Improvement of a Whole Sentence Maximum Entropy Language Model Using Grammatical Features*

Fredy Amaya† and **José Miguel Benedí**
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de vera s/n, 46022-Valencia (Spain)
{famaya, jbenedi}@dsic.upv.es

Abstract

In this paper, we propose adding long-term grammatical information in a Whole Sentence Maximun Entropy Language Model (WSME) in order to improve the performance of the model. The grammatical information was added to the WSME model as features and were obtained from a Stochastic Context-Free grammar. Finally, experiments using a part of the Penn Treebank corpus were carried out and significant improvements were acheived.

1 Introduction

Language modeling is an important component in computational applications such as speech recognition, automatic translation, optical character recognition, information retrieval etc. (Jelinek, 1997; Borthwick, 1997). Statistical language models have gained considerable acceptance due to the efficiency demonstrated in the fields in which they have been applied (Bahal et al., 1983; Jelinek et al., 1991; Ratnapharkhi, 1998; Borthwick, 1999).

Traditional statistical language models calculate the probability of a sentence s using the chain rule:

$$p(s) = p(w_1 w_2 \dots w_n) = \prod_{i=1}^{n} p(w_i|h_i) \quad (1)$$

where $h_i = w_1 \dots w_{i-1}$, which is usually known as the history of w_i. The effort in the language modeling techniques is usually directed to the estimation of $p(w_i|h_i)$. The language model defined by the expression $p(w_i|h_i)$ is named the conditional language model. In principle, the determination of the conditional probability in (1) is expensive, because the possible number of word sequences is very great. Traditional conditional language models assume that the probability of the word w_i does not depend on the entire history, and the history is limited by an equivalence relation ϕ, and (1) is rewritten as:

$$p(s) = p(w_1 w_2 \dots w_n) \approx \prod_{i=1}^{n} p(w_i|\phi(h_i)) \quad (2)$$

The most commonly used conditional language model is the n-gram model. In the n-gram model, the history is reduced (by the equivalence relation) to the last $n-1$ words. The power of the n-gram model resides in: its consistence with the training data, its simple formulation, and its easy implementation. However, the n-gram model only uses the information provided by the last $n-1$ words to predict the next word and so only makes use of local information. In addition, the value of n must be low (≤ 3) because for $n > 3$ there are problems with the parameter estimation.

Hybrid models have been proposed, in an attempt to supplement the local information with long-distance information. They combine different types of models, like n-grams, with long-distance information, generally by means of linear interpolation, as has been shown in (Belle-

*This work has been partially supported by the Spanish CYCIT under contract (TIC98/0423-C06).

†Granted by Universidad del Cauca, Popayán (Colombia)

garda, 1998; Chelba and Jelinek, 2000; Benedí and Sánchez, 2000).

A formal framework to include long-distance and local information in the same language model is based on the Maximum Entropy principle (ME). Using the ME principle, we can combine information from a variety of sources into the same language model (Berger et al., 1996; Rosenfeld, 1996). The goal of the ME principle is that, given a set of features (pieces of desired information contained in the sentence), a set of functions $f_1, \ldots f_m$ (measuring the contribution of each feature to the model) and a set of constraints[1], we have to find the probability distribution that satisfies the constraints and minimizes the relative entropy (Divergence of Kullback-Leibler) $D(p||p_0)$, with respect to the distribution p_0.

The general Maximum Entropy probability distribution relative to a prior distribution p_0 is given by the expression:

$$p(s) = \frac{1}{Z} p_0(s) e^{\sum_{i=1}^{n} \lambda_i f_i(s)} \qquad (3)$$

where Z is the normalization constant and λ_i are parameters to be found. The λ_i represent the contribution of each feature to the distribution.

From (3) it is easy to derive the Maximum Entropy conditional language model (Rosenfeld, 1996): if X is the context space and W is the vocabulary, then XxW is the states space, and if $(x, y) \in X$xW then:

$$p(y|x) = \frac{1}{Z(x)} e^{\sum_{i=1}^{n} \lambda_i f_i(x,y)} \qquad (4)$$

and $Z(x)$:

$$z(x) = \sum_{y} e^{\sum_{i=1}^{m} \lambda_i f_i(x,y)} \qquad (5)$$

where $z(x)$ is the normalization constant depending on the context x. Although the conditional ME language model is more flexible than n-gram models, there is an important obstacle to its general use: conditional ME language models have a high computational cost (Rosenfeld, 1996), specially the evaluation of the normalization constant (5).

