Combining Statistical Models for POS Tagging using Finite-State Calculus

Miikka Silfverberg
Helsinki University
Helsinki, Finland
miikka.silfverberg@helsinki.fi

Krister Lindén
Helsinki University
Helsinki, Finland
krister.linden@helsinki.fi

Abstract

We introduce a framework for POS tagging which can incorporate a variety of different information sources such as statistical models and hand-written rules. The information sources are compiled into a set of weighted finite-state transducers and tagging is accomplished using weighted finite-state algorithms. Our aim is to develop a fast and flexible way for trying out different tagger designs and combining them into hybrid systems. We test the applicability of the framework by constructing HMM taggers with augmented lexical models for English and Finnish. We compare our taggers with two existing statistical taggers TnT and Hunpos and find that we achieve superior accuracy.

1 Introduction

Part-of-Speech (POS) tagging, and other sequential labeling tasks like named entity recognition and chunking, constitute core tasks of language technology. Highly successful POS taggers for English have been constructed both using rule-based methods e.g. finite-state constraints used by Voutilainen (1995) and statistical methods e.g. Hidden Markov Models (HMM) used by Brants (2000).

Besides HMMs, other statistical models such as Conditional Random Fields and Maximum Entropy Models have recently been used to construct POS taggers, but HMMs remain one of the most widely used in practice. Though the more recent models surpass HMMs in accuracy, the great tagging speed and a fast development cycle of HMMs ensure a continuing popularity.

Accuracies for state of the art statistical taggers for English newspaper text surpass 97%, but results for applying these models on other languages are not always as encouraging. E.g. Dredze and Wallenberg (2008) report an accuracy 92.06% on tagging Icelandic using bidirectional sequence classification.

Low accuracy is partly due to the lack of sufficiently large tagged corpora, which can be used as training material. Reduction of accuracy can also result from the fact that the syntax and morphology of many languages differ substantially from English syntax and morphology. E.g. many languages do not have as rigid word order as English and many languages incorporate far more extensive morphological phenomena. Thus models, which have been developed for English, may not work well on many other languages.

These practical and theoretical problems associated with constructing POS taggers for virtually all of the world's languages, demonstrate the need for POS tagging models, which can incorporate a variety of different information sources including different kinds of statistical models but also more linguistic models like the ones utilized by Voutilainen (1995). Ideally the linguistic models could be used to fine-tune the result of the statistical tagging.

We propose a general framework for building POS taggers, where various kinds of statistical models and other POS tagging models can be combined using weighted finite-state calculus. Using this framework, developers can test a variety of models for tagging a language and apply the models in parallel. E.g. a statistical model trained with insufficient training data can be augmented with hand-made or machine-learned rules for common tagging errors.

In order to test the framework, we trained standard second order HMMs for Finnish and English and augmented these with extended lexical models using tag context.

We train and evaluate the English tagger using the Wall Street Journal (WSJ) corpus from Penn

Bolette Sandford Pedersen, Gunta Nešpore and Inguna Skadiņa (Eds.)
NODALIDA 2011 Conference Proceedings, pp. 183–190

Treebank II (Marcus et al., 1994). We compare the accuracy obtained by our model with the well known and widely used HMM tagger TnT (Brants, 2000) and a more recent open-source HMM tagger Hunpos (Halácsy et al., 2007), which also utilizes an extended lexical model. After improving upon the lexical model of Hunpos, our tagger obtains an accuracy of 96.67%, outperforming both TnT (96.46%) and Hunpos (96.58%).

For training and testing the Finnish tagger, we use morphologically analyzed and disambiguated newspaper text. The optimization of the model for Finnish requires some changes in the model. Together these changes improve the accuracy from a baseline second order HMM by more than 1%. We also train a Hunpos tagger for Finnish and compare it with our own tagger. The 96.02% accuracy, we obtain on the Finnish material, clearly outperforms Hunpos (95.62%).

We implemented all taggers using the freely available HFST-interface for weighted finite-state transducers (Lindén et al., 2009). An open-source interface for constructing taggers in our framework will be made publicly available.

