A Lemma-Based Approach to a Maximum Entropy Word Sense Disambiguation System for Dutch

Tanja Gaustad
Alfa-Informatica, University of Groningen
Postbus 716
9700 AS Groningen,
The Netherlands,
T.Gaustad@let.rug.nl

Abstract

In this paper, we present a corpus-based supervised word sense disambiguation (WSD) system for Dutch which combines statistical classification (maximum entropy) with linguistic information. Instead of building individual classifiers per ambiguous wordform, we introduce a lemma-based approach. The advantage of this novel method is that it clusters all inflected forms of an ambiguous word in one classifier, therefore augmenting the training material available to the algorithm. Testing the lemma-based model on the Dutch SENSEVAL-2 test data, we achieve a significant increase in accuracy over the wordform model. Also, the WSD system based on lemmas is smaller and more robust.

1 Introduction: WSD for Dutch

A major problem in natural language processing (NLP) for which no satisfactory solution has been found to date is word sense disambiguation (WSD). WSD refers to the resolution of lexical semantic ambiguity and its goal is to attribute the correct sense(s) to words in a certain context. For instance machine translation, information retrieval or document extraction could all benefit from the accurate disambiguation of word senses.

The WSD system for Dutch[1] presented here is a corpus-based supervised algorithm combining statistical classification with various kinds of linguistic information. The intuition behind the system is that linguistic information is beneficial for WSD which means that it will improve results over purely statistical approaches. The linguistic information includes lemmas, part-of-speech (PoS), and the context around the ambiguous word.

In this paper, we focus on a lemma-based approach to WSD for Dutch. So far, systems built individual classifiers for each ambiguous wordform (Hendrickx et al., 2002; Hoste et al., 2002). In the system presented here, the classifiers built for each ambiguous word are based on its lemma instead. Lemmatization allows for more compact and generalizable data by clustering all inflected forms of an ambiguous word together, an effect already commented on by Yarowsky (1994). The more inflection in a language, the more lemmatization will help to compress and generalize the data. In the case of our WSD system this means that less classifiers have to be built therefore adding up the training material available to the algorithm for each ambiguous wordform. Accuracy is expected to increase for the lemma-based model in comparison to the wordform model.

The paper is structured as follows: First, we will present the dictionary-based lemmatizer for Dutch which was used to lemmatize the data, followed by a detailed explanation of the lemma-based approach adopted in our WSD system. Next, the statistical classification algorithm, namely maximum entropy, and Gaussian priors (used for smoothing purposes) are introduced. We will then proceed to describe the corpus, the corpus preparation, and the system settings. We conclude the paper with results on the Dutch SENSEVAL-2 data, their evaluation and ideas for future work.

2 Dictionary-Based Lemmatizer for Dutch

Statistical classification systems, like our WSD system, determine the most likely class for a given instance by computing how likely the words or linguistic features in the instance are for any given class. Estimating these probabilities is difficult, as corpora contain lots of different, often infrequent, words. *Lemmatization*[2] is a method that can be used to reduce the number of wordforms that need to be taken into consideration, as estimation is more reliable for frequently occurring data.

[1] The interest in Dutch lies grounded in the fact that we are working in the context of a project concerned with developing NLP tools for Dutch (see `http://www.let.rug.nl/~vannoord/alp`).

[2] We chose to use lemmatization and not stemming because the lemma (or canonical dictionary entry form) can be used to look up an ambiguous word in a dictionary or an ontology like e.g. WordNet. This is not the case for a stem.

Lemmatization reduces all inflected forms of a word to the same lemma. The number of different lemmas in a training corpus will therefore in general be much smaller than the number of different wordforms, and the frequency of lemmas will therefore be higher than that of the corresponding individual inflected forms, which in turn suggests that probabilities can be estimated more reliably.

For the experiments in this paper, we used a lemmatizer for Dutch with dictionary lookup. Dictionary information is obtained from Celex (Baayen et al., 1993), a lexical database for Dutch. Celex contains 381,292 wordforms and 124,136 lemmas for Dutch. It also contains the PoS associated with the lemmas. This information is useful for disambiguation: in those cases where a particular wordform has two (or more) possible corresponding lemmas, the one matching the PoS of the wordform is chosen. Thus, in a first step, information about wordforms, their respective lemmas and their PoS is extracted from the database.