[1]The constraints usually involve the equality between theoretical expectation and the empirical expectation over the training corpus.

Although we can incorporate local information (like n-grams) and some kinds of long-distance information (like triggers) within the conditional ME model, the global information contained in the sentence is poorly encoded in the ME model, as happens with the other conditional models.

There is a language model which is able to take advantage of the local information and at the same time allows for the use of the global properties of the sentence: the Whole Sentence Maximum Entropy model (WSME) (Rosenfeld, 1997). We can include classical information such us n-grams, distance n-grams or triggers and global properties of the sentence, as features into the WSME framework. Besides the fact that the WSME model training procedure is less expensive than the conditional ME model, the most important training step is based on well-developed statistical sampling techniques. In recent works (Chen and Rosenfeld, 1999a), WSME models have been successfully trained using features of n-grams and distance n-grams.

In this work, we propose adding information to the WSME model which is provided by the grammatical structure of the sentence. The information is added in the form of features by means of a Stochastic Context-Free Grammar (SCFG). The grammatical information is combined with features of n-grams and triggers.

In section 2, we describe the WSME model and the training procedure in order to estimate the parameters of the model. In section 3, we define the grammatical features and the way of obtaining them from the SCFG. Finally, section 4 presents the experiments carried out using a part of the Wall Street Journal in order evalute the behavior of this proposal.

2 Whole Sentence Maximum Entropy Model

The whole sentence Maximum Entropy model directly models the probability distribution of the complete sentence[2]. The WSME language model has the form of (3).

In order to simplify the notation we write $\mu_i \equiv e^{\lambda_i}$, and define:

[2]By sentence, we understand any sequence of linguistic units that belongs to a certain vocabulary.

$$R(s) = \prod_{i=1}^{m} \mu_i^{f_i(s)} \quad (6)$$

so (3) is written as:

$$p(s) = \frac{1}{Z} p_0(s) R(s) \quad (7)$$

where s is a sentence and the μ_i are now the parameters to be learned.

The training procedure to estimate the parameters of the model is the Improved Iterative Scaling algorithmn (IIS) (Della Pietra et al., 1995). IIS is based on the change of the log-likelihood over the training corpus Ω, when each of the parameters changes from λ_i to $\lambda_i + \delta_i$, $\delta_i \in \mathbf{R}$. Mathematical considerations on the change in the log-likelihood give the training equation:

$$\sum_s p(s) f_i(s) e^{\delta_i f^{\#}(s)} - \sum_{w \in \Omega} \tilde{p}(s) f_i(s) = 0 \quad (8)$$

where $f^{\#}(s) = \sum_{i=1}^{m} f_i(s)$. In each iteration of the IIS, we have to find the value of the improvement δ_i in the parameters, solving (8) with respect to δ_i for each $i = 1 \ldots, m$.

The main obstacle in the WSME training process resides in the calculation of the first sum in (8). The sum extends over all the sentences w of a given length. The great number of such sentences makes it impossible, from computing perspective, to calculate the sum, even for a moderate length[3]. Nevertheless, such a sum is the statistical expected value of a function of w with respect to the distribution p: $E_p\left[f_i e^{\delta_i f^{\#}}\right]$. As is well known, it could be estimated using the sampling expectation as:

$$E_p\left[f_i e^{\delta_i f^{\#}}\right] \approx \frac{1}{M} \sum_{j=1}^{M} f_i(s_j) \beta_i^{f^{\#}(s_j)} \quad (9)$$

where $s_1 \ldots, s_M$ is a random sample from p and $\beta_i = e^{\delta_i}$.

Note that in (7) the constant Z is unknown, so direct sampling from p is not possible. In sampling from such types of probability distributions, the Monte Carlo Markov Chain (MCMC) sampling methods have been successfully used when the distribition is not totally known (Neal, 1993). MCMC are based on the convergence of certain Markov Chains to a target distribution p. In MCMC, a path of the Markov chain is ran for a long time, after which the visited states are considered as a sampling element. The MCMC sampling methods have been used in the parameter estimation of the WSME language models, specially the Independence Metropolis-Hasting (IMH) and the Gibb's sampling algorithms (Chen and Rosenfeld, 1999a; Rosenfeld, 1997). The best results have been obtainded using the (IMH) algorithm.

