Constituent Parsing with Incremental Sigmoid Belief Networks

Ivan Titov
Department of Computer Science
University of Geneva
24, rue Général Dufour
CH-1211 Genève 4, Switzerland
ivan.titov@cui.unige.ch

James Henderson
School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW, United Kingdom
james.henderson@ed.ac.uk

Abstract

We introduce a framework for syntactic parsing with latent variables based on a form of dynamic Sigmoid Belief Networks called Incremental Sigmoid Belief Networks. We demonstrate that a previous feed-forward neural network parsing model can be viewed as a coarse approximation to inference with this class of graphical model. By constructing a more accurate but still tractable approximation, we significantly improve parsing accuracy, suggesting that ISBNs provide a good idealization for parsing. This generative model of parsing achieves state-of-the-art results on WSJ text and 8% error reduction over the baseline neural network parser.

1 Introduction

Latent variable models have recently been of increasing interest in Natural Language Processing, and in parsing in particular (e.g. (Koo and Collins, 2005; Matsuzaki et al., 2005; Riezler et al., 2002)). Latent variables provide a principled way to include features in a probability model without needing to have data labeled with those features in advance. Instead, a labeling with these features can be induced as part of the training process. The difficulty with latent variable models is that even small numbers of latent variables can lead to computationally intractable inference (a.k.a. decoding, parsing). In this paper we propose a solution to this problem based on dynamic Sigmoid Belief Networks (SBNs) (Neal, 1992). The dynamic SBNs which we peopose, called Incremental Sigmoid Belief Networks (ISBNs) have large numbers of latent variables, which makes exact inference intractable. However, they can be approximated sufficiently well to build fast and accurate statistical parsers which induce features during training.

We use SBNs in a generative history-based model of constituent structure parsing. The probability of an unbounded structure is decomposed into a sequence of probabilities for individual derivation decisions, each decision conditioned on the unbounded history of previous decisions. The most common approach to handling the unbounded nature of the histories is to choose a pre-defined set of features which can be unambiguously derived from the history (e.g. (Charniak, 2000; Collins, 1999)). Decision probabilities are then assumed to be independent of all information not represented by this finite set of features. Another previous approach is to use neural networks to compute a compressed representation of the history and condition decisions on this representation (Henderson, 2003; Henderson, 2004). It is possible that an unbounded amount of information is encoded in the compressed representation via its continuous values, but it is not clear whether this is actually happening due to the lack of any principled interpretation for these continuous values.

Like the former approach, we assume that there are a finite set of features which encode the relevant information about the parse history. But unlike that approach, we allow feature values to be ambiguous, and represent each feature as a distribution over (binary) values. In other words, these history features are treated as latent variables. Unfortunately, inter-

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 632–639,
Prague, Czech Republic, June 2007.

preting the history representations as distributions over discrete values of latent variables makes the exact computation of decision probabilities intractable. Exact computation requires marginalizing out the latent variables, which involves summing over all possible vectors of discrete values, which is exponential in the length of the vector.

We propose two forms of approximation for dynamic SBNs, a neural network approximation and a form of mean field approximation (Saul and Jordan, 1999). We first show that the previous neural network model of (Henderson, 2003) can be viewed as a coarse approximation to inference with ISBNs. We then propose an incremental mean field method, which results in an improved approximation over the neural network but remains tractable. The resulting parser achieves significantly higher accuracy than the neural network parser (90.0% F-measure vs 89.1%). We argue that this correlation between better approximation and better accuracy suggests that dynamic SBNs are a good abstract model for natural language parsing.

2 Sigmoid Belief Networks

A belief network, or a Bayesian network, is a directed acyclic graph which encodes statistical dependencies between variables. Each variable S_i in the graph has an associated conditional probability distributions $P(S_i|Par(S_i))$ over its values given the values of its parents $Par(S_i)$ in the graph. A Sigmoid Belief Network (Neal, 1992) is a particular type of belief networks with binary variables and conditional probability distributions in the form of the logistic sigmoid function:

$$P(S_i{=}1|Par(S_i)) = \frac{1}{1{+}\exp(-\sum_{S_j \in Par(S_i)} J_{ij} S_j)},$$

where J_{ij} is the weight for the edge from variable S_j to variable S_i. In this paper we consider a generalized version of SBNs where we allow variables with any range of discrete values. We thus generalize the logistic sigmoid function to the normalized exponential (a.k.a. softmax) function to define the conditional probabilities for non-binary variables.