Dictionary lookup can be time consuming, especially for large dictionaries such as Celex. To guarantee fast lookup and a compact representation, the information extracted from the dictionary is stored as a finite state automaton (FSA) using Daciuk's (2000) FSA morphology tools.[3] Given a wordform, the compiled automaton provides the corresponding lemmas in time linear to the length of the input word. Contrasting this dictionary-based lemmatizer with a simple suffix stripper, such as the Dutch Porter Stemmer (Kraaij and Pohlman, 1994), our lemmatizer is more accurate, faster and more compact (see (Gaustad and Bouma, 2002) for a more elaborate description and evaluation).

During the actual lemmatization procedure, the FSA encoding of the information in Celex assigns every wordform all its possible lemmas. For ambiguous wordforms, the lemma with the same PoS as the wordform in question is chosen. All wordforms that were not found in Celex are processed with a morphological guessing automaton.[4]

The key features of the lemmatizer employed are that it is fast, compact and accurate.

3 Lemma-Based Approach

As we have mentioned in the previous section, lemmatization collapses all inflected forms of a given word to the same lemma. In our system, separate classifiers are built for every ambiguous wordform. Normally, this implies that the basis for grouping occurrences of particular ambiguous words together is that their wordform is the same. Alternatively, we chose for a model constructing classifiers based on *lemmas* therefore reducing the number of classifiers that need to be made.

As has already been noted by Yarowsky (1994), using lemmas helps to produce more concise and generic evidence than inflected forms. Therefore building classifiers based on lemmas increases the data available to each classifier. We make use of the advantage of clustering all instances of e.g. one verb in a single classifier instead of several classifiers (one for each inflected form found in the data). In this way, there is more training data per ambiguous wordform available to each classifier. The expectation is that this should increase the accuracy of our maximum entropy WSD system in comparison to the wordform-based model.

Figure 1 shows how the system works. During training, every wordform is first checked for ambiguity, i.e. whether it has more than one sense associated with all its occurrences. If the wordform is ambiguous, the number of lemmas associated with it is looked up. If the wordform has one lemma, all occurrences of this lemma in the training data are used to make the classifier for that particular wordform—and others with the same lemma. If a wordform has more than one lemmas, a classifier based on the wordform is built. This strategy has been decided on in order to be able to treat all ambiguous words, notwithstanding lemmatization errors or wordforms that can genuinely be assigned two or more lemmas.

An example of a word that has two different lemmas depending on the context is *boog*: it can either be the past tense of the verb *buigen* ('to bend') or the noun *boog* ('arch'). Since the Dutch SENSEVAL-2 data is not only ambiguous with regard to meaning but also with regard to PoS, both lemmas are subsumed in the wordform classifier for *boog*.

During testing, we check for each word whether there is a classifier available for either its wordform or its lemma and apply that classifier to the test instance.

4 Maximum Entropy Word Sense Disambiguation System

Our WSD system is founded on the idea of combining statistical classification with linguistic sources of knowledge. In order to be able to take full advantage of the linguistic information, we need a classification algorithm capable of incorporating the information provided. The main advantage of maximum entropy modeling is that heterogeneous and

[3] Available at `http://www.eti.pg.gda.pl/~jandac/fsa.html`

[4] Also available from the FSA morphology tools (Daciuk, 2000).

