2007

Evaluation of part of speech tagging on Persian text

F. Raja
University of Tehran, Iran

H. Amiri
University of Tehran, Iran

S. Tasharofi
University of Tehran, Iran

M. Sarmadi
University of Tehran, Iran

H. Hojjat
University of Tehran, Iran

See next page for additional authors

Publication Details

Raja, F, Amiri, H, Tasharofi, S, Sarmadi, M, Hojjat, H and Oroumchian, F, Evaluation of part of speech tagging 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

Evaluation of Part of Speech Tagging on Persian Text

Fahimeh Raja Hadi Amiri Samira Tasharofi Mehdi Sarmadi Hossein Hojjat **Farhad Oroumchian**

Department of Electrical and Computer Engineering, University of Tehran
{f.raja,h.amiri,s.tasharofi,mehdi,h.hojjat}@ece.ut.ac.ir

University of Wollongong in Dubai
farhadO@uowdubai.ac.ae

Abstract

One of the fundamental tasks in natural language processing is part of speech (POS) tagging. A POS tagger is a piece of software that reads text in some language and assigns a part of speech tag to each one of the words. Our main interest in this research was to see how easy it is to apply methods used in a language such as English to a new and different language such as Persian (Farsi) and what would be the performance of such approaches. This paper presents evaluation of several part of speech tagging methods on Persian text. These are a statistical tagging method, a memory based tagging approach and two different versions of Maximum Likelihood Estimation (MLE) tagging on Persian text. The two MLE versions differ in the way they handle the unknown words. We also demonstrate the value of simple heuristics and post-processing in improving the accuracy of these methods. These experiments have been conducted on a manually part of speech tagged Persian corpus with over two million tagged words. The results of the experiments are encouraging and comparable with the other languages such as English, German or Spanish[1].

1 Introduction

Part-of-speech (POS) tagging selects the most likely sequence of syntactic categories for the words in a sentence. It determines grammatical characteristics of the words, such as part of speech, grammatical number, gender, person, etc. This task is not trivial since many words are ambiguous: for example, English word "fly" can be a noun (e.g. a fly is a small insect) or a verb (e.g. the birds will fly north in summer). In recent years, there has been a growing interest in data-driven machine-learning disambiguation methods, which can be used in many situations such as tagging.

There are many different models for tagging which differ on their internal model or the amount of training or processing of information they need. Although there are many models and implementations available for the task of tagging, most of them are designed for and tested on English texts; less work has been done on tagging and tagger evaluation for languages like Persian that have quite different properties and script. In this paper we present the evaluation of a statistical part of speech tagger based on Markov chains, a memory based tagging approach and two different versions of Maximum Likelihood Estimation (MLE) tagging on Persian texts. For the Markov chains model, we took advantage of the TnT tagger which is written by Thorsten Brants and in literature its 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 and Gelbukh, 2003). Memory-based taggers are trained with a training set and they use learned information to tag a new text. In Maximum Likelihood Estimation approach for every word in the training set the tag which is assigned to the word more than other tags will be applied.

The main problem in training taggers is creating an annotated or tagged corpus. We used BijanKhan's tagged corpus (BijanKhan, 2004) for training and testing. However this corpus is built for other purposes and has very fine grained tags which are

[1] This work was partially supported by Iranian Telecommunication Research Center (ITRC) contract No. 500/12204.

not suitable for POS tagging experiments. Therefore, we made changes and prepared the corpus for the POS experiments.

In the rest of this paper, Section 2 describes the test corpus and our changes on it to make it appropriate for POS tagging. The TnT tagger is introduced in Section 3. In Section 4 and 5 memory-based POS tagging and Maximum Likelihood Estimation tagging is explained respectively. Section 6 discuses the accuracy of above POS methods for unknown words and shows some post-processing techniques to improve the accuracy of the methods for unknown words. Section 7 compares the results of different approaches and finally, Section 8 presents conclusion and future works.

2 The Corpus

The corpus which was used in this work 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 was tagged with a rich set of tags consisting of 550 different tags. The tags are organized in a tree structure. This vast amount of tags are used to achieve a fine grained part-of-speech tagging, i.e. a tagging that discriminates the subcategories in a general category. This large number of tags makes any machine learning process impracticable. So, we decided to reduce the number of tags (Oroumchian et al., 2006) as described below.