This paper is structured in the following way. We first review some earlier work on enhancing the accuracy of HMMs. We then introduce our framework for constructing taggers in section 4. In section 5 we introduce an HMM tagger augmented with contextual lexical probabilities, which we implemented for English and Finnish in our framework. We then evaluate the English and Finnish taggers using corpus data and compare them with TnT and Hunpos. Following evaluation, we present a brief discussion on our results and future work. Finally we conclude the paper.

2 Previous Work

Statistical POS tagging is a common task in natural language applications. POS taggers can be implemented using a variety of statistical models including Hidden Markov Models (HMM) (Church, 1999; Brants, 2000), Maximum Entropy Models (Tsuruoka et al., 2005) and Conditional Random Fields (Lafferty et al., 2001).

HMMs are probably the most widely used technique for POS tagging and one of the best known imlpementations of an HMM is TnT by Brants (2000). When tagging the WSJ corpus using the splits introduced by Collins (2000), TnT achieves an accuracy of 96.46%. Although more recent statistical techniques result in improved accuracy, HMMs have remained in use chiefly because of the speed of both developing a tagger and tagging.

Recently Banko and Moore (2004) and Halácsy et al. (2007) have worked on improving the accuracy of HMMs by adding tag context into the lexical model of the HMM. The technique was pioneered by Toutanova et al. (2003) in the context of Conditional Markov Models.

The strength of Banko and Moore (2004) is that their lexical models use both left and right context when determining the conditional probability which should be associated to a wordform given a tag. The Hunpos tagger by Halácsy et al. (2007) uses only the left tag context, but it does not require a full lexicon, which makes it very practical.

We combine the left and right tag context in lexical models with a guesser for unknown wordforms. Our approach differs from Hunpos in that we only use contextually dependent lexical probabilities for known words.

Besides evaluating our approach to POS tagging by constructing a tagger for English text, we also test our approach on Finnish. Work with statistical POS tagging for Finnish seems to be virtually non-existent. Silfverberg and Lindén (2010) derive a Finnish POS tagger for the Finnish Europarl corpus (Koehn, 2005), which achieves high accuracy i.e. 96.63%, but these results could be contested on the grounds that the Europarl corpus is translated into Finnish from other languages. Silfverberg and Lindén (2010) also use an extremely large (25 million tokens) corpus. We use Finnish newspaper text to train and evaluate the tagger. Our training corpus is comparable in size to the Wall Street Journal corpus.

3 Note on Terminology

We use the terms *analysis*, *POS tag* and *tag* interchangeably to refer to POS tags, which are given for words. The *correct tag* or *analysis* refers to the intended analysis of a word in a gold standard corpus. By the term an *analysis of a sentence*, we signify one possible way to assign a unique POS tag to each of the words in the sentence. We use the term *correct analysis of a sentence* to denote the unique analysis where all of the words receive their correct analyses.

The term *analysis or tag profile of a word* refers to the set of tags which can occur as its POS analyses.

If all of the analyses of a sentence are compiled into a transducer, the paths of the transducer correspond exactly to the analyses of the sentence. In this setting, we use the terms analysis and path interchangeably. We call the transducer, compiled from the tag profiles and associated probabilities, the *sentence transducer*.

4 A Framework for Constructing POS Taggers

Our framework factors POS tagging into two tasks: (i) assigning tag profiles and probabilities $p(w|t)$ to each word w in a sentence and each of its possible analyses t and (ii) re-scoring the different analyses of the entire sentence using parallel weighted models for word and tag sequences.[1]

In the first task, the tag profile for a word w and the probabilities $p(w|t)$ for each of its tags is estimated from a training corpus. The probabilities are independent of surrounding words and tags. For unknown words u, a number of guessers can be included. These estimate the probabilities $p(u|t)$ using the probabilities $p(s|t)$ for the suffixes of u. The suffix probabilities can be estimated from a training corpus.

A number of guessers can be used to estimate the distribution of analyses for different kinds of unknown words. Like Hunpos and TnT, we always include different guessers for upper case words and lower case words, which improves accuracy.

The tag profiles of words along with tag probabilities are compiled into a weighted finite-state transducer, which associates a probability for every possible analysis of the sentence. The probability assigned to a path at this stage is the product of lexical probabilities.