Exact inference with all but very small SBNs is not tractable. Initially sampling methods were used (Neal, 1992), but this is also not feasible for large networks, especially for the dynamic models of the type described in section 2.2. Variational methods have also been proposed for approximating SBNs (Saul and Jordan, 1999). The main idea of variational methods (Jordan et al., 1999) is, roughly, to construct a tractable approximate model with a number of free parameters. The free parameters are set so that the resulting approximate model is as close as possible to the original graphical model for a given inference problem.

2.1 Mean Field Approximation Methods

The simplest example of a variation method is the mean field method, originally introduced in statistical mechanics and later applied to unsupervised neural networks in (Hinton et al., 1995). Let us denote the set of visible variables in the model (i.e. the inputs and outputs) by V and hidden variables by $H = h_1, \ldots, h_l$. The mean field method uses a fully factorized distribution Q as the approximate model:

$$Q(H|V) = \prod_i Q_i(h_i|V).$$

where each Q_i is the distribution of an individual latent variable. The independence between the variables h_i in this approximate distribution Q does not imply independence of the free parameters which define the Q_i. These parameters are set to minimize the Kullback-Leibler divergence (Cover and Thomas, 1991) between the approximate distribution $Q(H|V)$ and the true distribution $P(H|V)$:

$$KL(Q\|P) = \sum_H Q(H|V) \ln \frac{Q(H|V)}{P(H|V)}, \tag{1}$$

or, equivalently, to maximize the expression:

$$L_V = \sum_H Q(H|V) \ln \frac{P(H,V)}{Q(H|V)}. \tag{2}$$

The expression L_V is a lower bound on the log-likelihood $\ln P(V)$. It is used in the mean field theory (Saul and Jordan, 1999) to approximate the likelihood. However, in our case of dynamic graphical models, we have to use a different approach which allows us to construct an incremental parsing method without needing to introduce the additional parameters proposed in (Saul and Jordan, 1999). We will describe our modification of the mean field method in section 3.3.

2.2 Dynamics

Dynamic Bayesian networks are Bayesian networks applied to arbitrarily long sequences. A new set of variables is instantiated for each position in the sequence, but the edges and weights for these variables are the same as in other positions. The edges which connect variables instantiated for different positions must be directed forward in the sequence, thereby allowing a temporal interpretation of the sequence. Typically a dynamic Bayesian Network will only involve edges between adjacent positions in the sequence (i.e. they are Markovian), but in our parsing models the pattern of interconnection is determined by structural locality, rather than sequence locality, as in the neural networks of (Henderson, 2003).

Using structural locality to define the graph in a dynamic SBN means that the subgraph of edges with destinations at a given position cannot be determined until all the parser decisions for previous positions have been chosen. We therefore call these models *Incremental* SBNs, because, at any given position in the parse, we only know the graph of edges for that position and previous positions in the parse. For example in figure 1, discussed below, it would not be possible to draw the portion of the graph after t, because we do not yet know the decision d_k^t.

The incremental specification of model structure means that we cannot use an undirected graphical model, such as Conditional Random Fields. With a directed dynamic model, all edges connecting the known portion of the graph to the unknown portion of the graph are directed toward the unknown portion. Also there are no variables in the unknown portion of the graph whose values are known (i.e. no visible variables), because at each step in a history-based model the decision probability is conditioned only on the parsing history. Only visible variables can result in information being reflected backward through a directed edge, so it is impossible for anything in the unknown portion of the graph to affect the probabilities in the known portion of the graph. Therefore inference can be performed by simply ignoring the unknown portion of the graph, and there is no need to sum over all possible structures for the unknown portion of the graph, as would be necessary for an undirected graphical model.