Although MCMC performs well, the distribution from which the sample is obtained is only an *approximation* of the target sampling distribution. Therefore samples obtained from such distributions may produce some bias in sample statistics, like sampling mean. Recently, another sampling technique which is also based on Markov Chains has been developed by Propp and Wilson (Propp and Wilson, 1996), the Perfect Sampling (PS) technique. PS is based on the concept of *Coupling From the Past*. In PS, several paths of the Markov chain are running from the past (one path in each state of the chain). In all the paths, the transition rule of the Markov chain uses the same set of random numbers to transit from one state to another. Thus if two paths coincide in the same state in time t, they will remain in the same states the rest of the time. In such a case, we say that the two paths are collapsed.

Now, if all the paths collapse at any given time, from that point in time, we are sure that we are sampling from the true target distribution p. The Coupling From the Past algorithm, systematically goes to the past and then runs paths in all states and repeats this procedure until a time T has been found. Once T has been found, the paths that begin in time $-T$ all paths collapse at time $t = 0$. Then we run a path of the chain from the state at time $t = -T$ to the actual time ($t = 0$), and the last state arrived is a sample from the target distribution. The reason for going from past to current time is technical, and is detailed in (Propp and Wilson, 1996). If the state space is huge (as is the case where the state space is the set of all sentences), we must define a stochastic order over

[3] the number of sentences s of length l is $|\mathcal{W}|^l$

the state space and then run only two paths: one beginning in the minimum state and the other in the maximum state, following the same mechanism described above for the two paths until they collapse. In this way, it is proved that we get a sample from the *exact* target distribution and not from an *approximate* distribution as in MCMC algorithms (Propp and Wilson, 1996). Thus, we hope that in samples generated with perfect sampling, statistical parameter estimators may be less biased than those generated with MCMC.

Recently (Amaya and Benedí, 2000), the PS was successfully used to estimate the parameters of a WSME language model . In that work, a comparison was made between the performance of WSME models trained using MCMC and WSME models trained using PS. Features of n-grams and features of triggers were used In both kinds of models, and the WSME model trained with PS had better performance. We then considered it appropriate to use PS in the training procedure of the WSME.

The model parameters were completed with the estimation of the global normalization constant Z. Using (7), we can deduce that $Z = E_{p_0}[R(s)]$ and thus estimate Z using the sampling expectation.

$$E_{p_0}[R(s)] \approx \frac{1}{M} \sum_{j=1}^{M} R(s_j)$$

where $s_1, \ldots, s_M$ is a random sample from p_0. Because we have total control over the distribition p_0, is easy to sample from it in the traditional way.

3 The grammatical features

The main goal of this paper is the incorporation of gramatical features to the WSME. Grammatical information may be helpful in many aplications of computational linguistics. The grammatical structure of the sentence provides long-distance information to the model, thereby complementing the information provided by other sources and improving the performance of the model. Grammatical features give a better weight to such parameters in grammatically correct sentences than in grammatically incorrect sentences, thereby helping the model to assign better probabilities to correct sentences from the language of the application. To capture the grammatical information, we use Stochastic Context-Free Grammars (SCFG).

Over the last decade, there has been an increasing interest in Stochastic Context-Free Grammars (SCFGs) for use in different tasks (K., 1979; Jelinek, 1991; Ney, 1992; Sakakibara, 1990). The reason for this can be found in the capability of SCFGs to model the long-term dependencies established between the different lexical units of a sentence, and the possibility to incorporate the stochastic information that allows for an adequate modeling of the variability phenomena. Thus, SCFGs have been successfully used on limited-domain tasks of low perplexity. However, SCFGs work poorly for large vocabulary, general-purpose tasks, because the parameter learning and the computation of word transition probabilities present serious problems for complex real tasks.

To capture the long-term relations and to solve the main problem derived from the use of SCFGs in large-vocabulary complex tasks,we consider the proposal in (Benedí and Sánchez, 2000): define a category-based SCFG and a probabilistic model of word distribution in the categories. The use of categories as terminal of the grammar reduces the number of rules to take into account and thus, the time complexity of the SCFG learning procedure. The use of the probabilistic model of word distribution in the categories, allows us to obtain the best derivation of the sentences in the application.

Actually, we have to solve two problems: the estimation of the parameters of the models and their integration to obtain the best derivation of a sentence.

The parameters of the two models are estimated from a training sample. Each word in the training sample has a part-of-speech tag (POStag) associated to it. These POStags are considered as word categories and are the terminal symbols of our SCFG.

