A Latent Variable Model for Generative Dependency Parsing

Ivan Titov and James Henderson

1 Introduction

Dependency parsing has been a topic of active research in natural language processing during the last several years. The CoNLL-2006 shared task [7] made a wide selection of standardized treebanks for different languages available for the research community and allowed for easy comparison between various statistical methods on a standardized benchmark. One of the surprising things discovered by this evaluation is that the best results are achieved by methods which are quite different from state-of-the-art models for constituent parsing, e.g. the deterministic parsing method of [36] and the minimum spanning tree parser of [29]. All the most accurate dependency parsing models are fully discriminative, unlike constituent parsing where all the state-of-the-art methods have a generative component [9, 18, 12]. Another surprising thing is the lack of latent variable models among the methods used in the shared task. Latent variable models would allow complex features to be induced automatically, which would be highly desirable in multilingual parsing, where manual feature selection might be very difficult and time consuming, especially for languages unknown to the parser developer.

In this paper we propose a generative latent variable model for dependency parsing. It is based on Incremental Sigmoid Belief Networks (ISBNs), a class of directed graphical model for structure prediction problems recently proposed in [46], where they were demonstrated to achieve competitive results on the constituent parsing task. As discussed in [46], computing the conditional probabilities which we need for parsing is in general intractable with ISBNs, but they can be approximated efficiently in several ways. In particular, the neural network constituent parsers in [17]

Ivan Titov
Dept Computer Science, Univ Illinois at Urbana-Champaign, e-mail: titov@uiuc.edu

James Henderson
Dept Computer Science, Univ Geneva, e-mail: james.henderson@unige.ch

and [18] can be regarded as coarse approximations to their corresponding ISBN model.

ISBNs use history-based probability models. The most common approach to handling the unbounded nature of the parse histories in these models is to choose a pre-defined set of features which can be unambiguously derived from the history (e.g. [8, 11, 35]). Decision probabilities are then assumed to be independent of all information not represented by this finite set of features. ISBNs instead use a vector of binary latent variables to encode the information about the parser history. This history vector is similar to the hidden state of a Hidden Markov Model. But unlike the graphical model for an HMM, which specifies conditional dependency edges only between adjacent states in the sequence, the ISBN graphical model can specify conditional dependency edges between states which are arbitrarily far apart in the parse history. The source state of such an edge is determined by the partial output structure built at the time of the destination state, thereby allowing the conditional dependency edges to be appropriate for the structural nature of the problem being modeled. This structure sensitivity is possible because ISBNs are a constrained form of switching model [31], where each output decision switches the model structure used for the remaining decisions.

We build an ISBN model of dependency parsing using the parsing order proposed in [35]. However, instead of performing deterministic parsing as in [35], we use this ordering to define a generative history-based model, by integrating word prediction operations into the set of parser actions. Then we propose a simple, language independent set of relations which determine how latent variable vectors are interconnected by conditional dependency edges in the ISBN model. ISBNs also condition the latent variable vectors on a set of explicit features, which we vary in the experiments.

In experiments we evaluate both the performance of the ISBN dependency parser compared to related work, and the ability of the ISBN model to induce complex history features. Our model achieves state-of-the-art performance on the languages we test from the CoNLL-2006 shared task, significantly outperforming the model of [36] on two languages out of three and demonstrating about the same results on the third. In order to test the model's feature induction abilities, we train models with two different sets of explicit conditioning features: the feature set individually tuned by [36] for each considered language, and a minimal set of local features. These models achieve comparable accuracy, unlike with the discriminative SVM-based approach of [36], where careful feature selection appears to be crucial. We also conduct a controlled experiment where we used the tuned features of [36] but disable the feature induction abilities of our model by elimination of the edges connecting latent state vectors. This restricted model achieves far worse results, showing that it is exactly the capacity of ISBNs to induce history features which is the key to its success. It also motivates further research into how feature induction techniques can be exploited in discriminative parsing methods.

We analyze how the relation accuracy changes with the length of the head-dependent relation, demonstrating that our model very significantly outperforms the state-of-the-art baseline of [36] on long dependencies. Additional experiments suggest that both feature induction abilities and use of the beam search contribute to this improvement.

We participated with this parser in the CoNLL-2007 shared task on dependency parsing. The model achieved the third overall results, which were only 0.4% worse than results of the best participating system. It is also important to note, that it achieved the best result among single model parsers: the parsers which achieved better results [16, 33] used combinations of models. We also study parser efficiency on the datasets from the CoNLL-2007 shared task.