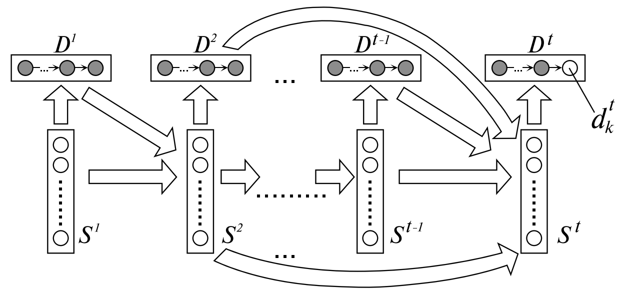

Figure 1: Illustration of an ISBN.

3 The Probabilistic Model of Parsing

In this section we present our framework for syntactic parsing with dynamic Sigmoid Belief Networks. We first specify the form of SBN we propose, namely ISBNs, and then two methods for approximating the inference problems required for parsing. We only consider generative models of parsing, since generative probability models are simpler and we are focused on probability estimation, not decision making. Although the most accurate parsing models (Charniak and Johnson, 2005; Henderson, 2004; Collins, 2000) are discriminative, all the most accurate discriminative models make use of a generative model. More accurate generative models should make the discriminative models which use them more accurate as well. Also, there are some applications, such as language modeling, which require generative models.

3.1 The Graphical Model

In ISBNs, we use a history-based model, which decomposes the probability of the parse as:

$$P(T) = P(D^1, ..., D^m) = \prod_t P(D^t | D^1, \ldots, D^{t-1}),$$

where T is the parse tree and $D^1, \ldots, D^m$ is its equivalent sequence of parser decisions. Instead of treating each D^t as atomic decisions, it is convenient to further split them into a sequence of elementary decisions $D^t = d_1^t, \ldots, d_n^t$:

$$P(D^t | D^1, \ldots, D^{t-1}) = \prod_k P(d_k^t | h(t,k)),$$

where $h(t,k)$ denotes the parsing history $D^1, \ldots, D^{t-1}, d_1^t, \ldots, d_{k-1}^t$. For example, a decision to create a new constituent can be divided in two elementary decisions: deciding to create a constituent and deciding which label to assign to it. We use a graphical model to define our proposed class of probability models. An example graphical model for the computation of $P(d_k^t | h(t,k))$ is illustrated in figure 1.

The graphical model is organized into vectors of variables: latent state variable vectors $S^{t'} = s_1^{t'}, \ldots, s_n^{t'}$, representing an intermediate state of the parser at derivation step t', and decision variable vectors $D^{t'} = d_1^{t'}, \ldots, d_l^{t'}$, representing a parser decision at derivation step t', where $t' \leq t$. Variables whose value are given at the current decision (t,k) are shaded in figure 1, latent and output variables are left unshaded.

As illustrated by the arrows in figure 1, the probability of each state variable $s_i^{t'}$ depends on all the variables in a finite set of relevant previous state and decision vectors, but there are no *direct* dependencies between the different variables in a single state vector. Which previous state and decision vectors are connected to the current state vector is determined by a set of structural relations specified by the parser designer. For example, we could select the most recent state where the same constituent was on the top of the stack, and a decision variable representing the constituent's label. Each such selected relation has its own distinct weight matrix for the resulting edges in the graph, but the same weight matrix is used at each derivation position where the relation is relevant.

As indicated in figure 1, the probability of each elementary decision $d_k^{t'}$ depends both on the current state vector $S^{t'}$ and on the previously chosen elementary action $d_{k-1}^{t'}$ from $D^{t'}$. This probability distribution has the form of a normalized exponential:

$$P(d_k^{t'} = d | S^{t'}, d_{k-1}^{t'}) = \frac{\Phi_{h(t',k)}(d)\, e^{\sum_j W_{dj} s_j^{t'}}}{\sum_{d'} \Phi_{h(t',k)}(d')\, e^{\sum_j W_{d'j} s_j^{t'}}}, \quad (3)$$

where $\Phi_{h(t',k)}$ is the indicator function of a set of elementary decisions that may possibly follow the parsing history $h(t',k)$, and the W_{dj} are the weights.

