers voor het corpus gesproken nederlands. *CGN technical report*, Katholieke Universiteit Brabant, Tilburg.