The fact that our model defines a probability model over parse trees, unlike the previous state-of-the-art methods [36, 29], makes it easier to use this model in applications which require probability estimates, e.g. in language processing pipelines. Also, as with any generative model, it may be easy to improve the parser's accuracy by using discriminative retraining techniques [18] or data-defined kernels [20], with or even without introduction of any additional linguistic features. In addition, there are some applications, such as language modeling, which require generative models. Another advantage of generative models is that they do not suffer from the label bias problems [6], which is a potential problem for conditional or deterministic history-based models, such as [35].

In the remainder of this paper, we will first review general ISBNs and how they can be approximated. Then we will define the generative parsing model, based on the algorithm of [35], and propose an ISBN for this model. The empirical part of the paper then evaluates both the overall accuracy of this method and the importance of the model's capacity to induce features. Additional related work will be discussed in the last section before concluding.

2 The Latent Variable Architecture

In this section we will begin by briefly introducing the class of graphical models we will be using, Incremental Sigmoid Belief Networks [46]. ISBNs are designed specifically for modeling structured data. They are latent variable models which are not tractable to compute exactly, but two approximations exist which have been shown to be effective for constituent parsing [46]. Finally, we present how these approximations can be trained.

2.1 Incremental Sigmoid Belief Networks

An ISBN is a form of Sigmoid Belief Network (SBN) [34]. SBNs are Bayesian Networks with binary variables and conditional probability distributions in the form:

$$P(S_i = 1|Par(S_i)) = \sigma(\sum_{S_j \in Par(S_i)} J_{ij} S_j),$$

where S_i are the variables, $Par(S_i)$ are the variables which S_i depends on (its parents), σ denotes the logistic sigmoid function, and J_{ij} is the weight for the edge from variable S_j to variable S_i in the graphical model. SBNs are similar to feed-forward neural networks, but unlike neural networks, SBNs have a precise probabilistic semantics for their hidden variables. ISBNs are based on a generalized version of SBNs where variables with any range of discrete values are allowed. The normalized exponential function ('soft-max') is used to define the conditional probability distributions at these nodes.

To extend SBNs for processing arbitrarily long sequences, such as a parser's sequence of decisions $D^1,...,D^m$, SBNs are extended to a form of Dynamic Bayesian Network (DBN). In DBNs, a new set of variables is instantiated for each position in the sequence, but the edges and weights are the same for each position in the sequence. The edges which connect variables instantiated for different positions must be directed forward in the sequence, thereby allowing a temporal interpretation of the sequence.

Incremental Sigmoid Belief Networks [46] differ from simple dynamic SBNs in that they allow the model structure to depend on the output variable values. Specifically, a decision is allowed to affect the placement of any edge whose destination is after the decision. This results in a form of switching model [31], where each decision switches the model structure used for the remaining decisions. The incoming edges for a given position are a discrete function of the sequence of decisions which precede that position. This makes the ISBN an "incremental" model, not just a dynamic model. The structure of the model is determined incrementally as the decision sequence proceeds.

ISBNs are designed to allow the model structure to depend on the output values without overly complicating the inference of the desired conditional probabilities $P(D^t|D^1,\ldots,D^{t-1})$, the probability of the next decision given the history of previous decisions. In particular, it is never necessary to sum over all possible model structures, which in general would make inference intractable.

2.2 Modeling Structures with ISBNs

ISBNs are designed for modeling structured data where the output structure is not given as part of the input. In dependency parsing, this means they can model the probability of an output dependency structure when the input only specifies the sequence of words (i.e. parsing). The difficulty with such problems is that the statistical dependencies in the dependency structure are local in the structure, and not necessarily local in the word sequence. ISBNs allow us to capture these statistical dependencies in the model structure by having model edges depend on the output variables which specify the dependency structure. For example, if an output specifies that there is a dependency arc from word w_i to word w_j, then any future decision involving w_j can directly depend on its head w_i. This allows the head w_i to be treated as local to the dependent w_j even if they are far apart in the sentence.

This structurally-defined notion of locality is particularly important for the model's latent variables. When the structurally-defined model edges connect latent variables, information can be propagated between latent variables, thereby providing an even larger structural domain of locality than that provided by single edges. This provides a potentially powerful form of feature induction, which is nonetheless biased toward a notion of locality which is appropriate for the structure of the problem, as discussed further in section 4.

2.3 Approximating ISBNs

In [46] we proposed two approximations for inference in ISBNs, both based on variational methods. The main idea of variational methods [22] is, roughly, to construct a tractable approximate model with a number of free parameters. The values of 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.

The simplest example of a variational method is the mean field method, which uses a fully factorized distribution $Q(H|V)=\prod_i Q_i(h_i|V)$ as the approximate model, where V are the visible (i.e. known) variables, $H = h_1,\ldots,h_l$ are the hidden (i.e. latent) variables, and each Q_i is the distribution of an individual latent variable h_i. The free parameters of this approximate model are the means μ_i of the distributions Q_i.