Given a category, the probability distribution of a word is estimated by means of the relative frequency of the word in the category, i.e. the relative frequency which the word w has been labeled with a POStag (a word w may belong to different categories).

To estimate the SCFG parameters, several algorithms have been presented (K. and S.J., 1991;

Pereira and Shabes, 1992; Amaya et al., 1999; Sánchez and Benedí, 1999). Taking into account the good results achieved on real tasks (Sánchez and Benedí, 1999), we used them to learn our category-based SCFG.

To solve the integration problem, we used an algorithm that computes the probability of the best derivation that generates a sentence, given the category-based grammar and the model of word distribution into categories (Benedí and Sánchez, 2000). This algorithm is based on the well-known Viterbi-like scheme for SCFGs.

Once the grammatical framework is defined, we are in position to make use of the information provided by the SCFG. In order to define the grammatical features, we first introduce some notation.

A *Context-Free Grammar* G is a four-tuple (N, Σ, P, S), where N is the finite set of non terminals, Σ is a finite set of terminals ($N \cap \Sigma \neq \emptyset$), $S \in N$ is the initial symbol of the grammar and P is the finite set of productions or rules of the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup \Sigma)^+$. We consider only context-free grammars in *Chomsky normal form*, that is grammars with rules of the form $A \rightarrow BC$ or $A \rightarrow v$ where $A, B, C \in N$ and $v \in \Sigma$.

A *Stochastic Context-Free Gramar* G_s is a pair (G, p) where G is a context-free grammar and p is a probability distribution over the grammar rules.

The grammatical features are defined as follows: let $s = w_1 \dots w_n$, a sentence of the training set. As mentioned above, we can compute the best derivation of the sentence s, using the defined SCFG and obtain the parse tree of the sentence.

Once we have the parse tree of all the sentences in the training corpus, we can collect the set of all the production rules used in the derivation of the sentences in the corpus.

Formally: we define the set $E(s) = \{(x, y, z) \,|z \rightarrow xy\}$, where $x, y, z \in \Sigma \cup N$. $E(s)$ is the set of all grammatical rules used in the derivation of s. To include the rules of the form $A \rightarrow v$, where $A \in N$ and $v \in \Sigma$, in the set $E(s)$, we make use of a special symbol \$ which is not in the terminals nor in the non-terminals. If a rule of the form $A \rightarrow u$ occurs in the derivation tree of s, the corresponding element in $E(s)$ is written as $(A, u, \$)$. The set $E = \cup_{s \in \Omega} E(s)$ (where Ω is the corpus), is the set of grammatical features.

E is the set representation of the grammatical information contained in the derivation trees of the sentences and may be incorporated to the WSME model by means of the characteristic functions defined as:

$$f_{(x,y,z)}(s) = \begin{cases} 1 & \text{if } (x, y, z) \in E(s) \\ 0 & \text{Othewise} \end{cases} \quad (10)$$

Thus, whenever the WSME model processes a sentence s, if it is looking for a specific grammatial feature, say (a, b, c), we get the derivation tree for s and the set $E(s)$ is calculated from the derivation tree. Finally, the model asks if the the tuple (a, b, c) is an element of $E(s)$. If it is, the feature is active; if not, the feature (a, b, c) does not contribute to the sentence probability. Therefore, a sentence may be a grammatically incorrect sentence (relative to the SCFG used), if derivations with low frequency appears.

4 Experimental Work

A part of the Wall Street Journal (WSJ) which had been processed in the Penn Treebanck Project (Marcus et al., 1993) was used in the experiments. This corpus was automatically labelled and manually checked. There were two kinds of labelling: POStag labelling and syntactic labelling. The POStag vocabulary was composed of 45 labels. The syntactic labels are 14. The corpus was divided into sentences according to the bracketing.

We selected 12 sections of the corpus at random. Six were used as training corpus, three as test set and the other three sections were used as held-out for tuning the smoothing WSME model. The sets are described as follow: the training corpus has 11,201 sentences; the test set has 6,350 sentences and the held-out set has 5,796 sentences.

A base-line Katz back-off smoothed trigram model was trained using the CMU-Cambridge statistical Language Modeling Toolkit [4] and used as prior distribution in (3) i.e. p_0. The vocabulary generated by the trigram model was used as vocabulary of the WSME model. The size of the vocabulary was 19,997 words.