After assigning tag profiles and probabilities for words, the second task is to re-score the paths of the sentence transducer. Different models can be used to accomplish this. Each of the models adds some weight to each of the analyses of the sentence and their combined effect determines the best path i.e. the most probable path. We could also incorporate models which forbid some analyses. This means that the analyses are discarded in favor of other analyses which initially seemed less likely. Such models could be used to correct systematic errors stemming from the statistical models.

The result of applying the re-scoring models to the tag profiles is computed using weighted intersecting composition by Silfverberg and Lindén (2009). After re-scoring, a best paths algorithm (Mohri and Riley, 2002) is used to extract the most probable analysis for the sentence.

5 Augmented HMM POS Tagger for English and Finnish

For English and Finnish we constructed POS taggers based on traditional second order HMMs augmented with models, which re-score lexical probabilitites according to tag context (this is the factor $p(w_i|t_{i-1}, t_i, t_{i+1})$ in the formula below). For the sentence w_i, ..., w_n, the taggers attempt to maximize the probability $p(t_1, ..., t_n|w_i, ..., w_n)$ over tag sequences t_1, ..., t_n. Because of the data sparseness problem, it is impossible to compute the probability directly, so the tagger instead maximizes its approximation

$$\prod_{i=1}^{n} p(t_i|t_{i-1}, t_{i-2})p(w_i|t_{i-1}, t_i, t_{i+1})p(w_i|t_i)$$

where the tag sequences t_1 ..., t_n ranges over all analyses of the sentence. The term $p(t_i|t_{i-1}, t_{i-2})$ is the standard second order HMM approximation for the probability of the tag t_i. The term $p(w_i|t_{i-1}, t_i, t_{i+1})$ conditions the probability of the word w_i on its tag context. Finally the term $p(w_i|t_i)$ is the standard HMM lexical probability.

In order to get the indices to match in the formula above, three additional symbols are needed, i.e. t_{-1}, t_0 and t_{n+1} denote sentence boundary symbols, which are added during training and tagging for improved accuracy. Using sentence boundary symbols is adopted from Brants (2000).

In order to get some estimates for the probability of tag trigrams, which did not occur in the training data, we use tag bigram $p(t_i|t_{i-1})$ and tag unigram $p(t_i)$ models in parallel to the trigram model. Similarly we use models which assign probability $p(w_i|t_{i-1}, t_i)$ and $p(w_i|t_i, t_{i+1})$ in order to deal with previous unseen tag trigrams and wordforms. Of course the lexical model also weights analyses of words, serving as a backup model even in the case where the tag bigrams with the wordform were previously unseen.

[1] Although the probabilities $p(t|w)$ would seem like a more natural choice in the lexical model, the approximation for probabilities used in the HMM model of the tagger require the inverted probabilities $p(w|t)$. For a more thorough discussion of HMMs see Manning and Schütze (1999).

5.1 Lexical Models

For each tag t and word w, our lexical model estimates the probability $p(w|t)$. For unknown words, we construct similar guessers as Brants (2000) and Halácsy et al. (2007). The guessers estimate the probability $p(w|t)$ using the probabilities $p(s_i|t)$ for each of the suffixes of w. These can be computed from training material. The estimate $p(w|t)$ is a smoothed sum of the estimates for all of the suffixes, as explained by Brants (2000).

Like Brants (2000), we train separate guessers for upper and lower case words. For Finnish, we additionally train a guesser for sentence initial words, because preliminary tests revealed that there were a lot of unknown sentence initial words. Using a separate guesser for these words yielded better results than using the upper case or the lower case guesser. For English, a separate guesser for sentence initial words does not improve accuracy.

For Finnish another modification was needed in addition to the added guesser. For unknown words, it seemed beneficial to use only the 10 highest ranking guesses. For English, reducing the number of guesses also reduces accuracy. The maximum number of guesses is therefore a parameter which needs to be estimated experimentally and can vary between languages.

5.2 Tag Sequence Models

We construct a set of finite-state transducers whose effect is equivalent to an HMM. For the sake of space reduction, we do not compile a single transducer equivalent to an HMM. Instead we split the HMM into component models, each of which weights n-grams of wordforms and tags in the sentence. We give a short overview here and refer to Silfverberg and Lindén (2010) for a more thorough discussion on how this is done.