We proposed two approximate models based on the variational approach in [46]. First, we showed that the neural network of [17] can be viewed as a coarse mean field approximation of ISBNs, which we call the feed-forward approximation. This approximation imposes the constraint that the free parameters μ_i of the approximate model are only allowed to depend on the distributions of their parent variables. This constraint increases the potential for a large approximation error, but it significantly simplifies the computations by allowing all the free parameters to be set in a single pass over the model.

The second approximation proposed in [46] takes into consideration the fact that, after each decision is made, all the preceding latent variables should have their means μ_i updated. This approximation extends the feed-forward approximation to account for the most important components of this update. We call this approximation the mean field approximation, because a mean field approximation is applied to handle the statistical dependencies introduced by the new decisions. This approximation was shown to be a more accurate approximation of ISBNs than the feed-forward approximation, but to remain tractable. It was also shown to achieve significantly better accuracy on constituent parsing.

2.4 Learning

Training these approximations of ISBNs is done to maximize the fit of the *approximate* models to the data. We use gradient descent, and a regularized maximum likelihood objective function. Gaussian regularization is applied, which is equivalent to the weight decay standardly used in neural networks. Regularization was reduced through the course of learning.

Gradient descent requires computing the derivatives of the objective function with respect to the model parameters. In the feed-forward approximation, this can be done with the standard Backpropagation learning used with neural networks. For the mean field approximation, propagating the error all the way back through the structure of the graphical model requires a more complicated calculation, but it can still be done efficiently (see [46] for details).

3 The Dependency Parsing Algorithm

The sequences of decisions $D^1,\ldots,D^m$ which we will be modeling with ISBNs are the sequences of decisions made by a dependency parser. For this we use the parsing strategy for projective dependency parsing introduced in [35], which is similar to a standard shift-reduce algorithm for context-free grammars [3]. It can be viewed as a mixture of bottom-up and top-down parsing strategies, where left dependencies are constructed in a bottom-up fashion and right dependencies are constructed top-down. For details we refer the reader to [35]. In this section we briefly describe the algorithm and explain how we use it to define our history-based probability model.

In this paper, as in the CoNLL-2006 and CoNLL-2007 shared tasks, we consider labeled dependency parsing. The state of the parser is defined by the current stack S, the queue I of remaining input words and the partial labeled dependency structure constructed by previous parser decisions. The parser starts with an empty stack S and terminates when it reaches a configuration with an empty queue I. The algorithm uses 4 types of decisions:

1. The decision **Left-Arc**$_r$ adds a dependency arc from the next input word w_j to the word w_i on top of the stack and selects the label r for the relation between w_i and w_j. Word w_i is then popped from the stack.
2. The decision **Right-Arc**$_r$ adds an arc from the word w_i on top of the stack to the next input word w_j and selects the label r for the relation between w_i and w_j.
3. The decision **Reduce** pops the word w_i from the stack.
4. The decision **Shift**$_{w_j}$ shifts the word w_j from the queue to the stack.

Unlike the original definition in [35] the **Right-Arc**$_r$ decision does not shift w_j to the stack. However, the only thing the parser can do after a **Right-Arc**$_r$ decision is to choose the **Shift**$_{w_j}$ decision. This subtle modification does not change the actual parsing order, but it does simplify the definition of our graphical model, as explained in section 4.

We use a history-based probability model, which decomposes the probability of the parse according to the parser decisions:

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

where T is the parse tree and $D^1, \ldots, D^m$ is its equivalent sequence of parser decisions. Since we need a generative model, the action **Shift**$_{w_j}$ also predicts the next word in the queue I, w_{j+1}, thus the $P(Shift_{w_j} | D^1, \ldots, D^{t-1})$ is a probability both of the shift operation and the word w_{j+1} conditioned on current parsing history.[1]

Instead of treating each D^t as an atomic decision, it is convenient to split it into a sequence of elementary decisions $D^t = d^t_1, \ldots, d^t_n$:

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

where $h(t,k)$ denotes the parsing history $D^1, \ldots, D^{t-1}, d^t_1, \ldots, d^t_{k-1}$. We split **Left-Arc**$_r$ and **Right-Arc**$_r$ each into two elementary decisions: first, the parser decides to create the corresponding arc, then, it decides to assign a relation r to the arc. Similarly, we decompose the decision **Shift**$_{w_j}$ into an elementary decision to shift a word and a prediction of the word w_{j+1}. In our experiments we use datasets from the CoNLL-2006 and CoNLL-2007 shared tasks, which provide additional properties for each word token, such as its part-of-speech tag and some fine-grain features. This information implicitly induces word clustering, which we use in our model: first we predict a part-of-speech tag for the word, then a set of word features, treating feature combination as an atomic value, and only then a particular word form. This approach allows us to both decrease the effect of sparsity and to avoid normalization across all the words in the vocabulary, significantly reducing the computational expense of word prediction.