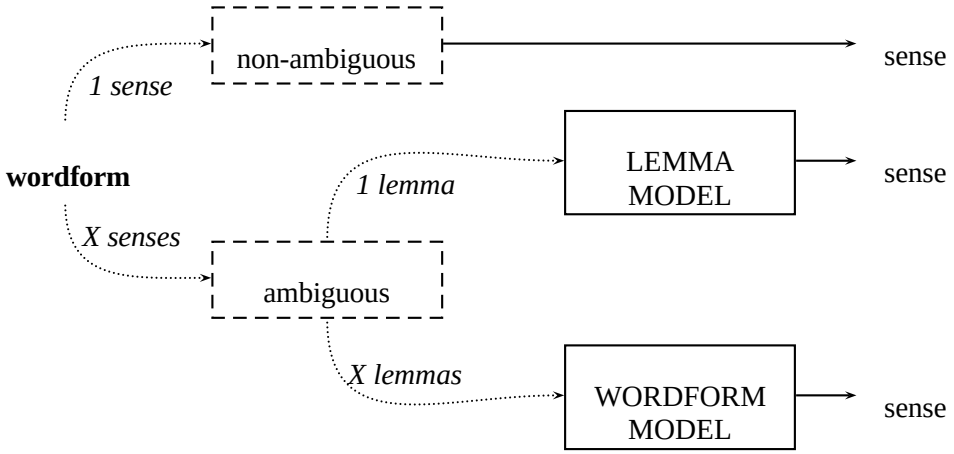

Figure 1: Schematic overview of the lemma-based approach for our WSD System for Dutch

overlapping information can be integrated into a single statistical model. Other learning algorithms, like e.g. decision lists, only take the strongest feature into account, whereas maximum entropy combines them all. Also, no independence assumptions as in e.g. Naive Bayes are necessary.

We will now describe the different steps in putting together the WSD system we used to incorporate and test our lemma-based approach, starting with the introduction of maximum entropy, the machine learning algorithm used for classification. Then, smoothing with Gaussian priors will be explained.

4.1 Maximum Entropy Classification

Several problems in NLP have lent themselves to solutions using statistical language processing techniques. Many of these problems can be viewed as a *classification task* in which linguistic classes have to be predicted given a context.

The statistical classifier used in the experiments reported in this paper is a *maximum entropy classifier* (Berger et al., 1996; Ratnaparkhi, 1997b). Maximum entropy is a general technique for estimating probability distributions from data. A probability distribution is derived from a set of events based on the computable qualities (characteristics) of these events. The characteristics are called *features*, and the events are sets of feature values.

If nothing about the data is known, estimating a probability distribution using the principle of maximum entropy involves selecting the most uniform distribution where all events have equal probability. In other words, it means selecting the distribution which maximises the entropy.

If data is available, a number of features extracted from the labeled training data are used to derive a set of constraints for the model. This set of constraints characterises the class-specific expectations for the distribution. So, while the distribution should maximise the entropy, the model should also satisfy the constraints imposed by the training data. A maximum entropy model is thus the model with maximum entropy of all models that satisfy the set of constraints derived from the training data.

The model consists of a set of features which occur on events in the training data. Training itself amounts to finding weights for each feature using the following formula:

$$p(c|x) = \frac{1}{Z} exp\left(\sum_{i=1}^{n} \lambda_i f_i(x,c)\right)$$

where the property function $f_i(x,c)$ represents the number of times feature i is used to find class c for event x, and the weights λ_i are chosen to maximise the likelihood of the training data and, at the same time, maximise the entropy of p. Z is a normalizing constant, constraining the distribution to sum to 1 and n is the total number of features.

This means that during training the weight λ_i for each feature i is computed and stored. During testing, the sum of the weights λ_i of all features i found in the test instances is computed for each class c and the class with the highest score is chosen.

A big advantage of maximum entropy modeling is that the features include any information which might be useful for disambiguation. Thus, dissimilar types of information, such as various kinds of linguistic knowledge, can be combined into a single model for WSD without having to assume independence of the different features. Furthermore, good results have been produced in other areas of

NLP research using maximum entropy techniques (Berger et al., 1996; Koeling, 2001; Ratnaparkhi, 1997a).

4.2 Smoothing: Gaussian Priors

Since NLP maximum entropy models usually have lots of features and lots of sparseness (e.g. features seen in testing not occurring in training), smoothing is essential as a way to optimize the feature weights (Chen and Rosenfeld, 2000; Klein and Manning, 2003). In the case of the Dutch SENSEVAL-2, for many ambiguous words there is little training data available, therefore making smoothing essential.

The intuition behind Gaussian priors is that the parameters in the maximum entropy model should not be too large because of optimization problems with infinite feature weights. In other words: we enforce that each parameter will be distributed according to a Gaussian prior with mean μ and variance σ^2. This prior expectation over the distribution of parameters penalizes parameters for drifting too far from their mean prior value which is $\mu = 0$.

Using Gaussian priors has a number of effects on the maximum entropy model. We trade off some expectation-matching for smaller parameters. Also, when multiple features can be used to explain a data point, the more common ones generally receive more weight. Last but not least accuracy generally goes up and convergence is faster.