We simulate the tag n-gram models of a second order HMM using six models compiled into transducers. We use one transducer which assigns probabilitites for the tag unigrams in the sentence, two transducers which assign probabilitites for tag bigrams and three transducers assigning probabilities for tag trigrams.

As an example of how the transducers operate, we explain the structure of the three transducers which assign probabilities to tag trigrams. As explained above: After lexical probabilities have been assigned to the words in a sentence, the words and their analyses are compiled into a finite-state transducer, which assigns probabilities to the possible analyses of the entire sentence. Each of the three component models of the trigram model re-weight the paths of this transducer.

The first one of the models starts with the first three words (1st, 2nd and 3rd word) of the sentence and assigns a probability for each analysis trigram of the word triplet. It then moves on to the next three words (4th, 5th and 6th word) and their analyses, and so on. Hence the first model assigns a probability for each triplet of words and its analyses, which begins at indices $3k + 1$ in the sentence.

The second model skips the first word of the sentence, but after that it behaves as the first model re-scoring first the analyses of the triple (2nd, 3rd and 4th) and going on. As a result, it assigns probabilities to trigrams starting at indices $3k + 2$ in the sentence. By skipping the first two words, the third trigram model assigns weight to triplets beginning at indices $3k$. The net effect is that each trigram of wordforms and tags gets weighted once by the trigram model.

The models re-weighting tag bigrams and tag unigrams are constructed in an analogous way to the tag trigram models and the unigram bigram and trigram probabilities are smoothed using deleted interpolation, as suggested by Brants (2000).

Each of the models assigns a minimum penalty probability $1/(N + 1)$ to unknown tag n-grams. Here N is the size of the training corpus.

5.3 Context Dependent Lexical Models

In addition to the transducers making up the HMM model, we construct context dependent lexical models, which assign probabilities

$$p(w_i|t_{i-1}, t_i, t_{i+1}),\ p(w_i|t_{i-1}, t_i),\ p(w_i|t_i, t_{i+1})$$

to word and tag combinations in analyses. The models which assign probabilities to word and tag bigram combinations are included in order to estimate the probability $p(w_i|t_{i-1}, t_i, t_{i+1})$ when the combination of w_i with tags t_{i-1}, t_i and t_{i+1} has not been seen during training.

The context dependent lexical models are only applied to known words, but they do also provide additional improvement for tagging accuracy of unknown words by directly using neighboring words in estimating their tag profiles and proba-

bilities. This is more reliable than using tag sequences.

The choice to only apply the models on known words is a convenient one. For known words context dependent lexical models were very easy for us to compile, since they are quite similar to ordinary tag n-gram models. Integrating them with the transducers making up the HMM model did not require any extra work besides estimating experimentally three coefficients which weight the models w.r.t. the HMM and each other. Weighted intersecting composition can be used to combine the sentence transducer and the re-scoring models regardless of how many models there are.

Similarly as in the HMM, unknown combinations of tags and words receive probability $1/(N+1)$, where N is the training corpus size.

6 Data

We trained taggers for English and Finnish using corpora compiled from newspaper text.

For English we used the Wall Street Journal Corpus in the Penn Treebank. We adopted the practice, introduced by Collins (2000), to use sections 0-18 for training lexical and tag models, sections 19-21 for fine tuning (like computing deleted interpolation coefficients) and sections 22-24 for testing.

For Finnish, we used a morphologically analyzed and disambiguated corpus of news from the 1995 volume of Helsingin Sanomat, the leading Finnish newspaper[2] (We used the news from the KA section of the corpus).

The morphological tagging in the Finnish corpus is machine-made and it has not been checked manually. This soon becomes evident when one examines the corpus, since there are a number of tagging errors. Thus our results for Finnish have to be considered tentative.

Table 1 shows the number of tokens in the training, fine-tuning and test materials used to construct and evaluate the taggers. The tokenization of the corpora is used as is and all token counts include words and punctuation. As the table shows, token counts for the Finnish and English corpora are comparable.

[2]Information about the corpus is available from `http://www.csc.fi/english/research/software/ftc`. It was compiled by The Research Institute for the Languages of Finland and CSC - IT Center for Science Ltd. The corpus can be obtained for academic use.