To overcome a minor shortcoming of the parsing algorithm of [35] we introduce a simple language independent post-processing step. Nivre's parsing algorithm allows unattached nodes to stay on the stack at the end of parsing, which is reasonable for treebanks with unlabeled attachment to root. However, this sometimes happens with languages where only labeled attachment to root is allowed. In these cases (only 35 tokens in Greek, 17 in Czech, 1 in Arabic, on the final testing set of the CoNLL-2007 shared task) we attached them using a simple rule: if there are no tokens in the sentence attached to root, then the considered token is attached to root with the most frequent root-attachment relation used for its part-of-speech tag. If there are other root-attached tokens in the sentence, it is attached to the next root-attached token

[1] In preliminary experiments, we also considered look-ahead, where the word is predicted earlier than it appears at the head of the queue I, and "anti-look-ahead", where the word is predicted only when it is shifted to the stack S. Early prediction allows conditioning decision probabilities on the words in the look-ahead and, thus, speeds up the search for an optimal decision sequence. However, the loss of accuracy with look-ahead was quite significant. The described method, where a new word is predicted when it appears at the head of the queue, led to the most accurate model and quite efficient search. The anti-look-ahead model was both less accurate and slower.

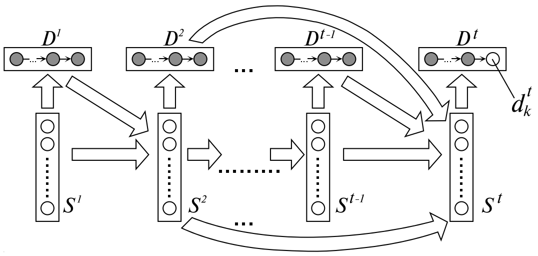

Fig. 1 An ISBN for estimating $P(d_k^t|h(t,k))$.

with the most frequent relation. Preference is given to the most frequent attachment direction for its part-of-speech tag. This rule guarantees that no loops are introduced by the post-processing.

4 An ISBN for Dependency Parsing

In this section we define the ISBN model we use for dependency parsing. An example of this ISBN for estimating $P(d_k^t|h(t,k))$ is illustrated in figure 1. It is organized into vectors of variables: latent state variable vectors $S^{t'} = s_1^{t'}, \ldots, s_n^{t'}$, representing an intermediate state at position t', and decision variable vectors $D^{t'}$, representing a decision at position t', where $t' \leq t$. Variables whose value are given at the current decision (t,k) are shaded in figure 1, latent and current decision variables are left unshaded.

As illustrated by the edges in figure 1, the probability of each state variable $s_i^{t'}$ (the individual circles in $S^{t'}$) 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. We will first discuss the state-to-state statistical dependencies, then the decision-to-state statistical dependencies, and finally the statistical dependencies to decision variables.

The most important design decision in building an ISBN model is choosing the finite set of relevant previous state vectors for the current decision. By connecting to a previous state, we place that state in the local context of the current decision. This specification of the domain of locality determines the inductive bias of learning with ISBNs. When deciding what information to store in its latent variables, an ISBN is more likely to choose information which is immediately local to the current decision. This stored information then becomes local to any following connected decision, where it again has some chance of being chosen as relevant to that decision. In this way, the information available to a given decision can come from arbitrarily far away in the chain of interconnected states, but it is much more likely to come from a state which is relatively local. Thus, we need to choose the set of local (i.e. connected) states in accordance with our prior knowledge about which previous decisions are likely to be particularly relevant to the current decision.

To choose which previous decisions are particularly relevant to the current decision, we make use of the partial dependency structure which has been decided so far in the parse. Specifically, the current latent state vector is connected to a set of 7 previous latent state vectors (if they exist) according to the following relationships:

1. **Input Context**: the last previous state with the same queue I.
2. **Stack Context**: the last previous state with the same stack S.
3. **Right Child of Top of** S: the last previous state where the rightmost right child of the current stack top was on top of the stack.
4. **Left Child of Top of** S: the last previous state where the leftmost left child of the current stack top was on top of the stack.
5. **Left Child of Front of** I[2] : the last previous state where the leftmost child of the front element of I was on top of the stack.
6. **Head of Top**: the last previous state where the head word of the current stack top was on top of the stack.
7. **Top of** S **at Front of** I: the last previous state where the current stack top was at the front of the queue.

Each of these 7 relations has its own distinct weight matrix for the resulting edges in the ISBN, but the same weight matrix is used at each position where the relation is relevant.