In the current experiments the Gaussian prior was set to $\sigma^2 = 1000$ (based on preliminary experiments) which led to an overall increase of at least 0.5% when compared to a model which was built without smoothing.

5 Corpus Preparation and Building Classifiers

In the context of SENSEVAL-2[5], the first sense-tagged corpus for Dutch was made available (see (Hendrickx and van den Bosch, 2001) for a detailed description). The training section of the Dutch SENSEVAL-2 dataset contains approximately 120,000 tokens and 9,300 sentences, whereas the test section consists of ca. 40,000 tokens and 3,000 sentences.

In contrast to the English WSD data available from SENSEVAL-2, the Dutch WSD data is not only ambiguous in word senses, but also with regard to PoS. This means that accurate PoS information is important in order for the WSD system to accurately achieve morpho-syntactic as well as semantic disambiguation.

[5]See `http://www.senseval.org/` for more information on SENSEVAL and for downloads of the data.

First, the corpus is lemmatized (see section 2) and part-of-speech-tagged. We used the Memory-Based tagger MBT (Daelemans et al., 2002a; Daelemans et al., 2002b) with the (limited) WOTAN tag set (Berghmans, 1994; Drenth, 1997) to PoS tag our data (see (Gaustad, 2003) for an evaluation of different PoS-taggers on this task). Since we are only interested in the main PoS-categories, we discarded all additional information from the assigned PoS. This resulted in 12 different tags being kept. In the current experiments, we included the PoS of the ambiguous wordform (important for the morpho-syntactic disambiguation) and also the PoS of the context words or lemmas.

After the preprocessing (lemmatization and PoS tagging), for each ambiguous wordform[6] all instances of its occurrence are extracted from the corpus. These instances are then transformed into *feature vectors* including the features specified in a particular model. The model we used in the reported experiments includes information on the wordform, its lemma, its PoS, contextwords to the left and right as well as the context PoS, and its sense/class.

(1) Nu ging hij bloemen plukken en maakte
now went he flowers pick and made
er een krans van.
it a crown of
'Now he went to pick flowers and made a crown of it.'

Below we show an example of a feature vector for the ambiguous word *bloem* ('flower'/'flour') in sentence 1:

```
bloemen bloem N nu gaan hij Adv
V Pron plukken en maken V Conj V
bloem_plant
```

The first slot represents the ambiguous wordform, the second its lemma, the third the PoS of the ambiguous wordform, the fourth to twelfth slots contain the context lemmas and their PoS (left before right), and the last slot represents the sense or class. Various preliminary experiments have shown a context size of ± 3 context words, i.e. 3 words to the left and 3 words to the right of the ambiguous word, to achieve the best and most stable results. Only context words within the same sentence as the ambiguous wordform were taken into account.

Earlier experiments showed that using lemmas as context instead of wordforms increases accuracy

[6]A wordform is 'ambiguous' if it has two or more different senses/classes in the training data. The sense '=' is seen as marking the basic sense of a word/lemma and is therefore also taken into account.

due to the compression achieved through lemmatization (as explained earlier in this paper and put to practice in the lemma-based approach). With lemmas, less context features have to be estimated, therefore counteracting data sparseness.

In the experiments presented here, no threshold was used. Experiments have shown that building classifiers even for wordforms with very few training instances yields better results than applying a frequency threshold and using the baseline count (assigning the most frequent sense) for wordforms with an amount of training instances below the threshold. It has to be noted, though, that the effect of applying a threshold may depend on the choice of learning algorithm.

6 Results and Evaluation

In order to be able to evaluate the results from the lemma-based approach, we also include results based on wordform classifiers. During training with wordform classifiers, 953 separate classifiers were built.

With the lemma-based approach, 669 classifiers were built in total during training, 372 based on the lemma of an ambiguous word (subsuming 656 wordforms) and 297 based on the wordform. A total of 512 unique ambiguous wordforms was found in the test data. 438 of these were classified using the classifiers built from the training data, whereas only 410 could be classified using the wordform model (see table 1 for an overview).