For our experiments, we replicated the same pattern of interconnection between state variables as described in (Henderson, 2003).[1] We also used the same left-corner parsing strategy, and the same set of decisions, features, and states. We refer the reader to (Henderson, 2003) for details.

Exact computation with this model is not tractable. Sampling of parse trees from the model is not feasible, because a generative model defines a joint model of both a sentence and a tree, thereby requiring sampling over the space of sentences. Gibbs sampling (Geman and Geman, 1984) is also impossible, because of the huge space of variables and need to resample after making each new decision in the sequence. Thus, we know of no reasonable alternatives to the use of variational methods.

3.2 A Feed-Forward Approximation

The first model we consider is a strictly incremental computation of a variational approximation, which we will call the feed-forward approximation. It can be viewed as the simplest form of mean field approximation. As in any mean field approximation, each of the latent variables is independently distributed. But unlike the general case of mean field approximation, in the feed-forward approximation we only allow the parameters of the distributions Q_i to depend on the distributions of their parents. This additional constraint increases the potential for a large Kullback-Leibler divergence with the true model, defined in expression (1), but it significantly simplifies the computations.

The set of hidden variables H in our graphical model consists of all the state vectors $S^{t'}$, $t' \leq t$, and the last decision d_k^t. All the previously observed decisions $h(t,k)$ comprise the set of visible variables V. The approximate fully factorisable distribution $Q(H|V)$ can be written as:

$$Q(H|V) = q_k^t(d_k^t) \prod_{t',i} \left(\mu_i^{t'}\right)^{s_i^{t'}} \left(1 - \mu_i^{t'}\right)^{1 - s_i^{t'}}.$$

where $\mu_i^{t'}$ is the free parameter which determines the distribution of state variable i at position t', namely its mean, and $q_k^t(d_k^t)$ is the free parameter which determines the distribution over decisions d_k^t.

Because we are only allowed to use information about the distributions of the parent variables to

[1] In the neural network of (Henderson, 2003), our variables map to their "units", and our dependencies/edges map to their "links".

compute the free parameters $\mu_i^{t'}$, the optimal assignment of values to the $\mu_i^{t'}$ is:

$$\mu_i^{t'} = \sigma\left(\eta_i^{t'}\right),$$

where σ denotes the logistic sigmoid function and $\eta_i^{t'}$ is a weighted sum of the parent variables' means:

$$\eta_i^{t'} = \sum_{t'' \in RS(t')} \sum_j J_{ij}^{\tau(t',t'')} \mu_j^{t''} + \sum_{t'' \in RD(t')} \sum_k B_{id_k^{t''}}^{\tau(t',t'')}, \quad (4)$$

where $RS(t')$ is the set of previous positions with edges from their state vectors to the state vector at t', $RD(t')$ is the set of previous positions with edges from their decision vectors to the state vector at t', $\tau(t', t'')$ is the relevant relation between the position t'' and the position t', and J_{ij}^{τ} and B_{id}^{τ} are weight matrices.

In order to maximize (2), the approximate distribution of the next decisions $q_k^t(d)$ should be set to

$$q_k^t(d) = \frac{\Phi_{h(t,k)}(d)\, e^{\sum_j W_{dj}\mu_j^t}}{\sum_{d'} \Phi_{h(t,k)}(d')\, e^{\sum_j W_{d'j}\mu_j^t}}, \quad (5)$$

as follows from expression (3). The resulting estimate of the tree probability is given by:

$$P(T) \approx \prod_{t,k} q_k^t(d_k^t).$$

This approximation method replicates exactly the computation of the feed-forward neural network in (Henderson, 2003), where the above means $\mu_i^{t'}$ are equivalent to the neural network hidden unit activations. Thus, that neural network probability model can be regarded as a simple approximation to the graphical model introduced in section 3.1.

In addition to the drawbacks shared by any mean field approximation method, this feed-forward approximation cannot capture backward reasoning. By backward (a.k.a. top-down) reasoning we mean the need to update the state vector means $\mu_i^{t'}$ after observing a decision d_k^t, for $t' \leq t$. The next section discusses how backward reasoning can be incorporated in the approximate model.