[4] Available at: http://svr-www.eng.cam.ac.uk/ prc14/toolkit.html

The estimation of the word-category probability distribution was computed from the training corpus. In order to avoid null values, the unseen events were labeled with a special "unknown" symbol which did not appear in the vocabulary, so that the probabilitie of the unseen envent were positive for all the categories.

The SCFG had the maximum number of rules which can be composed of 45 terminal symbols (the number of POStags) and 14 non-terminal symbols (the number of syntactic labels). The initial probabilities were randomly generated and three different seeds were tested. However, only one of them is here given that the results were very similar.

The size of the sample used in the ISS was estimated by means of an experimental procedure and was set at 10,000 elements. The procedure used to generate the sample made use of the "diagnosis of convergence" (Neal, 1993), a method by means of which an inicial portion of each run of the Markov chain of sufficient length is discarded. Thus, the states in the remaining portion come from the desired equilibrium distribution. In this work, a discarded portion of 3,000 elements was estabiliched. Thus in practice, we have to generate 13,000 instances of the Markov chain.

During the IIS, every sample was tagged using the grammar estimated above, and then the grammatical features were extracted, before combining them with other kinds of features. The adequate number of iterations of the IIS was established experimentally in 13.

We trained several WSME models using the Perfect Sampling algorithm in the IIS and a different set of features (including the grammatical features) for each model. The different sets of features used in the models were: n-grams (1-grams,2-grams,3-grams); triggers; n-grams and grammatical features; triggers and grammatical feautres; n-grams, triggers and grammatical features.

The n-gram features,(N), was selected by means of its frequency in the corpus. We select all the unigrams, the bigrams with frequency greater than 5 and the trigrams with frequency greater than 10, in order to mantain the proportion of each type of n-gram in the corpus.

The triggers, (T), were generated using a trigger toolkit developed by Adam Berger [5]. The triggers were selected in acordance with de mutual information. The triggers selected were those with mutual information greater than 0.0001.

Feat.	N	T	N+T
Without	143.197	145.432	129.639
With	125.912	122.023	116.42
% Improv.	12.10%	16.10%	10.2 %

Table 1: Comparison of the perplexity between models **with** grammatical features and models **without** grammatical features for WSME models over part of the WSJ corpus. N means features of n-grams, T means features of Triggers. The perplexity of the trained n-gram model was PP=162.049

The grammatical features, (G), were selected using the parser tree of all the sentences in the training corpus to obtain the sets $E(w)$ and their union E as defined in section 3.

The size of the initial set of features was: 12,023 n-grams, 39,428 triggers and 258 gramatical features, in total 51,709 features. At the end of the training procedure, the number of active features was significantly reduced to 4,000 features on average.

During the training procedure, some of the $\mu_i \approx 0$ and, so, we smooth the model. We smoothed it using a gaussian prior technique. In the gaussian technique, we assumed that the μ_i paramters had a gaussian (normal) prior probability distribution (Chen and Rosenfeld, 1999b) and found the maximum aposteriori parameter distribution. The prior distribution was $\mu_i \sim N(0, \sigma_i^2)$, and we used the held-out data to find the σ_i^2 parameters.

Table 1 shows the experimental results: the first row represents the set of features used. The second row shows the perplexity of the models without using grammatical features. The third row shows the perplexity of the models using grammatical features and the fourth row shows the improvement in perplexity of each model using grammatical features over the corresponding model without grammatical features. As can be seen in Table 1, all the WSME models performed

[5] Available at: htpp://www.cs.cmu.edu/afs/cs/user/aberger/www/

better than the n-gram model, however that is natural because, in the worst case (if all $\mu_i = 1$), the WSME models perform like the n-gram model.

In Table 1, we see that all the models using grammatical features perform better than the models that do not use it. Since the training procedure was the same for all the models described and since the only difference between the two kinds of models compared were the grammatical features, then we conclude that the improvement must be due to the inclusion of such features into the set of features. The average percentage of improvement was about 13%.

Also, although the model N+T performs better than the other model without grammatical features (N,T), it behaves worse than all the models with grammatical features (N+G improved 2.9% and T+G improvd 5.9% over N+T).

5 Conclusions and future work

In this work, we have sucessfully added grammatical features to a WSME language model using a SCFG to extract the grammatical information. We have shown that the the use of grammatical features in a WSME model improves the performance of the model. Adding grammatical features to the WSME model we have obtained a reduction in perplexity of 13% on average over models that do not use grammatical features. Also a reduction in perplexity between approximately 22% and 28% over the n-gram model has been obtained.

