Towards combining rule-based and statistical part of speech tagging in agglutinative languages

Levent Altunyurt, Zihni Orhan, Tunga Güngör

***Abstract**: We present a composite part of speech tagger for Turkish which combines the rule-based and statistical approaches. The tagger makes use of word frequencies and n-gram statistics from a corpus. We use the output of a morphological analyzer in order to get more accurate results and also to eliminate the sparse data problem. In addition, we employ a heuristics about the position of words in the sentences. Although the experiments have been performed on a very small corpus, the results have shown that the use of a composite approach and heuristics improves the accuracy of the tagger.*

***Keywords:** agglutinative language, part of speech tagger, rule-based and statistical method*

1. Introduction

Part of speech tagging is the process of marking up the words in a text with their corresponding parts of speech reflecting their syntactic category. Depending on the degree of automation used in the tagging process, the taggers can be classified as supervised or unsupervised. Supervised taggers typically rely on pre-tagged corpora which serve as the basis for creating tools (dictionary, word/tag frequencies, tag sequence probabilities, rule set, etc.) to be used throughout the tagging process. Unsupervised models, on the other hand, are those which do not require a pre-tagged corpus, but instead use sophisticated computational methods to automatically induce word groupings (i.e. tag sets). Based on these automatic groupings, they either calculate the probabilistic information needed by stochastic taggers or derive the context rules needed by rule-based systems.

There have been generally two distinct approaches to part of speech tagging. The rule-based approaches use contextual information to constrain the possible part of speech tags [6] or to assign a part of speech tag to a word [1,7,9]. These rules are often known as context frame rules. For instance, a context frame rule for English might be as follows: “if the word is preceded by a determiner and followed by a noun, tag it as an adjective”. In addition to context information, many taggers use morphological information to aid in the disambiguation process. One such rule might be: “if the word ends in *-ing* and is preceded by an auxiliary verb, label it as a verb”. Some systems go beyond using contextual and morphological information by including rules pertaining to such factors as capitalization and punctuation. The usefulness of this type of information highly depends on the language being tagged. In German, for example, information about capitalization proves extremely useful in the tagging of unknown nouns.

The statistical (stochastic) approaches select the most likely interpretation based on the estimation of statistics from unambiguously tagged text. Either word frequencies or n-gram probabilities can be used as the criterion to be maximized. The most common algorithm for implementing an n-gram approach is the Viterbi Algorithm, which avoids the polynomial expansion of a breadth first search by trimming the search tree. The next level of complexity that can be introduced into a stochastic tagger combines the previous two approaches, using both tag sequence probabilities and word frequency measurements. These approaches use a Markov model [3,14], a meximum-entropy model [12], a hidden Markov model [2,5,11,13], or a perceptron model [4].

In this paper, we propose a composite approach for part of speech tagging in Turkish, which combines rule-based and statistical approaches as well as making use of some characteristics of the language in terms of heuristics. We use both word frequencies and n-gram (unigram, bigram, trigram) probabilities. An important feature of the work is utilizing a morphological analyzer. We combine the output of the morphological analyzer with stochastic methods in order to improve the accuracy of the system. Morphological analyzer gives us the probable tags of the words. We observe that if a word does not exist in the corpus, the morphological analyzer can help us in guessing its tag.

2. The Proposed Method

The overall architecture of the system including the connections between the modules is shown in Fig. 1. The process follows three main steps. In the first step (right part of the figure), the statistics analyzer module compiles some statistical data from the training corpus. In the second step (left part of the figure), the tag set finder module extracts possible parts of speech for words in the test corpus using the morphological analyzer. Following this, the main module of the system, the tagger module, determines the parts of speech of the words. The tagger combines the word frequencies, n-gram probabilities, heuristics data, and data about candidate tags in order to arrive at the final decision. Finally, in the third

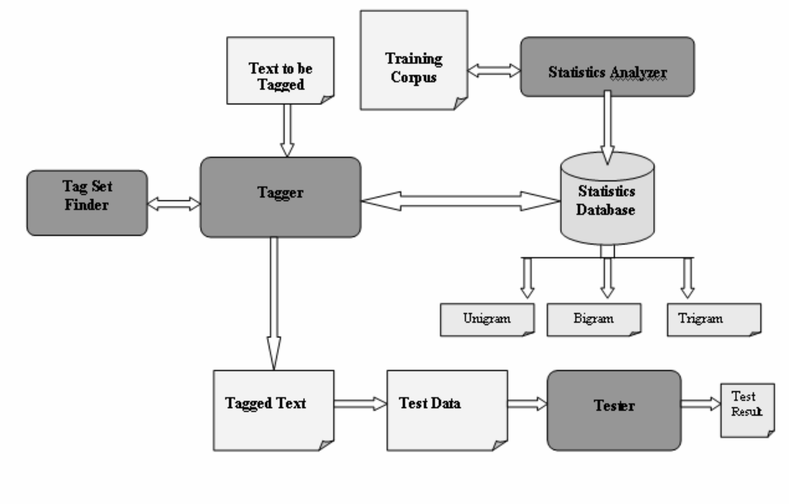

***Fig. 1.** Architecture of the system*

step (bottom part of the figure), the tester module evaluates the performance of the system by comparing the pre-tagged text with the text tagged by the system.