3.3 A Mean Field Approximation

This section proposes a more accurate way to approximate ISBNs with mean field methods, which we will call the mean field approximation. Again, we are interested in finding the distribution Q which maximizes the quantity L_V in expression (2). The decision distribution $q_k^t(d_k^t)$ maximizes L_V when it has the same dependence on the state vector means μ_k^t as in the feed-forward approximation, namely expression (5). However, as we mentioned above, the feed-forward computation does not allow us to compute the optimal values of state means $\mu_i^{t'}$.

Optimally, after each new decision d_k^t, we should recompute all the means $\mu_i^{t'}$ for all the state vectors $S^{t'}$, $t' \leq t$. However, this would make the method intractable, due to the length of derivations in constituent parsing and the interdependence between these means. Instead, after making each decision d_k^t and adding it to the set of visible variables V, we recompute only means of the current state vector S^t.

The denominator of the normalized exponential function in (3) does not allow us to compute L_V exactly. Instead, we use a simple first order approximation:

$$E_Q[\ln \sum_d \Phi_{h(t,k)}(d) \exp(\sum_j W_{dj} s_j^t)]$$
$$\approx \ln \sum_d \Phi_{h(t,k)}(d) \exp(\sum_j W_{dj} \mu_j^t), \quad (6)$$

where the expectation $E_Q[\ldots]$ is taken over the state vector S^t distributed according to the approximate distribution Q.

Unfortunately, even with this assumption there is no analytic way to maximize L_V with respect to the means μ_k^t, so we need to use numerical methods. Assuming (6), we can rewrite the expression (2) as follows, substituting the true $P(H,V)$ defined by the graphical model and the approximate distribution $Q(H|V)$, omitting parts independent of μ_k^t:

$$\begin{aligned} L_V^{t,k} &= \sum_i -\mu_i^t \ln \mu_i^t - (1-\mu_i^t)\ln\left(1-\mu_i^t\right) \\ &\quad + \mu_i^t \eta_i^t + \sum_{k'<k} \Phi_{h(t,k')}(d_{k'}^t) \sum_j W_{d_{k'}^t j} \mu_j^t \\ &\quad - \sum_{k'<k} \ln\left(\sum_d \Phi_{h(t,k')}(d) \exp(\sum_j W_{dj}\mu_j^t)\right), \quad (7) \end{aligned}$$

here, η_i^t is computed from the previous relevant state means and decisions as in (4). This expression is concave with respect to the parameters μ_i^t, so the global maximum can be found. We use coordinate-wise ascent, where each μ_i^t is selected by an efficient line search (Press et al., 1996), while keeping other $\mu_{i'}^t$ fixed.

3.4 Parameter Estimation

We train these models to maximize the fit of the *approximate* model to the data. We use gradient descent and a maximum likelihood objective function. This requires computation of the gradient of the approximate log-likelihood with respect to the model parameters. In order to compute these derivatives, the error should be propagated all the way back through the structure of the graphical model. For the feed-forward approximation, computation of the derivatives is straightforward, as in neural networks. But for the mean field approximation, it requires computation of the derivatives of the means μ_i^t with respect to the other parameters in expression (7). The use of a numerical search in the mean field approximation makes the analytical computation of these derivatives impossible, so a different method needs to be used to compute their values. If maximization of $L_V^{t,k}$ is done until convergence, then the derivatives of $L_V^{t,k}$ with respect to μ_i^t are close to zero:

$$F_i^{t,k} = \frac{\partial L_V^{t,k}}{\partial \mu_i^t} \approx 0 \text{ for all } i.$$

This system of equations allows us to use implicit differentiation to compute the needed derivatives.

4 Experimental Evaluation

In this section we evaluate the two approximations to dynamic SBNs discussed in the previous section, the feed-forward method equivalent to the neural network of (Henderson, 2003) (NN method) and the mean field method (MF method). The hypothesis we wish to test is that the more accurate approximation of dynamic SBNs will result in a more accurate model of constituent structure parsing. If this is true, then it suggests that dynamic SBNs of the form proposed here are a good abstract model of the nature of natural language parsing.

