# Discrete versus Probabilistic Sequence Classifiers for Domain-specific Entity Chunking

Sander Canisius [a] Antal van den Bosch [a] Walter Daelemans [b]

[a] *ILK / Language and Information Science, Tilburg University, The Netherlands*
[b] *CNTS, University of Antwerp, Belgium*

**Abstract**

We present a comparative case study of discrete and probabilistic sequence classification methods applied to two real-world entity chunking tasks in the medical domain. It is shown that a discrete version of maximum-entropy models that does not coordinate its decisions is outperformed by both architecturally-augmented discrete versions, and probabilistic versions combined with an inference step to select the best output label sequence. In addition, we show that among the various sequence-aware methods evaluated in this study, be they discrete or probabilistic, no significant performance difference could be observed. This suggests that probabilistic sequence labelling methods are not fundamentally more suited for the type of sequence-oriented entity chunking tasks evaluated in this study than augmented discrete approaches. Future research should point out whether this result generalises to more types of sequence tasks in natural language processing.

## 1 Introduction

Many tasks in natural language processing involve the complex mapping of sequences to other sequences. One class of processing tasks is the identification of entities in text, or entity chunking. This task involves the identification of the beginning and the end of an entity chunk, which can span multiple words, as well as assigning a particular label to the identified chunk. In domain-specific entity chunking, the label is one out of a limited list of domain labels, the automatic identification of which in unseen text is relevant to further processes, for instance, information retrieval, information extraction, or question answering.

One typical machine-learning approach to entity chunking is to rephrase the sequence-to-sequence mapping task as a decomposition into a sequence of local classification steps. In each step, a fixed-length feature vector is mapped to an isolated symbol in the output sequence. The standard representational approach to decompose sequence processes into local-classification cases, is *windowing*. Within a window, fixed-length subsequences of adjacent input symbols, representing a certain contextual scope, are mapped to one output symbol, typically associated with one of the input symbols, for example the middle one. After all local classifications have been made, a simple concatenation of the predicted output symbols yields the complete output sequence. The fact that the classifier is only trained to associate subsequences of input symbols to single output symbols is a problematic restriction: it may easily cause the classifier to produce invalid or impossible output sequences, since it is incapable of taking into account any decisions it has made earlier, or even decisions it might have to make further on in the input sequence.

Developing techniques that manage to circumvent this restriction has been a popular topic in machine learning research in recent years. Many of the techniques proposed no longer compose output sequences simply by concatenating the isolated predictions for each input symbol, but try to optimise the likelihood of the entire output sequence, rather than predicting the sequence of individually most-likely output symbols. To this end, some way to estimate the quality of

an entire output sequence is required. Methods such as maximum-entropy markov models and conditional random fields express this sequence quality in terms of probabilities, which adds the extra requirement of an underlying classifier capable of predicting valid class probabilities for each possible output symbol.

A large class of classifiers, henceforth referred to as discrete classifiers, do not model such probability distributions, but only predict a single best class label for each given instance. With a windowing approach, only predicting a single class as opposed to giving conditional probability estimates for every possible class means there is basically just one opportunity for predicting each symbol of the output sequence. In the probabilistic case, it is possible to prefer an output label sequence where one of the symbols actually is the second-best option if that means the overall quality of the entire output sequence is improved. Most discrete classifiers, however, lack the means for making such global trade-offs. Nevertheless, discrete classifiers, which include many commonly used learning algorithms such as memory-based learning and decision trees are frequently used for sequence labelling tasks, and in this context, alternatives to the probabilistic methods have been developed, based on clever feature engineering, meta-learning, or post-processing.

In this paper, an empirical study is described in which a maximum-entropy classifier is used both as a discrete classifier and a probabilistic one to perform two domain-specific entity chunking tasks. First, the classifier is taken to be a discrete classifier. In this setup, three existing discrete sequence labelling methods are applied to the sample tasks. The methods try to improve performance on sequence labelling tasks by introducing new features, using classifier stacking, or predicting small subsequences of the output sequence instead of single symbols. Next, a number of methods to exploit the probabilistic character of the maximum-entropy classifier are tested. The methods differ in the way the probability of candidate output label sequences is estimated, the search algorithm employed for finding the optimal output label sequence, or in both. The methods tested are conditional markov models (CMM), maximum-entropy markov models (MEMM), and conditional random fields (CRF).