The corpus used in this research is METU-Sabancı Turkish Treebank, which is a treebank including the morphological analysis of the words and the syntactic parse of the sentences [10]. Since in this research we deal with part of speech tagging, we used only morphological analyses in the corpus and ignored the syntactic parses. The corpus contains about 7,200 sentences and the tag set is formed of 13 parts of speech. We divided the corpus into two parts: the training set contains 6,000 sentences (about 85 %) and the test set contains the rest (about 15 %). We use the surface forms of the words instead of the root forms. This is due to the rich derivational morphology of Turkish: a root word may easily be affixed with several derivational (and inflectional) suffixes and change its part of speech. This is a quite common situation in Turkish and hence the tagger, given a word in surface form, should determine the part of speech of the surface form of the word.

2.1. Obtaining Statistical Data from Corpus

The statistical analyzer extracts n-gram (unigram, bigram, trigram) probabilities and some heuristics data from the training corpus. In calculating the n-gram probabilities, the number of times each word (unigram), two words sequence (bigram), and three words sequence (trigram) occur in the corpus is determined, for each possible sequences of parts of speech of these words. Then the n-gram probabilities, $P_n(.)$, are calculated using the following equations for unigram, bigram, and trigram, respectively:

(1) $$P_n(t_i \mid w_i) = \frac{C(w_i / t_i)}{C(w_i)}$$

(2) $$P_n(t_{i-1}t_i \mid w_{i-1}w_i) = \frac{C(w_{i-1} / t_{i-1}, w_i / t_i)}{C(w_{i-1}w_i)}$$

(3) $$P_n(t_{i-2}t_{i-1}t_i \mid w_{i-2}w_{i-1}w_i) = \frac{C(w_{i-2} / t_{i-2}, w_{i-1} / t_{i-1}, w_i / t_i)}{C(w_{i-2}w_{i-1}w_i)}$$

where w_i,w_{i-1},w_{i-2} denote, respectively, the *i*'th word, *i-1*'th word, and *i-2*'th word; t_i,t_{i-1},t_{i-2} denote the tags of, respectively, w_i,w_{i-1},w_{i-2}; w_j/t_j indicates that the word w_j has tag t_j; and C(.) denotes the frequency in the training corpus. As a result, given a word (unigram) or a word sequence (bigram and trigram), the probability that it occurs with a particular tag or tag sequence among all possible tags or tag sequences is determined.

In addition to the n-gram approach, we propose another statistical figure which is related to parts of speech of words depending on the positions in the sentence. We take into account the initial and final words in the sentences. Although Turkish is mainly characterized as a SOV (subject-object-verb) language, it is also regarded as a free word order language which means that these three grammatical categories can appear in any position within a sentence. However, in regular sentences, subject and verb usually appear in the initial and final parts of the sentences, and the words that form these grammatical categories are limited in terms of parts of speech. For instance, it is unlikely that the first word of a subject is a conjunction. Based on this observation, we devised a heuristics and tested its plausibility for finding the correct tag of ambiguous words. For this purpose, the statistical analyzer counts the number of times each of the parts of speech occurs in the initial position and occurs in the final position of the sentences in the training corpus. Then these frequencies are converted into probabilities and stored in a 13X2 (number of parts of speech X initial/final position) table. We represent with $P_f(t)$ and $P_l(t)$ the probabilities that the first words and the last words, respectively, in the sentences are of tag t.

2.2. Tagging Using Statistical Data and Morphological Data

After the training phase where relevant statistical data have been collected from the training corpus, the tagger is activated on the test corpus. The tagger employs a sentence based approach rather a word based approach. That is, first all the possible tags for the words and the word sequences in the sentence are determined, then the combination of the tags with the highest probability for the whole sentence is selected.