2.1 Selecting the Suitable Tags

BijanKhan's corpus uses 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. As 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 as follows. We considered all the tags with three or more levels in hierarchy and changed them to two-level ones. Hence, both of the above examples will reduce to a two-level tag, namely “N_PL”. The new tag shows that they are plural nouns. After rewriting all the tags in the corpus in this manner, the corpus contained only 81 different tags.
2. Among the 81 remaining tags in the corpus, 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 part-of-speech tagger, all these tags were renamed to “DEFAULT” tag. So, the number of tags in the tag set was reduced to 72 tags in this step.
3. In the third step, some of the two-level tags were also reduced to one-level tags. Those were tags that rarely appeared in the corpus and were unnecessarily too specific. Examples of these tags 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 was reduced to 42.
4. In this step we removed the tags that appeared rarely in the corpus. These are noun (N) and short infinitive verbs (V_SNFL). We considered the semantic relationship between these tags and their corresponding words. For example, since the words with tag “N” are single words, we replaced the “N” tag with the “N_SING” tag. 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, there were only 40 tags remained in the corpus.

2.2 Statistical Analysis of the Corpus

Table 1 shows the tags and their corresponding frequencies in the corpus.

Studying the table carefully reveals that the tag “N_SING” (singular noun) is the most frequent tag in the corpus. On the other hand, the “NN” tag with only twice occurrence is the least frequent tag.

Tag Name	Frequency in Corpus	Probability
ADJ	22	8.46826E-06
ADJ_CMPR	7443	0.002864966
ADJ_INO	27196	0.010468306
ADJ_ORD	6592	0.002537398
ADJ_SIM	231151	0.088974829
ADJ_SUP	7343	0.002826473
ADV	1515	0.000583155
ADV_EXM	3191	0.001228282
ADV_I	2094	0.000806024
ADV_NEG	1668	0.000642048
ADV_NI	21900	0.008429766
ADV_TIME	8427	0.003243728
AR	3493	0.001344528
CON	210292	0.080945766
DEFAULT	80	3.07937E-05
DELM	256595	0.098768754
DET	45898	0.017667095
IF	3122	0.001201723
INT	113	4.34961E-05
MORP	3027	0.001165155
MQUA	361	0.000138956
MS	261	0.000100464
N_PL	160419	0.061748611
N_SING	967546	0.372428585
NN	2	7.69842E-07
NP	52	2.00159E-05
OH	283	0.000108933
OHH	20	7.69842E-06
P	319858	0.123119999
PP	880	0.00033873
PRO	61859	0.023810816
PS	333	0.000128179
QUA	15418	0.005934709
SPEC	3809	0.001466163
V_AUX	15870	0.006108693
V_IMP	1157	0.000445353
V_PA	80594	0.031022307
V_PRE	42495	0.01635721
V_PRS	51738	0.019915033
V_SUB	33820	0.013018022
Max	967546	0.372428585
Min	2	7.69842E-07
Sum	2597937	1

Table 1The tags distribution

2.3 Providing Test and Training Sets

After recreating the corpus with only 40 different tags, it was subdivided into "training" and "test" sets. The training set was used for learning, i.e. fitting the parameters of the taggers. The test set was used for assessing the performance of the taggers.

In our experiments, training and test sets were created by randomly dividing the corpus into two parts with an 85% to 15% ratio. In order to avoid accidental results, each experiment repeated five times. Then the result of 5 runs was averaged and used for drawing conclusions

Table 2 shows the number of tokens in each set. The training and test columns show the number and the percentage of the tokens that is used for the training and test sets. For example in run 1 (with the total of 2,598,216 tokens), we used 84.52 percent of the tokens (2,196,166 tokens) for training and the remaining (402050 tokens) for testing the methods.

In Table 3, we show the percentage of the known words (seen before in the training set) and unknown words (words that are new for the tagger) in the test set.

