Comparing different POS-tagging techniques for Hungarian

András Kuba[1], Tibor Bakota[1], András Hócza[1], Csaba Oravecz[2]

[1]University of Szeged, Department of Informatics
Research Group on Artificial Intelligence
andkuba@inf.u-szeged.hu, bakotat@math.u-szeged.hu, hocza@inf.u-szeged.hu
[2]Research Institute for Linguistics at the Hungarian Academy of Sciences
oravecz@nytud.hu

Keywords: POS tagging, rule-based methods, Hidden Markov Model

Abtsract

In recent years, many different techniques have been developed for tagging natural languages, but only a few of them were implemented for Hungarian. The most commonly used types are statistics based taggers, but also the rule based ones are very widespread. We currently took up the challenge to compare four taggers. The first one, *VMM*, has been developed at the University of Szeged, Department of Informatics, and it is based on Hidden Markov Model method. The second one, also developed at the Department of Informatics is a rule based tagger called *RGLearn*. These two taggers were compared with the well-known TnT software, and the C4.5 algorithm. TnT is also a statistics based tagger, but it operates with trigrams, while *VMM* is just a bigram tagger. The comparison was performed on the so-called Szeged Corpus, which is a manually annotated set of text containing approx. 1.2 million words from different topic areas. Because very fine MSD encoding was used (there are approx. 1 500 different tags in the MSD encoding system), the taggers aren't expected to perform well. In our case POS tagging is the problem where the ambiguity class of each word is known, and the tagger decides which tag sequence represents the correct meaning of the sentence. The reason of this assumption is that these taggers should be part of a *ToolChain*, where in an earlier phase the *HuMor* morphological tagging software generates the possible tags of each word. In this sense *RGLearn* was found to be the best, performing 96,16% per word accuracy, *VMM* performed 95,98%, TnT 95,08%, and finally C4.5 94,94%. Here we have to note that information about ambiguity classes of words were not available for TnT. During the training TnT generates a suffix tree, which is used for a naive morphological examination of each word in order to determine its possible tags. This heuristics rapidly increases the accuracy on unknown words. After these tests were finished, a list of mistaken words was extracted for each tagger, and than compared. We've found that *VMM* and *TnT* had more mistakes in common than any other two taggers. This is due to the fact that both taggers use statistical methods. Using these results, we've come to conclusions about how these taggers could be combined in order to produce better results. The list of mistakes was forwarded to linguists for analysis, and to find out weather the machine or human made the mistake. The results also are used to make corrections to the corpus. Till now some 8000 mistakes were noticed and corrected by using this method, which is about 0.7% of the whole corpus.