The structure of the paper is as follows. First, we introduce the two chunking sequence segmentation tasks studied in this paper, in Section 2. The two subsequent sections report on empirical results for the different methods proposed for correcting the near-sightedness problem of a naive baseline method based on simple windowing of the input sequence and no other information. We first report on experiments with discrete classification methods, viz. a feedback-loop approach, stacking, and predicting class trigrams in Section 3. Second, we present results obtained with probabilistic methods, viz. conditional markov models, maximum-entropy markov models, and conditional random fields, in Section 4. In Section 5, we discuss the implications of the experimental findings presented in the two foregoing sections. Finally, Section 6 sums up and discusses the main results of the comparison.

## 2 Data and Methodology

The two data sets that have been used for this study are examples of sentence-level entity chunking tasks: concept extraction from general medical encyclopedic texts (henceforth MED), and labelling of DNA, RNA, protein, cellular, and chemical terms in MEDLINE abstracts (GENIA). MED is a data set extracted from a semantic annotation of (parts of) two Dutch-language medical encyclopedias. On the chunk-level of this annotation, there are labels for various medical concepts, such as disease names, body parts, and treatments, forming a set of twelve concept types in total. Chunk sizes range from one to a few tokens. The target application for which this data set was developed is a question analyser for a question-answering system for factual medical question. As the input sentences for such a system will typically contain at least one domain-specific concept, all sentences without any concept have been removed from the data set. Using a 90%–10% split for producing training and test sets, there are 428,502 training examples and 47,430 test examples.

> *Bij* [*infantiel botulisme*]$_{disease}$ *kunnen in extreme gevallen* [*ademhalingsproblemen*]$_{symptom}$ *en* [*algehele lusteloosheid*]$_{symptom}$ *optreden.*

The GENIA corpus [5] is a collection of annotated abstracts taken from the National Library of Medicine's MEDLINE database. Apart from part-of-speech tagging information, the corpus annotates a subset of the substances and the biological locations involved in reactions of proteins.

In contrast with the MED data set, GENIA does include sentences that do not contain any concept; in fact, this is the case for many of them. Using a 90%–10% split for producing training and test sets, there are 458,593 training examples and 50,916 test examples.

> *Most hybrids express both* $[KBF1]_{protein}$ *and* $[NF\text{-}kappa\ B]_{protein}$ *in their nuclei , but one hybrid expresses only* $[KBF1]_{protein}$.

Apart from having a similar size, both data sets are alike in the sense that most words are outside chunks; for GENIA, many sentences may even contain no chunks at all. Thus, the class distributions of both tasks are highly skewed, and only a few words are actually relevant and should be assigned a non-negative class. In this respect, the tasks differ from, for example, syntactic sequence labelling tasks such as part-of-speech tagging or base-phrase chunking, where almost all tokens are assigned a relevant class. However, for all tasks mentioned, whenever chunks are present in a sentence, there is likely to be interaction between them, where the presence of one chunk of a certain type may be a strong indication of the presence of another chunk of the same or a different type in the same sentence.

### 2.1 Experimental Setup

The experiments in this study have all been performed using the maximum-entropy classification framework. This method is especially suited for the experiments, since maximum-entropy classifiers can both be used in discrete and in probabilistic mode. Any probabilistic classifier can easily be made to emulate a discrete one by taking the target label with the highest conditional probability as the predicted class. By doing this, the discrete sequence labelling methods and their probabilistic counterparts can be compared objectively without having to take into account differences originating, for example, from classifier biases. The current study uses the maximum-entropy classifier as implemented in the maxent toolkit (version 20040930) by Zhang Le[1].

In Section 4, several different probabilistic sequence classification methods are evaluated. Two of them –a maximum-entropy based conditional markov model, and a maximum-entropy markov model– have been implemented on top of the beforementioned maxent toolkit. For a third, a conditional random field model, the implementation as provided by MALLET [2] has been used.