All these relations but the last one are motivated by linguistic considerations. The current decision is primarily about what to do with the current word on the top of the stack and the current word on the front of the queue. The *Input Context* and *Stack Context* relationships connect to the most recent states used for making decisions about each of these words. The *Right Child of Top of S* relationship connects to a state used for making decisions about the most recently attached dependent of the stack top. Similarly, the *Left Child of Front of I* relationship connects to a state for the most recently attached dependent of the queue front. The *Left Child of Top of S* is the first dependent of the stack top, which is a particularly informative dependent for many languages. Likewise, the *Head of Top* can tell us a lot about the stack top, if it has been chosen already.

A second motivation for including a state in the local context of a decision is that it might contain information which has no other route for reaching the current decision. In particular, it is generally a good idea to ensure that the immediately preceding state is always included somewhere in the set of connected states. This requirement ensures that information, at least theoretically, can pass between any two states in the decision sequence, thereby avoiding any hard independence assumptions. The last relation, *Top of S at Front of I*, is included mainly to fulfill this requirement. Otherwise, after a **Shift**$_{w_j}$ operation, the preceding state would not be selected by any of the relationships.

Each latent variable in the ISBN parser is also conditionally dependent on a set of explicit features of the parsing history, which are depicted in figure 1 as connections from decision vectors to state vectors. The precise set of explicit features

[2] We refer to the *head* of the queue as the *front*, to avoid unnecessary ambiguity of the word *head* in the context of dependency parsing.

can be adapted to a particular language. However, as long as these explicit features include all the new information from the last parser decision, the performance of the model is not very sensitive to this design choice. This is because the state-to-state connections give ISBNs the ability to induce their own complex features of the parse history, as discussed above. For the CoNLL-2006 experiments, we tried different feature sets to validate this hypothesis that the feature set is not very important. For most of the languages in the CoNLL-2007 experiments we used the set proposed in [36]. However, we had to remove from this set all the lookahead features to obtain a valid generative history-based model.

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:

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

where $\Phi_{h(t',k)}$ is the indicator function of the set of elementary decisions that may possibly follow the last decision in the history $h(t',k)$, and the W_{dj} are the weights. Now it is easy to see why the original decision **Right-Arc**$_r$ [35] had to be decomposed into two distinct decisions: the decision to construct a labeled arc and the decision to shift the word. Use of this composite **Right-Arc**$_r$ would have required the introduction of individual parameters for each pair (w,r), where w is an arbitrary word in the lexicon and r - an arbitrary dependency relation.

5 Searching for the Best Tree

ISBNs define a probability model which does not make any a-priori assumptions of independence between any decision variables. As we discussed in section 4, the use of relations based on partial output structure makes it possible to take into account statistical interdependencies between decisions closely related in the output structure, but separated by multiple decisions in the input structure. This property leads to exponential complexity for complete search. However, the success of the deterministic parsing strategy using the same parsing order [36], suggests that it should be relatively easy to find an accurate approximation to the best parse with heuristic search methods. Unlike [36], we can not use a lookahead in our generative model, as was discussed in section 3, so a greedy method is unlikely to lead to a good approximation. Instead we use a pruning strategy similar to that described in [17], where it was applied to a considerably harder search problem: constituent parsing with a left-corner parsing order.

We apply fixed beam pruning after each decision **Shift**$_{w_j}$, because knowledge of the next word in the queue I helps distinguish unlikely decision sequences. We could have used best-first search between **Shift**$_{w_j}$ operations, but this still leads to relatively expensive computations, especially when the set of dependency relations is large. However, most of the word pairs can possibly participate only in a very limited number of distinct relations. Thus, we pursue only a fixed number of relations r after each **Left-Arc**$_r$ and **Right-Arc**$_r$ operation.

Experiments with a variety of post-shift beam widths confirmed that very small validation performance gains are achieved with widths larger than 30, and sometimes even a beam of 5 was sufficient. We found also that allowing 5 different relations after each dependency prediction operation was enough that it had virtually no effect on the validation accuracy. For some experiments for the CoNLL-2007 shared task we used a larger beam, but as we will discuss in the empirical section, again, gains were not large.

6 Empirical Evaluation

In this section we evaluate the ISBN model for dependency parsing on three of the treebanks from the CoNLL-2006 shared task, and on all ten of the languages from the CoNLL-2007 shared task. Most of this section focuses on the experiments using the CoNLL-2006 data, but we also report our performance relative to the other systems which participated in the CoNLL-2007 shared task, and we analyze the efficiency of the model on this data.