Given a sentence, the first process is extracting all the possible parts of speech for each individual word. For this purpose, a Turkish morphological analyzer (PC-Kimmo) is used by the tag set finder module [8]. PC-Kimmo is based on two-level morphology formalism and outputs all the possible morphological parses for a given word. The tag set used by PC-Kimmo is slightly different than that of the Turkish Treebank, and we make a conversion from the first one to the second one. For each word in the sentence, the tag set finder module sends the word to PC-Kimmo as input, gets the possible parts of speech for the word, and calculates a probability for each possible part of speech of the word depending on the number of analyses yielding that part of speech. We use the notation $P_m(t_i|w_i)$ to denote the probability that the word w_i is assigned tag t_i by the morphological analyzer. For instance, if a word w has three possible parses where two of them result in a noun and one results in an adjective, then the probability that the word is a noun is $P_m(\text{noun}|w)=0.67$ and the probability that it is an adjective is $P_m(\text{adj}|w)=0.33$. It should be noted that this part of the process is independent of the corpus data. That is, all possible parts of speech for a word is obtained, regardless of whether the word appears within the corpus with these parts of speech or not.

The tagger then combines the statistical data collected by the statistical analyzer module (as explained in Section 2.1) and the morphological data collected by the tag set finder module. This is done in three steps as shown below. If the currently analyzed word of the sentence is w_i, then for each possible tag ti of wi, the probability that ti is the correct tag of w_i is calculated with the following equations:

$$(4)\quad P(t_i \mid w_i) = \begin{cases} 0.5 * P_m(t_i \mid w_i) + 0.5 * P_n(t_i \mid w_i) & \text{if } w_i \text{ is in corpus} \\ 0.5 * P_m(t_i \mid w_i) + 0.5 * P_f(t_i) & \text{if } w_i \text{ is not in corpus and it is first word} \\ 0.5 * P_m(t_i \mid w_i) + 0.5 * P_l(t_i) & \text{if } w_i \text{ is not in corpus and it is last word} \\ P_m(t_i \mid w_i) & \text{if } w_i \text{ is not in corpus and it is not first/last word} \end{cases}$$

$$(5)\quad P(t_i \mid w_i) = \begin{cases} 0.5 * P(t_i \mid w_i) + 0.5 * P_n(t_{i-1}t_i \mid w_{i-1}w_i) & \text{if } w_{i-1}w_i \text{ is in corpus} \\ P(t_i \mid w_i) & \text{otherwise} \end{cases}$$

$$(6)\quad P(t_i \mid w_i) = \begin{cases} 0.5 * P(t_i \mid w_i) + 0.5 * P_n(t_{i-2}t_{i-1}t_i \mid w_{i-2}w_{i-1}w_i) & \text{if } w_{i-2}w_{i-1}w_i \text{ is in corpus} \\ P(t_i \mid w_i) & \text{otherwise} \end{cases}$$

In the first step (Eqn. (4)), if the word w_i exists in the corpus, then the unigram probability obtained from the corpus and the probability obtained from the morphological analyzer are added after weighing with a factor of 0.5. If the word does not exist in the corpus, then it is looked for whether it is the first or last word in the sentence. In the first case, the morphological analyzer probability is combined with the probability that the first words of sentences are of tag t_i, while in the second case, it is combined with the probability that the last words of sentences are of tag t_i. In the case that all of these three conditions fail, only the morphological analyzer probability is taken into account. In the second step (Eqn. (5)), there are two possibilities: If w_i and the preceding word w_{i-1} exist in the corpus as a bigram, then the probability found in the previous step is combined with the bigram probability; otherwise, the probability found in the previous step does not change. The last step (Eqn. (6)) is similar to the second step, where we combine the previous step probability with the trigram probability, provided that the trigram exists in the corpus. The multiplicative factor 0.5 was determined empirically.

When we look at the above formulation, we observe that if the analyzed word w_i does not exist in the corpus at all, then we use the result of the morphological analyzer (and we also make use of the first word and last word heuristics when applicable). This approach handles the sparse data problem successfully. Sparse data is a serious problem for all statistical part of speech taggers, even in the existence of very large corpora. It is a more serious problem in our case since we use a small sized corpus. We also observe that if unigram exists but bigram and trigram do not exist, then the unigram probability is used as the final probability. In the case that bigram and/or trigram exist, they are also taken into account in calculating the final probability.

After the tag probabilities are determined, the possible paths for the sentence are computed. Each word in the sentence may have several possible tags and all possible combinations of these individual tags yield a number of alternative paths for the sentence. Thus the tagger calculates a score for each of the paths from the tag probabilities and the path (tag sequence) with the highest score is selected.

3. Experiments and Results

In order to test the accuracy of the proposed method, we have performed three experiments by using different parts of the corpus as training set and test set in each. In addition to calculating the performance of the proposed method, we have also calculated the performance when only the morphological analyzer is used (without any statistical data from the corpus). We consider this as a baseline performance. The results of the experiments are shown in Table 1.