Instances for all experiments are generated for each token of a sentence, with features for seven-token windows of words and their (predicted) part-of-speech tags. The class labels assigned to the instances form an IOB encoding of the chunks in the sentence, as proposed by Ramshaw and Marcus [6]. In this encoding the class label for a token specifies whether the token is inside (I), outside (O), or at the beginning of a chunk (B). An additional type label appended to this symbol denotes the type of the chunk. The instances are used in exactly this form in all experiments for all methods; no feature selection or construction is performed to optimise the instances for a specific task or method. Keeping the feature vectors unchanged over all experiments and methods is arguably the most objective setup for comparing the results.

Generalisation performance is measured by the precision, recall, and F-score ($\beta = 1$) on correctly identified and labelled entity chunks in test data. Experimental results are presented in terms of a mean score, and an approximate 90%-confidence interval; both of those are estimated with bootstrap resampling [4]. To enhance readability, confidence intervals are assumed to be centred around the mean, where the width of the halves at both sides of the mean is taken to be the maximum of the true widths obtained in the resampling process.

## 3 Discrete Classification

In this section, a maximum-entropy classifier is treated as though it were a discrete classifier. This is achieved by always taking the target label with highest conditional probability. To start with, a *baseline* score has been established, for which a classifier is trained on examples representing fixed-width windows of input symbols only, mapping to IOB-style class labels, as described earlier. This baseline system is then supplemented with three methods for adding sequence-awareness to discrete classifiers: a feedback loop, stacking, and trigram classes. Each of these will be introduced briefly in the following subsections.

[1] URL: `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`

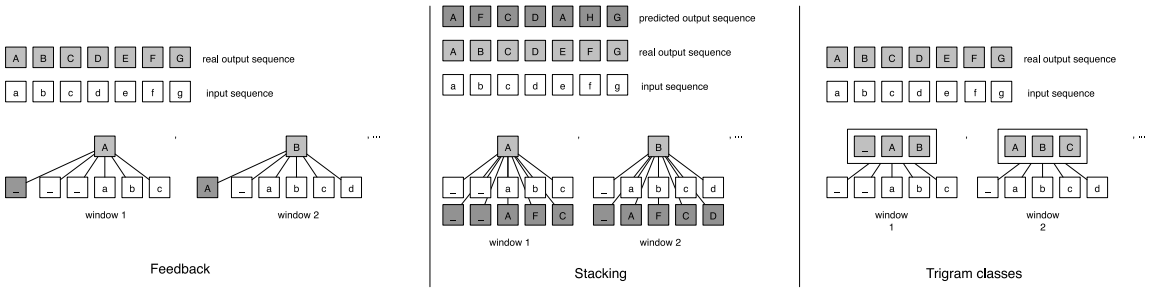


Figure 1: Three types of windowing processes aimed at optimising sequence predictions, generated from an input sequence mapping to an output sequence. Left: a feedback loop copying the previous prediction into the input window. Middle: windows created by the second-stage classifier in a stacking architecture, copying predictions of the first-stage classifier into the input window. Right: windows mapping to trigrams of classes.

Table 1: Mean performance scores and confidence intervals for the various classification methods on the Genia task.

| Method | Precision | Recall | $F_{\beta=1}$ |
|---|---|---|---|
| Baseline | 54.9 ±1.16 | 54.1 ±1.23 | 54.5 ±1.02 |
| Feedback | 67.3 ±1.04 | 57.6 ±1.25 | 62.1 ±1.11 |
| Stacking | 57.8 ±1.21 | 55.3 ±1.07 | 56.5 ±1.11 |
| Trigram | 61.8 ±1.15 | 56.0 ±1.27 | 58.8 ±1.12 |
| cmm | 67.7 ±0.96 | 57.9 ±1.07 | 62.4 ±1.01 |
| memm | 67.1 ±1.14 | 57.7 ±1.13 | 62.1 ±1.15 |
| crf | 66.8 ±1.10 | 59.2 ±1.14 | 62.8 ±1.08 |

## 3.1 The Feedback-loop Method

One method for providing a classifier access to its previous decisions is a feedback-loop approach, which extends the windowing approach by feeding previous decisions of the classifier as features into the current input of the classifier. An early application of this method in the context of maximum-entropy classification is the work of Ratnaparkhi on part-of-speech tagging [7]. The left part of Figure 1 illustrates how the windowing approach is extended with information about previous classifier decisions as input features. The number of decisions fed back into the input can be varied. In the experiments described here, the feedback loop iteratively updates a memory of only the single most recent prediction, which was experimentally found to be an adequate setting.