Run	Training Tokens/Percent	Test Tokens/Percent	Total
1	2196166 / 84.52	402050 / 15.47	2598216
2	2235558 / 86.04	362658 / 13.96	2598216
3	2192411 / 84.38	405805 / 15.61	2598216
4	2178963 / 83.86	419253 / 16.13	2598216
5	2186811 / 84.16	411405 / 15.83	2598216
Avg.	2197982 / 84.59	400234.2 / 15.40	

Table 2 : Test and Training Sets

Run	Known Words Percentage	Unknown words Percentage
1	97.97	2.03
2	98.06	1.94
3	97.92	2.08
4	97.91	2.09
5	97.97	2.03
Avg.	97.96	2.04

Table 3 Percentage of Known and Unknown words in the Test Set

3 The Markov modle (TnT Tagger)

One of the robust statistical models in Part of Speech tagging is using Markov chains in order to

estimate the probabilities of assigning particular tags to words based on the words surrounding it. For this experiments we took advantage of Brants's TnT (Trigrams'n'Tags) tagger (Brants, 2000) which is a statistical part of speech tagger, trainable on different languages and with 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 part-of-speech 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 German newspaper data, when the words seen before (the words in its lexicon) the results are 11% points better than for the words not seen before (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).

3.1 TnT Experimental Results

For the evaluation purpose, the tagged test file was compared with the original manually tagged file and the differences were recorded.

Considering the tagging accuracy as the percentage of correctly assigned tags, we have evaluated the performance of the taggers 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. It is interesting to know how it would cope with words that did not appear in its training.
Table 4 depicts the results of the experiments for known and unknown words and the overall accuracy of the tagger in each run. In general:

1. The overall part-of-speech tagging accuracy of TnT tagger is around 96.64%.
2. The accuracy of known tokens is significantly higher than that of unknown tokens (97.01% vs. 77.77%). It shows 19.24% points accuracy difference between the words seen before and those not seen before.

Run	Known words	Unknown words	Overall
1	96.94%	75.12%	96.52%
2	97.18%	80.09%	96.86%
3	96.96%	77.34%	96.57%
4	96.96%	77.69%	96.58%
5	97.03%	78.62%	96.67%
Avg.	97.01%	77.77%	96.64%

Table 4 TnT Accuracy

4 Memory-Based POS Tagging

Memory-based POS tagging uses some specifications of each word such as its possible tags, and a fixed width context (tag of previous words which are not ambiguous) as *features*. A memory-based tagger uses memory-based learning algorithms to learn from a training set and then tags the test set with knowledge of what is learned previously. Memory based learning is also known as Lazy Leaning, Example Based learning, or Case Based Learning (Daelemans et al., 1996). Usually memory based learners build a tree like data structure of learned instances kept in memory. And when a new instance is added, they use some similarity metrics to measure the distance between the features of the new item with features of existing classes to classify and place the new instance in the data structure (Daelemans et al., 1996).

Two main algorithms for memory based learning are "Weighted MBL: IB1-IG" (Daelemans and Van den Bosch, 1992) and "Optimized weighted MBL: IGTREE" (Daelemans, Van den Bosch, and Weijters, 1997).

IB1-IG is a memory-based learning algorithm that builds a database of instances during learning. After the instance base is built, new instances are classified by matching them to all instances in the instance base, and by calculating with each match the distance between the new instance X and the memory instance Y. In IB1-IG the distance metric is a weighted sum of the distance per feature (Zavrel and Daelemans, 1999). Because the search for the nearest neighbors in IB1-IG is time consuming and POS taggers has to run very fast, IGTREE proposed use of decision trees for search. In IGTREE the instance memory is reconstructed in such way that it contains the same information

as before but in a compressed decision tree structure (Zavrel and Daelemans, 1999).

4.1 Memory Based Tagger Experimental Results

In our experiments we used MBT which is a tool for memory based tagging. MBT generates a tagger by working through the annotated corpus and creating three data structures: a *lexicon*, associating words to tags as evident in the training corpus, a case base for *known words*, and a case base for *unknown words.* Case Bases are compressed using IGTREE for efficiency (Daelemans et al., 1996).