We used the Penn Treebank WSJ corpus (Marcus et al., 1993) to perform the empirical evaluation of the considered approaches. It is expensive to train the MF approximation on the whole WSJ corpus, so instead we use only sentences of length at most 15, as in (Taskar et al., 2004) and (Turian and Melamed, 2006). The standard split of the corpus into training (sections 2–22, 9,753 sentences), validation (section 24, 321 sentences), and testing (section 23, 603 sentences) was performed.[2]

	R	P	F_1
Bikel, 2004	87.9	88.8	88.3
Taskar et al., 2004	89.1	89.1	89.1
NN method	**89.1**	**89.2**	**89.1**
Turian and Melamed, 2006	89.3	89.6	89.4
MF method	**89.3**	**90.7**	**90.0**
Charniak, 2000	90.0	90.2	90.1

Table 1: Percentage labeled constituent recall (R), precision (P), combination of both (F_1) on the testing set.

As in (Henderson, 2003; Turian and Melamed, 2006) we used a publicly available tagger (Ratnaparkhi, 1996) to provide the part-of-speech tag for each word in the sentence. For each tag, there is an unknown-word vocabulary item which is used for all those words which are not sufficiently frequent with that tag to be included individually in the vocabulary. We only included a specific tag-word pair in the vocabulary if it occurred at least 20 time in the training set, which (with tag-unknown-word pairs) led to the very small vocabulary of 567 tag-word pairs.

During parsing with both the NN method and the MF method, we used beam search with a post-word beam of 10. Increasing the beam size beyond this value did not significantly effect parsing accuracy. For both of the models, the state vector size of 40 was used. All the parameters for both the NN and MF models were tuned on the validation set. A single best model of each type was then applied to the final testing set.

Table 1 lists the results of the NN approximation and the MF approximation, along with results of dif-

[2]Training of our MF method on this subset of WSJ took less than 6 days on a standard desktop PC. We would expect that a model for the entire WSJ corpus can be trained in about 3 months time. The training time is about linear with the number of words, but a larger state vector is needed to accommodate all the information. The long training times on the entire WSJ would not allow us to tune the model parameters properly, which would have increased the randomness of the empirical comparison, although it would be feasible for building a system.

ferent generative and discriminative parsing methods (Bikel, 2004; Taskar et al., 2004; Turian and Melamed, 2006; Charniak, 2000) evaluated in the same experimental setup. The MF model improves over the baseline NN approximation, with an error reduction in F-measure exceeding 8%. This improvement is statically significant.[3] The MF model achieves results which do not appear to be significantly different from the results of the best model in the list (Charniak, 2000). It should also be noted that the model (Charniak, 2000) is the most accurate generative model on the standard WSJ parsing benchmark, which confirms the viability of our generative model.

These experimental results suggest that Incremental Sigmoid Belief Networks are an appropriate model for natural language parsing. Even approximations such as those tested here, with a very strong factorisability assumption, allow us to build quite accurate parsing models. The main drawback of our proposed mean field approach is the relative computational complexity of the numerical procedure used to maximize $L_V^{t,k}$. But this approximation has succeeded in showing that a more accurate approximation of ISBNs results in a more accurate parser. We believe this provides strong justification for more accurate approximations of ISBNs for parsing.

5 Related Work

There has not been much previous work on graphical models for full parsing, although recently several latent variable models for parsing have been proposed (Koo and Collins, 2005; Matsuzaki et al., 2005; Riezler et al., 2002). In (Koo and Collins, 2005), an undirected graphical model is used for parse reranking. Dependency parsing with dynamic Bayesian networks was considered in (Peshkin and Savova, 2005), with limited success. Their model is very different from ours. Roughly, it considered the whole sentence at a time, with the graphical model being used to decide which words correspond to leaves of the tree. The chosen words are then removed from the sentence and the model is recursively applied to the reduced sentence.

Undirected graphical models, in particular Conditional Random Fields, are the standard tools for shallow parsing (Sha and Pereira, 2003). However, shallow parsing is effectively a sequence labeling problem and therefore differs significantly from full parsing. As discussed in section 2.2, undirected graphical models do not seem to be suitable for history-based full parsing models.