## 3.2 Stacking

Stacking, a term popularised by Wolpert [9] in an artificial neural network context, refers to a class of meta-learning systems that learn to correct errors made by lower-level classifiers. We implement stacking by adding a windowed sequence of previous and subsequent output class labels to the original input features (here, we copy a window of three predictions to the input, centred around the middle position), and providing these enriched examples as training material to a second-stage classifier. The middle part of Figure 1 illustrates the procedure. Given the (possibly erroneous) output of a first classifier on an input sequence, a certain window of class symbols from that predicted sequence is copied to the input, to act as predictive features for the real class label.

## 3.3 Predicting Class Trigrams

Van den Bosch and Daelemans [8] recently proposed a new discrete method for sequence labelling based on predicting trigrams of class labels. In this method, each token in a sequence is labelled with a complex class label composed of the labels of the token itself, and those of the tokens directly

Table 2: Mean performance scores and confidence intervals for the various classification methods on the MED task.

| METHOD | PRECISION | RECALL | $F_{\beta=1}$ |
|---|---|---|---|
| BASELINE | 62.3 ±1.12 | 60.8 ±1.06 | 61.5 ±0.98 |
| FEEDBACK | 68.5 ±1.16 | 60.0 ±1.13 | 63.9 ±0.89 |
| STACKING | 63.2 ±1.23 | 60.8 ±1.13 | 62.0 ±1.10 |
| TRIGRAM | 66.9 ±1.07 | 59.6 ±1.01 | 63.1 ±1.08 |
| CMM | 68.8 ±1.26 | 59.6 ±1.09 | 63.9 ±0.99 |
| MEMM | 68.8 ±1.09 | 59.3 ±1.26 | 63.7 ±1.09 |
| CRF | 66.8 ±1.14 | 60.2 ±1.14 | 63.4 ±0.99 |

to its left and to its right in the sequence. Doing this for each token of a sequence, every token's label is effectively predicted three times: once by the preceding token's trigram, once by its own trigram, and once by the following token's trigram. Aiming to exploit this redundancy, a simple majority voting rule is applied to the overlapping and possibly conflicting predictions for a certain token. If all three overlapping trigrams predict a different class for a token, the confidence of the classifier for the predicted trigrams is used to break the tie.

The right part of Figure 1 illustrates the procedure by which windows are created with class trigrams. Each windowed instance maps to a class label that incorporates three atomic class labels, namely the focus class label that was the original unigram label, plus its immediate left and right neighbouring class labels.

### 3.4 Results

The top half of Tables 1 and 2 shows the performance scores for the discrete sequence labelling methods introduced in this section, and compares them with the performance of a naive baseline classifier that treats each token as an isolated classification case. The error reductions with respect to F-score attained by the best method, the feedback-loop method, is 6.2% for MED, and 16.7% for GENIA.

## 4 Probabilistic Classification

Rather than suggesting one target label that is expected to be the most likely class for a test instance, as do discrete classifiers, probabilistic classifiers compute the conditional probability for each possible target label. This property opens up interesting possibilities for considering alternative label sequences, different from the one obtained by concatenating all individually most likely token labels. As a consequence of the probabilistic interpretation that can be given to target labels, the probability of an entire sequence of labels can be determined using simple multiplication, provided certain independence requirements are met.

Sequence labelling methods based on probabilistic classifiers can generally be decomposed into two components: the first is concerned with estimating the conditional probability of a candidate output label sequence; the second dictates a procedure for efficiently finding the best label sequence out of all possible candidates. In this section, three different methods based on this general framework are applied to the sample tasks: conditional markov models, maximum-entropy markov models, and conditional random fields.

### 4.1 Conditional Markov Model

Conditional markov models (CMM), as used for example by Ratnaparkhi [7], can be seen as a probabilistic extension of the feedback-loop method described in Section 3.1. As with the method described earlier, a prespecified number of previous decisions of the classifier are fed back to the input as features for the current test instance. Unlike in the discrete case however, classification of

a token does not yield one partial labelling, namely, the partial labelling up to the current token followed by the current classification, but as many partial labellings as there are target labels, namely the partial labelling until the current token followed by any of the possible target labels.