Selecting appropriate feature sets for known and unknown words has important impact on the accuracy of the results. After different experiments, we chose "*ddfa*" as the feature set for known words. First and second *d*s stand for disambiguated tag of two previous words of the current word in the text and the *f* means the focus word, the word which we want to find its appropriate tag. Finally the *a* stands for one ambiguous word after the current word. That is, choosing the appropriate tag for each known word, based on the tags of two words before it and the possible tags of the word after it (Zavrel and Daelemans, 1999; Zavrel and Daelemans, 1997).

The feature set chosen for unknown word is *"dfass"* [2]. As known words features, *d* is the disambiguated tag of the word before current word, *a* stands for ambiguous tags of the word after current word, the *ss* represents two suffix letters of the current word.

The results on the 5 test sets, described in section 2, are depicted in Table 5. There is about 20% difference (96.86% vs. 75.15%) between accuracy of POS tagging for known and unknown words. However since there not that many unknown words in this collection, this difference has not affected the overall performance of the system.

Run	Known words	Unknown words	Overall
1	96.43%	88.55%	96.27%
2	96.72%	91.80%	96.62%
3	96.98%	64.23%	96.30%
4	97.04%	66.18%	96.39%
5	97.10%	68.31%	96.51%
Avg.	96.86%	75.15%	96.42%

Table 5 MBT Accuracy

[2] The *f* in unknown words features indicates position of the focus word and it is not included in actual feature set.

5 Maximum Likelihood Estimation

As a bench mark for POS tagging, we chose Maximum Likelihood Estimation (MLE) approach for its simplicity and ease of implementation. In this approach, for every word in the training set we calculated the tag which is assigned to the word more than the other tags (Allen, 1995). For this purpose, we calculated the maximum likelihood probabilities for each tag assigned to each word and then we pick a tag with the greatest maximum likelihood probability for each word and make it the only tag assignable to that word. We call this tag the *designated* tag for that word.

Table 6 shows the results of MLE for known words, unknown words, and the overall accuracy respectively.

Run	Known words	Unknown words	Overall
1	96.50%	12%	94.55%
2	96.78%	16%	94.91%
3	96.53%	18%	94.53%
4	96.53%	9%	94.51%
5	96.64%	23%	94.68%
Avg.	96.60%	15%	94.63%

Table 6 MLE Accuracy

To obtain the result depicted in Table 6, we considered the "DEFAULT" tag as *designated* tag for unknown words. An analysis of failure after the experiments revealed that from all the "DEFAULT" tags assigned, at most 19 of them were correct and the rest were wrong. That is why the accuracy of this system on unknown words is very low (15%) in comparison with the other methods. Instead of the "DEFAULT" tag, we can choose to assign the most common tag in the corpus to the unknown words. The most frequent tag based on Table 1 is "N_SING" (Singular Noun) which appears 967546 times in the corpus. Table 7 shows the result of MLE with "N_SING" as designated tag. This approach improves the overall accuracy to 95.37% and boosts the accuracy of the unknown words to 54.11% which is still lower than other methods but more than 3 times better than before.

Run	Unknown words	Overall
1	52.60%	95.61%
2	56.63%	96.00%
3	51.49%	95.59%
4	55.48%	95.67%
5	54.34%	95.78%
Avg.	54.11%	95.73%

Table 7 MLE with "N_SING" as designated tag Accuracy

6 Heuristic Post Processing

In the previous section we reported on the application of different methods to Persian language without any particular adjustment for the language. In this section we discus how simple morphological heuristics about the Persian language can improve the accuracy of predicting the tags for the unknown words.

As depicted in Tables 6 and 7, the MLE method doesn't have an acceptable accuracy rate for the unknown words. Therefore we investigated the unknown words and their tags in the test collection. We realized that first; the correct tag for most of the unknown words is "N_SING". That explains why the MLE method that selects "N_SING" as designated tag works has better results. Second, some of the unknown words were plural nouns ("N_PL") which were incorrectly tagged as "DEFAULT" or "N_SING" by MLE. In Persian language, plural nouns end with substrings like "ها", "های", "ان","ات" etc. For example the word "نیمکت" (bench in English) is a singular noun ("N_SING") and "نیمکت ها" (benches) is its plural form ("N_PL"). Hence, we can post -process the output of the MLE method (or any other method) with a simple heuristic as: if a word ends with any of the plural suffixes it should be tagged as "N_PL". However, this solution doesn't work for all such words. As an example consider the word "مدرسه ات" (your school in English). This word has the substring "ات" at its tail but it is a single noun. So based on this heuristic it will be tagged incorrectly as "N_PL". Similar heuristics could be formed for many of the part of speech tags in Table 1. Table 8 lists part of speech tags with their most common suffixes (ending substrings) along with their frequency of occurrences in the test collection.