We are working on the implementation of other kinds of grammatical features which are based on the POStags sentences obtained using the SCFG that we have defined. The prelimary experiments have shown promising results.

We will also be working on the evaluation of the word-error rate (WER) of the WSME model. In the case of WSME model the WER may be evaluated in a type of post-procesing using the n-best utterances.

References

F. Amaya and J. M. Benedí. 2000. Using Perfect Sampling in Parameter Estimation of a Wole Sentence Maximum Entropy Language Model. *Proc. Fourth Computational Natural Language Learning Workshop, CoNLL-2000*.

F. Amaya, J. A. Sánchez, and J. M. Benedí. 1999. Learning stochastic context-free grammars from bracketed corpora by means of reestimation algorithms. *Proc. VIII Spanish Symposium on Pattern Recognition and Image Analysis*, pages 119–126.

L.R. Bahal, F.Jelinek, and R. L. Mercer. 1983. A maximun likelihood approach to continuous speech recognition. *IEEE Trans. on Pattern analysis and Machine Intelligence*, 5(2):179–190.

J. R. Bellegarda. 1998. A multispan language modeling framework for large vocabulary speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6 (5):456–467.

J.M. Benedí and J.A. Sánchez. 2000. Combination of n-grams and stochastic context-free grammars for language modeling. *Porc. International conference on computational lingustics (COLING-ACL)*, pages 55–61.

A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. 1996. A Maximun Entropy aproach to natural languaje processing. *Computational Linguistics*, 22(1):39–72.

A. Borthwick. 1997. Survey paper on statistical language modeling. Technical report, New York University.

A. Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. PhD Dissertation Proposal, New York University.

C. Chelba and F. Jelinek. 2000. Structured language modeling. *Computer Speech and Language*, 14:283–332.

S. Chen and R. Rosenfeld. 1999a. Efficient sampling and feature selection in whole sentence maximum entropy language models. *Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

S. Chen and R. Rosenfeld. 1999b. A gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University.

S. Della Pietra, V. Della Pietra, and J. Lafferty. 1995. Inducing features of random fields. Technical Report CMU-CS-95-144, Carnegie Mellon University.

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. 1991. A dynamic language model for speech recognition. *Proc. of Speech and Natural Language DARPA Work Shop*, pages 293–295.

F. Jelinek. 1991. Up from trigrams! the struggle for improved language models. *Proc. of EUROSPEECH, European Conference on Speech Comunication and Technology*, 3:1034–1040.

F. Jelinek. 1997. *Statistical Methods for Speech Recognition*. The MIT Press, Massachusetts Institut of Technology. Cambridge, Massachusetts.

Lari K. and Young S.J. 1991. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, pages 237–257.

Baker J. K. 1979. Trainable grammars for speech recognition. *Speech comunications papers for the 97th meeting of the Acoustical Society of America*, pages 547–550.

M. P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotates corpus of english: the penn treebanck. *Computational Linguistics*, 19.

R. M. Neal. 1993. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Departament of Computer Science, University of Toronto.

H. Ney. 1992. Stochastic grammars and pattern recognition. In P. Laface and R. De Mori, editors, *Speech Recognition and Understanding. Recent Advances*, pages 319–344. Springer Verlag.

F. Pereira and Y. Shabes. 1992. Inside-outsude reestimation from partially bracketed corpora. *Proceedings of the 30th Annual Meeting of the Assotiation for Computational Linguistics*, pages 128–135. University of Delaware.

J. G. Propp and D. B. Wilson. 1996. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252.

A. Ratnapharkhi. 1998. *Maximum Entropy models for natural language ambiguity resolution*. PhD Dissertation Proposal, University of Pensylvania.

R. Rosenfeld. 1996. A Maximun Entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228.

R. Rosenfeld. 1997. A whole sentence Maximim Entropy language model. *IEEE workshop on Speech Recognition and Understanding*.

Y. Sakakibara. 1990. Learning context-free grammars from structural data in polinomila time. *Theoretical Computer Science*, 76:233–242.

J. A. Sánchez and J. M. Benedí. 1999. Learning of stochastic context-free grammars by means of estimation algorithms. *Proc. of EUROSPEECH, European Conference on Speech Comunication and Technology*, 4:1799–1802.