As the use of a feedback loop makes the current classification depend on the results of previous classifications, each token in the sequence has to be classified in the context of each possible partial labelling up to that point. Clearly, this approach, if applied naively, gives rise to an exponential increase in possible partial labellings at each token. Therefore, CMMs employ a beam search to find the eventual best labelling. With beam search, at each point in time, only a prespecified number of partial labellings – those having highest probability – are considered for expansion, all the other candidates are discarded.

### 4.2 Maximum-entropy Markov Model

Another implementation of the idea of supplementing a probabilistic classifier with a search procedure is the maximum-entropy markov model (MEMM), proposed by McCallum et al. [3]. Derived from hidden markov models, MEMMs are modelled after a probabilistic state machine, in which, in the simplest case, a state corresponds to the output label of the previous token, and for each state, a separate conditional probability distribution determines the next state, that is, the output label for the current token, given the feature vector of this token. A slight modification of the Viterbi algorithm is used to determine the optimal path through the state machine given the input sequence.

### 4.3 Conditional Random Fields

Conditional random fields (CRF) [1] have been designed to resolve some of the shortcomings of MEMMs. The main difference lies in the number of probabilistic models used for estimating the conditional probability of a label sequence: MEMMs use a separate probabilistic model for each state, whereas CRFs have a single model for estimating the likelihood of an entire label sequence. The use of a single probabilistic model leads to a more realistic distribution of the probability mass among the alternative paths. As a result, CRFs tend to be less biased towards states with few successor states than CMMs and MEMMs.

### 4.4 Results

The performance scores for the probabilistic sequence labelling methods applied to the two benchmark tasks are listed in the bottom half of Tables 1 and 2. On MED, CMM attains the highest score, equal to the one obtained by the discrete feedback-loop method; the error reduction in F-score is 6.2%. The overall highest score on GENIA is obtained by CRF, with an error reduction of 18.2%. However, in none of the cases is the difference with the other high-ranked methods significant.

## 5 Discussion

The previous two sections reported on a series of experiments in which several machine learning approaches to domain-specific entity chunking have been evaluated. The approaches can be divided into an approach that ignores any sequential context apart from its window features, and those that have been extended specifically to deal with sequentially-structured data. Members of the latter group, in turn, can be categorised as either discrete or probabilistic methods. All of these approaches have been tested against two data sets; both of which are quite similar domain-specific entity chunking tasks. However, one striking difference is the presence of sentences without any concept in GENIA, and the absence of those in MED.

In order to put the results obtained in the previous two sections in perspective, this section will review the effect of the factors mentioned above.

### 5.1 Sequence-aware vs. sequence-unaware methods

As was already mentioned in the introduction of this paper, machine learning approaches not specifically equipped for classifying sequentially-structured data often have difficulty performing

tasks in which token labels are sequentially correlated; a situation which can be improved upon by more special-purpose sequence classification methods. The findings of the current study can only confirm this assumption. Comparing the performance of the baseline system with that of the best-performing sequence-oriented approach, the error in F-score is reduced by 18.2% on GENIA, while on MED a somewhat more modest but still significant reduction of 6.2% is observed.

A remarkable observation is that the sequence-oriented methods, be they discrete or probabilistic, all improve precision more strongly than recall. Whereas the two are reasonably in balance for the baseline system, there is a striking imbalance for the sequence methods. This situation is most apparent on the MED task, where the recall is not improved at all. From this, it can be concluded that the sequence methods do not necessarily predict more true positives –since recall would benefit from this as well– but rather manage to decrease the total number of positive predictions without negatively affecting the number of true positives. To a lesser extent, the same goes for the GENIA task.

### 5.2 Probabilistic vs. discrete approaches

The sequence-oriented methods surveyed in this paper are either based on a probabilistic base classifier, in case of which various different output label sequences are traded off on the sequence level; or on a discrete classifier, which only has a single opportunity for classifying each input token. In recent years, probabilistic approaches tend to be preferred for sequence-oriented tasks. It is claimed that discrete methods, being unable to support viterbi-like inference procedures, must resort to sub-optimal methods to classify sequential data [3].