	English	Finnish
Training	969905	1027514
Tuning	148158	181437
Testing	171138 (2.43%)	156572 (10.41%)

Table 1: Summary of token counts for the data used for evaluation. The counts include words and punctuation. The amount of words, which were not seen during training, is indicated in parentheses.

	English	Finnish
POS Tags	81	776

Table 2: Number of POS tags in the Finnish and English corpora.

The amount of unknown words in the test corpus for Finnish is high. This is to be expected given the extensive morphology of the language. The extensive morphology is also reflected in the tag counts in table 2, which shows that the tag profile of the Finnish corpus is nearly ten times as large as the tag profile of the WSJ.[3]

Of the tags, in the Finnish corpus, 471 occur ten times or more, 243 occur one hundred times or more and 86 occur one thousand times or more. We conclude that there is a large number of tags which are fairly frequent. The corresponding figures for English are 58 tags occurring ten times or more, 44 tags occurring one hundred times or more and 38 tags occurring one thousand times or more.

The average number of possible analyses for words in the English corpus is 2.34. In the Finnish corpus, a word receives on average 1.45 analyses. The high number of analyses in the English corpus is partly explained by certain infrequent analyses of the frequent words ”a” and ”the”. When these words are excluded, the average number of analyses drops to 2.06.

When reporting accuracy, we divide the number of correctly tagged tokens with the total number of tokens in the test material, i.e. accuracy counts include punctuation. In this we follow the ACLWiki State of the Art page for POS tagging[4]. All re-

[3]There are 45 unique POS markers (such as `NN` and `JJ`) used in WSJ, but there are some unresolved ambiguities left in the corpus. That is why some words have POS tags consisting of more than one marker (eg. `VBG|NN|JJ`) making the total number of POS tags 81.

[4]`http://www.aclweb.org/aclwiki/`

sults on accuracy are reported for the test materials, which were not seen during training.

7 Evaluation

We trained four separate taggers both for English and Finnish. The accuracies for the different models are shown in table 3.

	1	2	3	4
Eng	96.42%	96.55%	96.70%	96.77%
Fin	95.56%	95.87%	95.98%	96.02%

1. Second order HMM.
2. Second order HMM augmented with lexical probabilities $p(w_i|t_{i-1}, t_i)$.
3. Second order HMM augmented with lexical probabilities $p(w_i|t_{i-1}, t_i)$ and $p(w_i|t_i, t_{i+1})$.
4. Second order HMM augmented with lexical probabilities $p(w_i|t_{i-1}, t_i)$, $p(w_i|t_i, t_{i+1})$ and $p(w_i|t_{i-1}, t_i, t_{i+1})$.

Table 3: Summary of tagging accuracies using different models. For Finnish, a separate guesser is used for sentence initial words.

The first tagger is a standard second order HMM. The only divergence from the HMM introduced by Brants (2000) is training a separate guesser for sentence initial words for Finnish and limiting the number of guesses to 10 for Finnish. This considerably improves the accuracy of the tagger from 94.91% to 95.56%.

The second tagger, we evaluate, is an HMM augmented by lexical probabilities conditioned on left tag context $p(w_i|t_{i-1}, t_i)$. This model roughly corresponds to the model Hunpos uses for POS tagging. As pointed out in section 5.3, the difference is that we do not estimate context dependent lexical probabilities for unknown words. This seems to lead to a slight reduction in accuracy.

In the third tagger, we add lexical probabilitites conditioned on right context $p(w_i|t_i, t_{i+1})$ and in the fourth tagger we add the final statistical model, which additionally uses lexical probabilities conditioned on both right and left tag context $p(w_i|t_{i-1}, t_i, t_{i+1})$.

The second tagger performs nearly as well as Hunpos and the third and fourth taggers perform better. This is to be expected, since the taggers incorporate right lexical context, which Hunpos cannot utilize.

index.php?title=POS_Tagging_(State_of_the_art)

	Seen	Unseen	Overall
TnT	96.77%	85.19%	96.46%
Hunpos	96.88%	86.13%	96.58%
Hfst	97.13%	83.72%	96.77%

Table 4: Summary of tagging accuracies for WSJ using TnT, Hunpos and Hfst. The accuracies are given for seen, unseen and all tokens. Hfst is our own tagger.