There were also some unknown words with "ب" (B in English) or "ن" (N in English) letters at their heads (starting character). The real tag for most of such words is "V_SUB", so for each unknown word that starts with the letter "ب" or "ن" we can choose the tag "V_SUB" as its designated tag. Similarly, all of the words that start with the prefix "می" (MI) or "نمی" (NeMI) can be tagged with the "V_PRS" (Present Verb) tags. These tags and their frequencies are listed in Table 9.

real tag of the unknown word	unknown word's tail morphemes	Number of occurrence
ADJ_CMPR	تر، تری	339
ADJ_SUP	ترین	251
N_PL	ها، های، هایی، ان، هایم، هایت، هایش، هایمان، هایتان، هایشان، ین، ات،ان	7052
V_PA	ام، ای، یم، ید، ند	686
V_PRE	ست	786

Table 8 Unknown Words Features (Tail)

Real tag of unknown word	unknown word's head morphemes/ letters	Number of occurrence
V_SUB	ب،ن	446
V_PRS	می، نمی	478

Table 9 Unknown Words Features (Head)

Hence, a new set of new models could be created based on the above post processing heuristics. Based on these heuristics we will post process the output of taggers and for unknown words, we will modify their tags based on these suffixes or prefixes. For example, by applying the above heuristic post-processing to the tags of the unknown words for the MLE-"DEFUALT" model, an average 19.33 percent improvement for unknown words, (19.48% versus 0.15%) can be observed. Applying the same heuristic post processing to the output of MLE-N-SING model will result in an average 11.64 percent improvement for unknown words. These results are depicted in Tables 10 and 11.

Run	Without Post processing	With Post processing	Improvement
1	12%	17.99%	17.87%

2	16%	19.27%	19.11%
3	18%	20.25%	20.07%
4	9%	18.89%	18.80%
5	23%	21.01%	20.78%
Avg.	15%	19.48%	19.33%

Table 10 Comparison of the accuracy of the MLE with "DEFAULT" as designated tag with and without Post Processing

Run	Without Post processing	With Post processing	Improvement
1	52.60%	63.55%	10.95%
2	56.63%	67.78%	11.15%
3	51.49%	64.20%	12.71%
4	55.48%	66.52%	11.04%
5	54.34%	66.72%	12.38%
Avg.	54.11%	65.75%	11.64%

Table 11 Comparison of the accuracy of the MLE with "N-SING" as designated tag with and without Post Processing

Table 12 shows the overall accuracy of both MLE methods after heuristic post processing. In general, by using this post-processing the overall accuracy of MLE method is improved 0.40 by using “DEFAULT” tag as designated tag and 0.24 by using “N_SING” as designated tag. Again, since the number of unknown words is not many, these improvements do not significantly affect the overall performance of the system.

Designated Tag	Accuracy	Improvement
"DEFAULT"	95.03%	0.40%
"N_SING"	95.97%	0.24%

Table 12 Overall Accuracy of MLE+Post-Processing

We also applied the post-processing to the Memory-Based tagging. The results are shown in Table 13. This results show 5.29 improvements on average.

Run	Without Post processing	With Post processing	Improvement
1	88.55%	91.68%	3.13%
2	91.80%	93.17%	1.37%
3	64.23%	73.06%	8.83%
4	66.18%	72.24%	6.06%
5	68.31%	75.41%	7.10%
Avg.	75.15%	81.11%	5.29%

Table 13 MBT +Post-Processing Results

Moreover, we applied the post-processing to TnT as well. In general, our results show that the weaker POS taggers benefited more from this heuristic post-processing.