We compare our generative models with the best parsers from the CoNLL-2006 task, including the SVM-based parser of [36] (the MALT parser), which uses the same parsing algorithm. To test the feature induction abilities of our model we compare results with two feature sets, the feature set tuned individually for each language by [36], and another feature set which includes only obvious local features. This simple feature set comprises only features of the word on top of the stack S and the front word of the queue I. We compare the gain from using tuned features with the similar gain obtained by the MALT parser. To obtain these results we train the MALT parser with the same two feature sets.[3]

In order to distinguish the contribution of ISBN's feature induction abilities from the contribution of our estimation method and search, we perform another experiment. We use the tuned feature set and disable the feature induction abilities of the model by removing all the edges between latent variables vectors. Comparison of this restricted model with the full ISBN model shows how important the feature induction is. Also, comparison of this restricted model with the MALT parser, which uses the same set of features, indicates whether our generative estimation method and use of beam search is beneficial.

[3] The tuned feature sets were obtained from http://w3.msi.vxu.se/˜nivre/research/MaltParser.html. We removed lookahead features for ISBN experiments but preserved them for experiments with the MALT parser. Analogously, we extended simple features with 3 words lookahead for the MALT parser experiments.

6.1 Experimental Setup for the CoNLL-2006 Data

For the CoNLL-2006 experiments, we used the CoNLL-2006 distributions of Danish DDT treebank [24], Dutch Alpino treebank [4] and Slovene SDT treebank [14]. The choice of these treebanks was motivated by the fact that they all are freely distributed and have very different sizes of their training sets: 195,069 tokens for Dutch, 94,386 tokens for Danish and only 28,750 tokens for Slovene. It is generally believed that discriminative models win over generative models with a large amount of training data, so we expected to see a similar trend in our results. Test sets are about equal and contain about 5,000 scoring tokens.

We followed the experimental setup of the shared task and used all the information provided for the languages: gold standard part-of-speech tags and coarse part-of-speech tags, word form, word lemma (lemma information was not available for Danish) and a set of fine-grain word features. As we explained in section 3, we treated these sets of fine-grain features as an atomic value when predicting a word. However, when conditioning on words, we treated each component of this composite feature individually, as it proved to be useful on the development set. We used frequency cutoffs: we ignored any property (e.g., word form, feature or even part-of-speech tag[4]) which occurs in the training set less than 5 times. Following [36], we used pseudo-projective transformation they proposed to cast non-projective parsing tasks as projective.

ISBN models were trained using a small development set taken out from the training set, which was used for tuning learning parameters and for early stopping. The sizes of the development sets were: 4,988 tokens for larger Dutch corpus, 2,504 tokens for Danish and 2,033 tokens for Slovene. The MALT parser was trained always using the entire training set. We expect that the mean field approximation should demonstrate better results than feed-forward approximation on this task as it is theoretically expected and confirmed on the constituent parsing task [46]. However, the sizes of testing sets would not allow us to perform any conclusive analysis, so we decided not to perform these comparisons here. Instead we used the mean field approximation for the smaller two corpora and used the feed-forward approximation for the larger one. Training the mean field approximations on the larger Dutch treebank is feasible, but would significantly reduce the possibilities for tuning the learning parameters on the development set and, thus, would increase the randomness of model comparisons.

All model selection was performed on the development set and a single model of each type was applied to the testing set. We used a state variable vector consisting of 80 binary variables, as it proved sufficient on the preliminary experiments. For the MALT parser we replicated the parameters from [36] as described in detail on their web site.

The labeled attachment scores for the ISBN with tuned features (TF) and local features (LF) and ISBN with tuned features and no edges connecting latent variable

[4] Part-of-speech tags for multi-word units in the Danish treebank were formed as concatenation of tags of the words, which led to quite sparse set of part-of-speech tags.

Table 1 Labeled attachment score on the CoNLL-2006 testing sets of Danish, Dutch and Slovene treebanks.

		Danish	Dutch	Slovene
ISBN	TF	85.0	79.6	72.9
	LF	84.5	79.5	72.4
	TF-NA	83.5	76.4	71.7
MALT	TF	85.1	78.2	70.5
	LF	79.8	74.5	66.8
MST		84.8	79.2	73.4
Average score		78.3	70.7	65.2

vectors (TF-NA) are presented in table 1, along with results for the MALT parser both with tuned and local feature, the MST parser [29], and the average score across all systems in the CoNLL-2006 shared task. The MST parser is included because it demonstrated the best overall result in the task, non significantly outperforming the MALT parser, which, in turn, achieved the second best overall result. The labeled attachment score is computed using the same method as in the CoNLL-2006 shared task, i.e. ignoring punctuation. Note, that though we tried to completely replicate training of the MALT parser with the tuned features, we obtained slightly different results. The original published results for the MALT parser with tuned features were 84.8% for Danish, 78.6% for Dutch and 70.3% for Slovene. The improvement of the ISBN models (TF and LF) over the MALT parser is statistically significant for Dutch and Slovene. Differences between their results on Danish are not statistically significant.