Table 4 shows accuracies for TnT, Hunpos and the best of our models, which we call **Hfst**, when tagging WSJ. It clearly performs the best out of all the taggers on all tokens and it has very high accuracy on known tokens. For unknown words, its accuracy is nevertheless somewhat lower than for TnT and Hunpos.

	Seen	Unseen	Overall
Hunpos	98.06%	76.83%	95.62%
Hfst	97.98%	81.04%	96.02%

Table 5: Summary of tagging accuracies for the Finnish test corpus using Hunpos and Hfst. The accuracies are given for seen, unseen and all tokens. Hfst is our own tagger.

Table 5 shows accuracies for Hunpos and Hfst on the Finnish test corpus. The accuracy on known words is markedly high for both taggers. This is probably partly due to the low average number of analyses per word, which makes analyzing known words easier than in English text. Conversely, the accuracy on unknown words is quite low and much lower for Hunpos than for Hfst.

By increasing the number of guesses from 10 to 40 for unknown words, we accomplish a similar reduction in accuracy on unknown words (from 81.04% to 79.29%) for Hfst as Hunpos exhibits. This points to the direction that the problems Hunpos encounters in tagging unknown Finnish words are in fact due to its unrestricted guesser.

In conclusion, the Hfst tagger has better overall performance than both Hunpos and TnT.

8 Discussion and Future Work

Because of extensive and fairly regular morphology, words in Finnish contain a lot of information about their part-of-speech and inflection. Hence

words which share long suffixes are probably more likely to get the same correct POS analysis in Finnish than in English.

By considering more guesses for a Finnish unknown word, one at the same time considers more guesses which were suggested on basis of words with short suffixes in common with the unknown word. This problem is worsened by smoothing, which reduces the differences between the probabilities of suggestions. In fact we believe that this implies that the kind of guesser suggested by Brants (2000) and used in Hunpos is not the ideal choice for Finnish. And an architecture which makes it possible to try out different guesser designs, could make a tagger toolkit adaptable for a larger variety of languages than a traditional HMM.

The need for an added guesser for sentence initial words in Finnish can be understood rather easily by examining the test corpus. Sentences are fairly short, on average 10.3 words. The number of unknown words in the corpus is high and sentence initial words are not an exception. Both using the guesser for lower case words and upper case words produces poor results, because the first one underestimates the number of proper names among sentence initial words and the second one grossly overestimates it. Hence a guesser trained either on all words in the test material or only sentence initial words is needed.

The need for a sentence initial guesser in fact speaks in favor of Hunpos, since it incorporates a contextually dependent lexical model also for unknown words. Therefore it needs no special tweaks in order to perform well on sentence boundaries. Still, our baseline second order HMM achieves 95.56% which is extremely close to Hunpos 95.62%. The baseline model uses context dependent lexical probabilities only in the sense that it uses the separate guesser for sentence initial words. Perhaps this is indeed the only place where a context sensitive guesser has added effect in the Finnish corpus.

There is a lot of work left with the Hfst tagger. The accuracy of the guesser needs to be improved even when tagging English. There should not be any reason why it could not be made at least as accurate as the guesser in TnT.

Another improvement would be a rule compiler which compiles hand-written rules into sequential models, that are compatible with the statistical models which are used currently. Especially for Finnish, such a rule compiler would be a significant asset, because e.g. the disambiguation of analyses of verb forms often leads to long distance dependencies, which n-gram models capture poorly. It is not evident how TnT or Hupos could be adapted to using e.g. hand-written tagging rules in order to improve performance. But it would be an easy task for Hfst, if only there existed a suitable rule compiler.

A third direction of future work, would be to try the framework for other sequential labelling tasks such as tokenizing, chunking and named entity recognition.

9 Conclusion

We have demonstrated a framework for constructing POS taggers, which is capable of incorporating a variety of knowledge sources for POS tagging. We showed that it is possible, even straight forward, to combine different statistical models into one tagger. We hope that we have also demonstrated that it would be fairly straight forward to incorporate other kinds of models as well.