7 A Comparison of the Different Approaches

Table 14 compares the overall results obtained in our experiments. The MLE approach that assigns the "DEFAULT" tag to the unknown words produces the least accurate results for Persian text. MLE approach that assigns "N-SING" tag to unknown words if combined with post-processing produces much better but still an average tagger. We believe this approach could be considered as a bench mark for lower end of part of speech taggers because it is built on reasonable but simple assumptions and heuristics. The best approaches are TnT and the memory based approach (MBT) when combined with post-processing. The MBT+post processing has the highest accuracy rate on the unknown words compared to the other methods.

Table 15 compares our results with the results reported on other languages. The accuracy obtained for the TnT and MBT+post processing models are comparable to the accuracy of part of speech taggers on the other languages.

Approach/Accuracy	Known Words	Unknown Words	Overall
MLE-DEFAULT	96.60%	0.15%	94.63%
MLE-N_SING	96.60%	54.11%	95.73%
MLE-DEFAULT+ Post-Processing	96.60%	19.48%	95.03%
MLE-N_SING +Post-Processing	96.60%	65.75%	95.97%
MBT	96.86%	75.15%	96.42%
MBT +Post-Processing	96.86%	81.11%	96.63%
TnT	97.01%	77.77%	96.64%

Table 14 Comparison of the results

Language	Known accuracy	Unknown accuracy
Persian (TnT)	97.01%	77.77%
Persian (MBT+Post-	96.86%	81.11%

Processing)		
English	97.0%	85.5%
Germany	97.7%	89.0%
Spanish	96.5%	79.8%

Table 15 The Results of other Languages

8 Conclusion and Future Works

This paper describes experiments conducted with Markov Model, Memory based and Maximum Likelihood approaches for POS tagging of Persian text. A POS corpus was created for these experiments and the taggers were trained on 85% of the corpus and were tested on the remaining 15%. The results show that with the statistical part of speech tagger (TnT) without prior linguistic knowledge, we can generate a reasonable POS tagger for Persian language. We also experimented with simple heuristics that could be applied in post-processing of the output of the taggers. These heuristics were based on modifying the tags for unknown words after examining a few prefix or suffix characters of the words. Our results show that these simple heuristics have significant impact on improving the tagging of the unknown words especially for the weaker models.

The overall and unknown word performance of memory based approach with post-processing and the TnT system without post processing are similar to that of the other languages such as English, German and Spanish.

In future we would like to continue these experiments with other types of Part of Speech tagging models and more heuristic post-processing. We also like to investigate the effect of the size of the training on the effectiveness of the taggers and build other test collections.

Acknowledgements

Many thanks go to Thorsten Brants for his attention to our e-mails and giving us his very efficient and user friendly tool. We 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 tagged corpus.

References

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

Jakub Zavrel and Walter Daelemans. 1997. *Memory-based learning: Using similarity for smoothing*. In Proc. Of 35th annual meeting of the ACL.

Jakub Zavrel and Walter Daelemans. 1999. *Recent Advances in Memory-Based Part of Speech Tagging*. VI Simposio Internacionale de Comunicacion.

James Allen. 1995. *Natural Language Understanding*. Second Edition. The Benjain/Cummings Publishing Company, Inc., Redwood City, California, USA.

Mahmood BijanKhan. 2004. The Role of the Corpus in Writing a Grammar: An Introduction to a Software. *Iranian Journal of Linguistics*, 19(2).

Rada Mihalcea. 2003. *Performance Analysis of a Part of Speech Tagging Task*. In Proc. Computational Linguistics and Intelligent Text Processing, Gelbukh A. Editor, Centro de Investigacin en Computacin IPN.

Raúl M. Carrasco and Alexander Gelbukh. 2003. *Evaluation of TnT Tagger for Spanish*. In Proc. Fourth Mexican International Conference on Computer Science ENC'03.

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

Thorsten Brants. 2000. *TnT – a Statistical Part-of-Speech Tagger*. In Proc. sixth conference on applied natural language processing ANLP-2000.

Walter Daelemans, Antal V. D. Bosch and Ton Weijters. 1996. *IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms*. In Aha,D.(ed.). AI Review Special Issue on Lazy Learning.