We include the accuracy of the WSD system on all words for which classifiers were built (ambig) as well as the overall performance on all words (all), including the non-ambiguous ones. This makes our results comparable to other systems which use the same data, but maybe a different data split or a different number of classifiers (e.g. in connection with a frequency threshold applied). The baseline has been computed by always choosing the most frequent sense of a given wordform in the test data.

The results in table 2 show the average accuracy for the two different approaches. The accuracy of both approaches improves significantly (when applying a paired sign test with a confidence level of 95%) over the baseline. This demonstrates that the general idea of the system, to combine linguistic features with statistical classification, works well. Focusing on a comparison of the two approaches, we can clearly see that the lemma-based approach works significantly better than the wordform only model, thereby verifying our hypothesis.

Another advantage of the approach proposed, besides increasing the classification accuracy, is that less classifiers need to be built during training and therefore the WSD system based on lemmas is smaller. In an online application, this might be an important aspect of the speed and the size of the application. It should be noted here that the degree of generalization through lemmatization strongly depends on the data. Only inflected wordforms occurring in the corpus are subsumed in one lemma classifier. The more different inflected forms the training corpus contains, the better the "compression rate" in the WSD model. Added robustness is a further asset of our system. More wordforms could be classified with the lemma-based approach compared to the wordform-based one (438 vs. 410).

In order to better assess the real gain in accuracy from the lemma-based model, we also evaluated a subpart of the results for the lemma-based and the wordform-based model, namely the accuracy of those wordforms which were classified based on their lemma in the former approach, but based on their wordform in the latter case. The comparison in table 3 clearly shows that there is much to be gained from lemmatization. The fact that inflected wordforms are subsumed in lemma classifiers leads to an error rate reduction of 8% and a system with less than half as many classifiers.

In table 4, we see a comparison with another WSD systems for Dutch which uses Memory-Based learning (MBL) in combination with local context (Hendrickx et al., 2002). A big difference with the system presented in this article is that extensive parameter optimization for the classifier of each ambiguous wordform has been conducted for the MBL approach. Also, a frequency threshold of minimally 10 training instances was applied, using the baseline classifier for all words below that threshold. As we can see, our lemma-based WSD system scores the same as the Memory-Based WSD system, without extensive "per classifier" parameter optimization. According to Daelemans and Hoste (2002), different machine learning results should be compared once all parameters have been optimized for all classifiers. This is not the case in our system, and yet it achieves the same accuracy as an optimized model. Optimization of parameters for each ambiguous wordform and lemma classifier might help increase our results even further.

7 Conclusion and Future Work

In this paper, we have introduced a lemma-based approach for a statistical WSD system using maximum entropy and a number of linguistic sources of information. This novel approach uses the advantage of more concise and more generalizable in-

		lemma-based	wordforms
Training	# classifiers built	669	953
	based on wordforms	297	953
	based on lemmas	372	na
	# wordforms subsumed	656	na
Testing	# unique ambiguous wordforms	512	512
	# classifiers used	387	410
	based on wordforms	230	410
	based on lemmas	70	na
	# wordforms subsumed	208	na
	# wordforms seen 1st time	74	102

Table 1: Overview of classifiers built during training and used in testing with the lemma-based and the wordform-based approach

Model	ambig	all
baseline all ambiguous words	78.47	89.44
wordform classifiers	83.66	92.37
lemma-based classifiers	**84.15**	**92.45**

Table 2: WSD Results (in %) with the lemma-based approach compared to classifiers based on wordforms

formation contained in lemmas as key feature: classifiers for individual ambiguous words are built on the basis of their lemmas, instead of wordforms as has traditionally been done. Therefore, more training material is available to each classifier and the resulting WSD system is smaller and more robust.

The lemma-based approach has been tested on the Dutch SENSEVAL-2 data set and resulted in a significant improvement of the accuracy achieved over the system using the traditional wordform based approach. In comparison to earlier results with a Memory-Based WSD system, the lemma-based approach performs the same, involving less work (no parameter optimization).

A possible extension of the present approach is to include more specialized feature selection and also to optimize the settings for each ambiguous wordform instead of adopting the same strategy for all words in the corpus. Furthermore, we would like to test the lemma-based approach in a multi-classifier voting scheme.