6.2 Discussion of Results on the CoNLL-2006 Data

The ISBN with tuned features (TF) achieved significantly better accuracy than the MALT parser on 2 languages (Dutch and Slovene), and demonstrated essentially the same accuracy on Danish. The results of the ISBN are among the two top published results on all three languages, including the best published results on Dutch. All three models, MST, MALT and ISBN, demonstrate much better results than the average result in the CoNLL-2006 shared task. These results suggest that our generative model is quite competitive with respect to the best models, which are both discriminative.[5] We would expect further improvement of ISBN results if we applied discriminative retraining [18] or reranking with data-defined kernels [20], even without introduction of any additional features.

We can see that the ISBN parser achieves about the same results with local features (LF). Local features by themselves are definitely not sufficient for the construction of accurate models, as seen from the results of the MALT parser with

[5] Note that the development set accuracy predicted correctly the testing set ranking of ISBN TF, LF and TF-NA models on each of the datasets, so it is fair to compare the best ISBN result among the three with other parsers.

Table 2 F_1 score of labeled attachment as a function of dependency length on the CoNLL-2006 testing sets of Danish, Dutch and Slovene.

		to root	1	2	3 - 6	> 6
Danish	ISBN	95.1	95.7	90.1	84.1	74.7
	MALT	95.4	96.0	90.8	84.0	71.6
Dutch	ISBN	79.8	92.4	86.2	81.4	71.1
	MALT	73.1	91.9	85.0	76.2	64.3
Slovene	ISBN	76.1	92.5	85.6	79.6	54.3
	MALT	59.9	92.1	85.0	78.4	47.1
Average	ISBN	83.6	93.5	87.3	81.7	66.7
	MALT	76.2	93.3	87.0	79.5	61.0
	Improvement	**7.5**	**0.2**	**0.4**	**2.2**	**5.7**

local features (and look-ahead). This result demonstrates that ISBNs are a powerful model for feature induction.

The results of the ISBN without edges connecting latent state vectors is slightly surprising and suggest that without feature induction the ISBN is significantly worse than the best models. This shows that the improvement is coming mostly from the ability of the ISBN to induce complex features and not from either using beam search or from the estimation procedure. It might also suggest that generative models are probably worse for the dependency parsing task than discriminative approaches (at least for larger datasets). This motivates further research into methods which combine powerful feature induction properties with the advantage of discriminative training. Although discriminative reranking of the generative model is likely to help, the derivation of fully discriminative feature induction methods is certainly more challenging.

In order to better understand differences in performance between ISBN and MALT, we analyzed how relation accuracy changes with the length of the head-dependent relation. The harmonic mean between precision and recall of labeled attachment, F_1 measure, for the ISBN and MALT parsers with tuned features is presented in table 2. F_1 score is computed for four different ranges of lengths and for attachments directly to root. Along with the results for each of the languages, the table includes their mean (Average) and the absolute improvement of the ISBN model over MALT (Improvement). It is easy to see that accuracy of both models is generally similar for small distances (1 and 2), but as the distance grows the ISBN parser starts to significantly outperform MALT, achieving 5.7% average improvement on dependences longer than 6 word tokens. When the MALT parser does not manage to recover a long dependency, the highest scoring action it can choose is to reduce the dependent from the stack without specifying its head, thereby attaching the dependent to the root by default. This explains the relatively low F_1 scores for attachments to root (evident for Dutch and Slovene): though recall of attachment to root is comparable to that of the ISBN parser (82.4% for MALT against 84.2% for ISBN, on average over 3 languages), precision for the MALT parser is much worse (71.5% for MALT against 83.1% for ISBN, on average).

The considerably worse accuracy of the MALT parser on longer dependencies might be explained both by use of a non-greedy search method in the ISBN and the ability of ISBNs to induce history features. To capture a long dependency, the MALT parser should keep a word on the stack during a long sequence of decision. If at any point during the intermediate steps this choice seems not to be locally optimal, then the MALT parser will choose the alternative and lose the possibility of the long dependency.[6] By using a beam search, the ISBN parser can maintain the possibility of the long dependency in its beam even when other alternatives seem locally preferable. Also, long dependencies are often more difficult, and may be systematically different from local dependencies. The designer of a MALT parser needs to discover predictive features for long dependencies by hand, whereas the ISBN model can automatically discover them. Thus we expect that the feature induction abilities of ISBNs have a strong effect on the accuracy of long dependencies. This prediction is confirmed by the differences between the results of the normal ISBN (TF) and the restricted ISBN (TF-NA) model. The TF-NA model, like the MALT parser, is biased toward attachment to root; it attaches to root 12.0% more words on average than the normal ISBN, without any improvement of recall and with a great loss of precision. The F_1 score on long dependencies for the TF-NA model is also negatively effected in the same way as for the MALT parser. This confirms that the ability of the ISBN model to induce features is a major factor in improving accuracy of long dependencies.