We constructed taggers for English and Finnish, which obtain superior accuracy compared to two widely known and used taggers TnT and Hunpos based on HMMs. For Finnish we modified the guessers used to tag unknown words in order to achieve added accuracy. Because of the modular design of our system, this did not require changes in any of the other models. We believe that the accuracy on tagging Finnish 96.02% shows that our taggers can be successfully adapted to languages which differ substantially from English.

Acknowledgments

We thank the HFST team for their support. We would also like to thank the anonymous reviewers of this paper. Their comments were appreciated. Miikka Silfverberg was financed by LangNet the Finnish doctoral programme in language studies.

References

Michelle Banko and Robert C. Moore. 2004. *Part of Speech Tagging in Context*. Proceedings of the 20th international conference on Computational Linguistics, COLING-2004, Stroudsburg, PA, USA.

Thorsten Brants. 2000. *A Statistical Part-of-Speech Tagger*. Proceedings of the sixth conference on

Applied natural language processing, ANLP-2000, Seattle, USA.

Kenneth Church. 1988. *A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text*. Proceedings of the second conference on Applied natural language processing, ANLP-1988, Austin, Texas, USA.

Michael Collins. 2002. *Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms*. Proceedings of the ACL-02 conference on Empirical methods in natural language processing, EMNLP-2002, Philadelphia, USA.

Dóra Csendes, János Csirik and Tibor Gyimóthy. 2004. *The Szeged Corpus: A POS tagged and Syntactically Annotated Hungarian Natural Language Corpus*. Proceedings of the 7th International Conference on Text Speech and Dialogue, TSD-2004, Brno, Czech Repulic.

Mark Dredze and Joel Wallenberg. 2008. *Icelandic data driven part of speech tagging*. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, HLT-2008, Stroudsburg, PA, USA.

Péter Halácsy, András Kornai and Csaba Oravecz. 2007. *HunPos – An Open Source Trigram Tagger*. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, ACL-2007, Prague, Czech Republic.

Philipp Koehn. 2005. *Europarl: A Parallel Corpus for Statistical Machine Translation*. Machine Translation Summit, MTS-2005, Phuket, Thailand.

John Lafferty, Andrew MacCallum and Fernando Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of the Eighteenth International Conference on Machine Learning, ICML-2001, Williamtown, MA, USA.

Krister Lindén, Miikka Silfverberg and Tommi Pirinen. 2009. *Hfst Tools for Morphology – an Efficient Open-Source Package for Construction of Morphological Analyzers*. Workshop on Systems and Frameworks for Computational Morphology, SFCM-2009, Zürich, Switzerland.

Christopher Manning and Hinrich Schütze. 1999. *Foundations of Natural Language Processing*. The MIT Press, Massachusets, USA.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz and Britta Schasberger. 1994. *The Penn Treebank: Annotating Predicate Argument Structure*. ARPA Human Language Technology Workshop, ARPA-1994, Plainsboro, New Jersey, USA.

Mehryar Mohri and Michael Riley. 2002. *An Efficient Algorithm for the n-Best-Strings Problem*. 7th International Conference on Spoken Language Processing, ICSLP-2002, Denver, USA.

Miikka Silfverberg and Krister Lindén. 2010. *Part-of-Speech Tagging Using Parallel Weighted Finite-State Transducers*. 7th International Conference on Natural Language Processing, ICETAL-2010, Reykjavik, Iceland.

Miikka Silfverberg and Krister Lindén. 2009. *Conflict Resolution Using Weighted Rules in HFST-TwolC*. The 17th Nordic Conference of Computational Linguistics, NODALIDA-2009, Odense, Denmark.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, HLT-NAACL-2003, Edmonton, Canada.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou and Jun'ichi Tsuji. *Developing a Robust Part-of-Speech Tagger for Biomedical Text, Advances in Informatics*. 10th Panhellenic Conference on Informatics, PCI-2005, Volos, Greece.

Atro Voutilainen. *A Syntax-Based Part-of-Speech Analyser*. Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics, EACL-1995, Dublin, Ireland.

ISSN 1736-6305 Vol. 11
http://hdl.handle.net/10062/16955