Sigmoid Belief Networks were used originally for character recognition tasks, but later a dynamic modification of this model was applied to the reinforcement learning task (Sallans, 2002). However, their graphical model, approximation method, and learning method differ significantly from those of this paper.

6 Conclusions

This paper proposes a new generative framework for constituent parsing based on dynamic Sigmoid Belief Networks with vectors of latent variables. Exact inference with the proposed graphical model (called Incremental Sigmoid Belief Networks) is not tractable, but two approximations are considered. First, it is shown that the neural network parser of (Henderson, 2003) can be considered as a simple feed-forward approximation to the graphical model. Second, a more accurate but still tractable approximation based on mean field theory is proposed. Both methods are empirically compared, and the mean field approach achieves significantly better results, which are non-significantly different from the results of the most accurate generative parsing model (Charniak, 2000) on our testing set. The fact that a more accurate approximation leads to a more accurate parser suggests that ISBNs are a good abstract model for constituent structure parsing. This empirical result motivates research into more accurate approximations of dynamic SBNs.

We focused in this paper on generative models of parsing. The results of such a generative model can be easily improved by a discriminative reranking model, even without any additional feature engineering. For example, the discriminative training techniques successfully applied in (Henderson, 2004) to the feed-forward neural network model can be directly applied to the mean field model proposed in this paper. The same is true for reranking with data-defined kernels, with which we would

[3] We measured significance of all the experiments in this paper with the randomized significance test (Yeh, 2000).

expect similar improvements as were achieved with the neural network parser (Henderson and Titov, 2005). Such improvements should situate the resulting model among the best current parsing models.

References

Dan M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4).

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. ACL*, pages 173–180, Ann Arbor, MI.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. ACL*, pages 132–139, Seattle, Washington.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. ICML*, pages 175–182, Stanford, CA.

Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. John Wiley, New York, NY.

S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

James Henderson and Ivan Titov. 2005. Data-defined kernels for parse reranking derived from probabilistic models. In *Proc. ACL*, Ann Arbor, MI.

James Henderson. 2003. Inducing history representations for broad coverage statistical parsing. In *Proc. HLT-NAACL*, pages 103–110, Edmonton, Canada.

James Henderson. 2004. Discriminative training of a neural network statistical parser. In *Proc. ACL*, Barcelona, Spain.

G. Hinton, P. Dayan, B. Frey, and R. Neal. 1995. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161.

M. I. Jordan, Z.Ghahramani, T. S. Jaakkola, and L. K. Saul. 1999. An introduction to variational methods for graphical models. In Michael I. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA.

Terry Koo and Michael Collins. 2005. Hidden-variable models for discriminative reranking. In *Proc. EMNLP*, Vancouver, B.C., Canada.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proc. ACL*, Ann Arbor, MI.

Radford Neal. 1992. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113.

Leon Peshkin and Virginia Savova. 2005. Dependency parsing with dynamic bayesian network. In *AAAI, 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania.

W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. 1996. *Numerical Recipes*. Cambridge University Press, Cambridge, UK.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. EMNLP*, pages 133–142, Univ. of Pennsylvania, PA.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proc. ACL*, Philadelphia, PA.

Brian Sallans. 2002. *Reinforcement Learning for Factored Markov Decision Processes*. Ph.D. thesis, University of Toronto, Toronto, Canada.

Lawrence K. Saul and Michael I. Jordan. 1999. A mean field learning algorithm for unsupervised neural networks. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 541–554. MIT Press, Cambridge, MA.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. HLT-NAACL*, Edmonton, Canada.

Ben Taskar, Dan Klein, Michael Collins, Daphne Koller, and Christopher Manning. 2004. Max-margin parsing. In *Proc. EMNLP*, Barcelona, Spain.

Joseph Turian and Dan Melamed. 2006. Advances in discriminative parsing. In *Proc. COLING-ACL*, Sydney, Australia.

Alexander Yeh. 2000. More accurate tests for the statistical significance of the result differences. In *Proc. COLING*, pages 947–953, Saarbruken, Germany.