Table 1. Performance of the method

	Exp. 1	Exp. 2	Exp. 3
Baseline	69.3 %	62.8 %	68.1 %
Proposed method	84.7 %	79.9 %	82.2 %

We see that using statistical data greatly improves the performance when compared with the baseline. We also observe that the success rates are not as high as expected. There are two reasons for this. The first reason stems from the fact that, being an agglutinative language, Turkish has a very complex derivational and inflectional morphology. There are about 200 suffixes that can be attached to words and it is possible to derive several millions of words from a single root word. A word may change its part of speech freely by affixing different suffixes. These issues impose some difficulties for tagging of agglutinative languages in general and of Turkish in particular.

The second and more important reason originates from the corpus size. We used a corpus with about 7,200 sentences, which is a very small size for our task. The difficulty in this task and in other statistical natural language processing tasks is acquiring a large enough and manually tagged corpus. Such corpora exist for widely used languages like English, but they are not available for Turkish. During the experiments, we have observed that most of the trigrams and even the bigrams in the test set did not exist in the training set. It is quite likely that the performance ratio of the proposed method will increase significantly in the case of a corpus with a size of a few million words.

4. Conclusion

In this paper, we proposed a composite part of speech tagger for Turkish. Part of speech taggers mostly follow one of two main paradigms: rule-based tagging or statistical tagging. In this work we have combined these two approaches and also made use of two additional features. One feature is incorporating a morphological analyzer, which is used to obtain the parts of speech of words independent of the corpus. This feature eliminates the sparse data problem and gives a reasonable result when the corpus does not include the analyzed word. The second feature is making use of the word order property of the language. For this purpose, we have exploited the observation that some parts of speech do not appear in the initial and final positions of sentences. The experiments have shown an increase in the accuracy of the tagger when this novel heuristics was used.

As a future work, the heuristics about the word order can be improved by taking other parts (in addition to the initial and final words) of the sentences into account. Such heuristics may prove useful especially for fixed order languages like English, since the positions of the grammatical categories in sentences in these languages do not change and thus constraining the possible parts of speech that can appear in those positions.

Acknowledgement

This work was supported by Boğaziçi University Research Fund, Grant no. 06A102.

References

[1] Brill, E., “A simple rule-based part of speech tagger”, Proceedings of 3rd Conference on ANLP, pp. 152-155, Trento, 1992.

[2] Charniak, E., Hendrickson, C., Jacobson, N., Perkowitz, M., “Equations for part-of-speech tagging”, Proceedings of Conference on Artificial Intelligence, pp. 784-789, 1993.

[3] Church, K.W., “A stochastic parts program and noun phrase parser for unrestricted text”, Proceedings of 2nd Conference on ANLP, pp. 136-143, Austin, Texas, 1988.

[4] Collins, M., “Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms”, Proceedings of EMNLP, 2002.

[5] Cutting, D., Kupiec, J., Pedersen, J., Sibun, P., “A practical part-of-speech tagger”, Proceedings of 3rd Conference on ANLP, pp. 133-140, Trento, 1992.

[6] Karlsson, F., Voutilainen, A., Heikkila, J., Anttilla, A., *Constraint grammar: a language independent system for parsing unrestricted text*, Mouton de Gruyter, 1995.

[7] Mikheev, A., “Learning part-of-speech guessing rules from lexicon: extension to non-concatenative operations”, Proceedings of COLING, pp. 770-775, 1996.

[8] Oflazer, K., “Two-level description of Turkish morphology”, Literary and Linguistic Computing, Vol. 9, No. 2, pp. 137-148, 1993.

[9] Oflazer, K., Kuruöz, İ., “Tagging and morphological disambiguation of Turkish text”, Proceedings of 4th Conference on ANLP, pp. 144-149, 1994.

[10] Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G., *Building a Turkish treebank*, Editor: Abeille, A., in: *Building and exploiting syntactically annotated corpora*, Kluwer, 2003.

[11] Pla, F., Molina, A., “Part-of-speech tagging with lexicalized HMM”, Proceedings of Recent Advances in Natural Language Processing, Bulgaria, 2001.

[12] Rathnaparki, A., “A maximum-entropy model for part-of-speech tagging”, Proceedings of Conference on EMNLP, pp. 133-142, Philadelphia, 1996.

[13] Schütze, H., “Distributional part-of-speech tagging”, Proceedings of 7th EACL Conference, pp. 141-148, Dublin, 1995.

[14] Schütze, H., Singer, Y., “Part-of-speech tagging using a variable memory Markov model”, Proceedings of ACL, New Mexico, 1994.

Levent Altunyurt
Department of Computer Engineering, Boğaziçi
University, Bebek 34342, İstanbul, Turkey
e-mail: levent.altunyurt@gmail.com
Zihni Orhan
Department of Computer Engineering, Boğaziçi
University, Bebek 34342, İstanbul, Turkey
zihni@boun.edu.tr
Tunga Güngör
Department of Computer Engineering, Boğaziçi
University, Bebek 34342, İstanbul, Turkey
gungort@boun.edu.tr