However, the results presented in this paper do not support this claim. In fact, no significant differences in performance have been observed between the best performing probabilistic and discrete methods; neither in precision, nor in recall, and consequently not in F-score either. These findings suggest that, although discrete methods are unable to trade off entire sequence labellings on a global level, at least for the two tasks evaluated in the current study, the approaches used by the discrete methods provide suitable alternatives. Whether this observation holds more generally for other types of sequence processing tasks as well, will need to be pointed out by more extensive experimental studies.

### 5.3 Concept sparseness

One of the main differences between the two data sets used in this study lies in the distribution of concepts in sentences. In the MED data, every sentence contains at least one concept, whereas in GENIA, sentences without any concept occur quite frequently as well. This difference might provide a partial explanation for the different behaviour of the sequence methods with respect to recall on the two data sets. From the large difference between precision and recall, we already concluded that the sequence methods primarily improve precision by predicting fewer positives, rather than predicting more true positives. However, when predicting fewer positives, there is always the risk of missing some positive labels, that is, predicting false negatives, and thereby committing recall errors. A possible explanation for the better recall performance on GENIA then may be the fact that the chance of committing recall errors is simply lower on data sets that comprise a lower percentage of chunks to begin with.

## 6 Conclusion

Classifiers trained on entity chunking tasks that make isolated, near-sighted decisions on output symbols and that do not optimise the resulting output sequences afterwards or internally through a feedback loop, tend to produce weaker models for sequence processing tasks than classifiers that do. The two entity chunking tasks investigated in this paper are challenging tasks; not only because they demand the classifier to be able to segment and label variable-width chunks while obeying the syntax of the chunk analysis, but also because positive examples of labelled chunks are scattered sparsely in the data.

In this paper, special-purpose sequence labelling methods have been classified into two main categories. One of these categories expects the underlying classifier to be probabilistic, that is,

rather than predicting a single discrete class for each test instance, it should estimate the conditional probability of each possible target class. Given such a probabilistic output, a search algorithm is employed to find the most likely label sequence, as opposed to the sequence of individually most likely class predictions.

The other category uses classifiers that do not model conditional probability distributions, but instead predict a single most likely output class for each test instance. For this type of classifier there is a wide range of methods that aim to improve performance on sequence labelling tasks, including adding a feedback-loop to the classification procedure, classifier stacking, and predicting class trigrams.

In a series of experiments, several discrete, and probabilistic sequence labelling methods implemented on top of a maximum-entropy base classifier were applied to two benchmark tasks; both of them, high-level entity chunking tasks in the medical domain. Results from these experiments clearly show that the enhanced methods easily improve upon a baseline classifier that considers each token as a separate, isolated classification case. More surprising is the fact that there are hardly any differences in performance among the various sequence-oriented methods. In particular, when it comes to the type of sequence-oriented entity chunking tasks targeted in this study, we did not find any evidence supporting the claim that probabilistic methods have an advantage over discrete ones. This leads us to the assertion that, for this specific type of sequential tasks, no method is fundamentally more suited than another. In practise, this means that other factors than whether or not a classifier is capable of probabilistic predictions, may guide the choice for the specific type of learning method to use. In future work, we intend to investigate whether this result generalises to more types of sequence processing tasks in natural language processing.

**Acknowledgements**
This research was funded by NWO, the Netherlands Organization for Scientific Research, as part of the IMIX Programme.

# References

[1] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, 2001.

[2] A. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[3] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA, 2000.

[4] E. Noreen. *Computer-intensive methods for testing hypotheses: an introduction.* John Wiley and sons, 1989.

[5] T. Ohta, Y. Tateisi, J.-D. Kim, H. Mima, and J. Tsujii. Genia corpus: an annotated research abstract corpus in molecular biology domain. In *Human Language Technology Conference (HLT 2002)*, pages 73–77, 2002.

[6] L.A. Ramshaw and M.P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL/SIGDAT Workshop on Very Large Corpora, Cambridge, Massachusetts, USA*, pages 82–94, 1995.

[7] A. Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, May 17-18, 1996, University of Pennsylvania*, 1996.

[8] A. Van den Bosch and W. Daelemans. Improving sequence segmentation learning by predicting trigrams. In I. Dagan and D. Gildea, editors, *Proceedings of the Ninth Conference on Computational Natural Language Learning*, 2005.

[9] D. H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241–259, 1992.