Acknowledgments

This research was carried out within the framework of the PIONIER Project *Algorithms for Linguistic Processing*. This PIONIER Project is funded by NWO (Dutch Organization for Scientific Research) and the University of Groningen. We are grateful to Gertjan van Noord and Menno van Zaanen for comments and discussions.

References

R. Harald Baayen, Richard Piepenbrock, and Afke van Rijn. 1993. The CELEX lexical database (CD-ROM). Linguistic Data Consortium, University of Pennsylvania, Philadelphia.

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Johan Berghmans. 1994. WOTAN—een automatische grammaticale tagger voor het Nederlands. Master's thesis, Nijmegen University, Nijmegen.

Stanley Chen and Ronald Rosenfeld. 2000. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.

Jan Daciuk. 2000. Finite state tools for natural language processing. In *Proceedings of the COLING 2000 Workshop "Using Toolsets and Architectures to Build NLP Systems"*, pages 34–37, Centre Universitaire, Luxembourg.

Walter Daelemans and Véronique Hoste. 2002. Evaluation of machine learning methods for natural language processing tasks. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 755–760, Las Palmas, Gran Canaria.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2002a. MBT: Memory-Based tagger, reference guide. Tech-

Model	ambig	#classifiers
baseline	76.77	192
wordform classifiers	78.66	192
lemma-based classifiers	**80.39**	**70**

Table 3: Comparison of results (in %) for wordforms with different classifiers in the lemma-based and wordform-based approach

Model	all
baseline all words	89.4
wordform classifiers	92.4
lemma-based classifiers	92.5
Hendrickx et al. (2002)	92.5

Table 4: Comparison of results (in %) on the Dutch SENSEVAL-2 Data with different WSD systems

nical Report ILK 02-09, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University, Tilburg. version 1.0.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2002b. TiMBL: Tilburg Memory-Based learner, reference guide. Technical Report ILK 02-10, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University, Tilburg. version 4.3.

Erwin Drenth. 1997. Using a hybrid approach towards Dutch part-of-speech tagging. Master's thesis, Alfa-Informatica, University of Groningen, Groningen.

Tanja Gaustad and Gosse Bouma. 2002. Accurate stemming of Dutch for text classification. In Mariët Theune, Anton Nijholt, and Hendri Hondorp, editors, *Computational Linguistics in the Netherlands 2001*, Amsterdam. Rodopi.

Tanja Gaustad. 2003. The importance of high quality input for WSD: An application-oriented comparison of part-of-speech taggers. In *Proceedings of the Australasian Language Technology Workshop (ALTW 2003)*, pages 65–72, Melbourne.

Iris Hendrickx and Antal van den Bosch. 2001. Dutch word sense disambiguation: Data and preliminary results. In *Proceedings of Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 13–16, Toulouse.

Iris Hendrickx, Antal van den Bosch, Véronique Hoste, and Walter Daelemans. 2002. Dutch word sense disambiguation: Optimizing the localness of context. In *Proceedings of the ACL 2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia.

Véronique Hoste, Iris Hendrickx, Walter Daelemans, and Antal van den Bosch. 2002. Parameter optimization for machine-learning of word sense disambiguation. *Natural Language Engineering, Special Issue on Word Sense Disambiguation Systems*, 8(4):311–325.

Dan Klein and Christopher Manning. 2003. Maxent models, conditional estimation, and optimization without the magic. ACL 2003 Tutorial Notes. Sapporo.

Rob Koeling. 2001. *Dialogue-Based Disambiguation: Using Dialogue Status to Improve Speech Understanding*. Ph.D. thesis, Alfa-Informatica, University of Groningen, Groningen.

Wessel Kraaij and Renée Pohlman. 1994. Porter's stemming algorithm for Dutch. In L.G.M. Noordman and W.A.M. de Vroomen, editors, *Informatiewetenschap 1994: Wetenschapelijke bijdragen aan de derde STINFON Conferentie*, pages 167–180, Tilburg.

Adwait Ratnaparkhi. 1997a. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Providence.

Adwait Ratnaparkhi. 1997b. A simple introduction to maximum entropy models for natural language processing. Technical Report IRCS Report 97-08, IRCS, University of Pennsylvania, Philadelphia.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *32th Annual Meeting of the Association for Computational Linguistics (ACL 1994)*, Las Cruces.