6.3 The CoNLL-2007 Experiments

We also evaluated the ISBN parser on all 10 languages considered in the CoNLL-2007 shared task [15, 2, 27, 10, 5, 26, 21, 42, 13, 30, 37]. For these experiments we used only the simplest feed-forward approximation of an ISBN. We would expect better performance with the more accurate approximation based on variational inference proposed and evaluated in [46]. We did not try this because, on larger treebanks it would have taken too long to tune the model with this better approximation, and using different approximation methods for different languages would not be compatible with the shared task rules.

ISBN models were trained using a small development set taken out from the training set, which was used for tuning learning and decoding parameters, for early stopping and very coarse feature engineering. The sizes of the development sets were different: starting from less than 2,000 tokens for smaller treebanks to 5,000 tokens for the largest one. The relatively small sizes of the development sets limited our ability to perform careful feature selection, but this should not have significantly affected the model performance, as suggested by the small difference in scores be-

[6] The MALT parser is trained to keep the word as long as possible: if both **Shift** and **Reduce** decisions are possible during training, it always prefers to shift. Though this strategy should generally reduce the described problem, it is evident from the low precision score for attachment to root, that it can not completely eliminate it.

Table 3 Labeled attachment score (LAS) and unlabeled attachment score (UAS) on the final CoNLL-2007 testing sets

	Arabic	Basque	Catalan	Chinese	Czech	English	Greek	Hungarian	Italian	Turkish	**Average**
LAS	74.1	75.5	87.4	82.1	77.9	88.4	73.5	77.9	82.3	79.8	**79.90**
UAS	83.2	81.9	93.4	87.9	84.2	89.7	81.2	82.2	86.3	86.2	**85.62**

tween the model with tuned features (TF) and the model with local features (LF) obtained in the previous set of experiments.[7] We used the base feature model defined in [36] for all the languages but Arabic, Chinese, Czech, and Turkish. For Arabic, Chinese, and Czech, we used the same feature models used in the CoNLL-2006 shared task by [36], and for Turkish we used again the base feature model but extended it with a single feature: the part-of-speech tag of the token preceding the current top of the stack. We used frequency cutoffs of 20 for Greek and Chinese and a cutoff of 5 for the rest. Because the threshold value affects the model efficiency, we selected the larger threshold when validation results with the smaller threshold were comparable.

Results on the final testing set are presented in table 3. Unlike the 2006 shared task, in the CoNLL-2007 shared task punctuation tokens were included in evaluation. The model achieves relatively high scores on each individual language, significantly better than each average result in the shared task. This leads to the third best overall average results in the shared task, both in average labeled attachment score and in average unlabeled attachment score. The absolute error increase in labeled attachment score over the best system is only 0.4%. We attribute ISBN's success mainly to its ability to automatically induce features, as this significantly reduces the risk of omitting any important highly predictive features. This makes an ISBN parser a particularly good baseline when considering a new treebank or language, because it does not require much effort in feature engineering.

It is also important to note that the model is quite efficient. Figure 2 shows the tradeoff between accuracy and parsing time as the width of the search beam is varied, on the development set. This curve plots the average labeled attachment score over Basque, Chinese, English, and Turkish as a function of parsing time per token.[8] Accuracy of only 1% below the maximum can be achieved with average processing time of 17 ms per token, or 60 tokens per second.[9]

[7] Use of cross-validation with our model is relatively time-consuming and, thus, not quite feasible for the shared task.

[8] A piecewise-linear approximation for each individual language was used to compute the average. Experiments were run on a standard 2.4 GHz desktop PC.

[9] For Basque, Chinese, and Turkish this time is below 7 ms, but for English it is 38 ms. English, along with Catalan, required the largest beam across all 10 languages. Note that accuracy in the lowest part of the curve can probably be improved by varying latent vector size and frequency cut-offs. Also, efficiency was not the main goal during the implementation of the parser, and it is likely that a much faster implementation is possible.

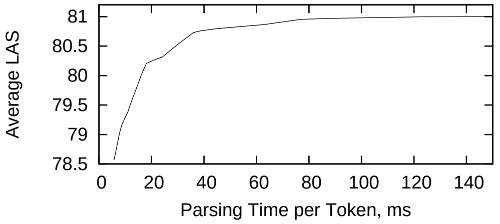

Fig. 2 Average labeled attachment score on Basque, Chinese, English, and Turkish CoNLL-2007 development sets as a function of parsing time per token

7 Related Work

There has not been much previous work on latent variable models for dependency parsing. Dependency parsing with Dynamic Bayesian Networks was considered in [38], with limited success. Roughly, the model considered the whole sentence at a time, with the DBN 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. Recently several latent variable models for constituent parsing have been proposed [23, 28, 41, 43, 39, 25, 40]. In [28, 41, 39, 25, 40] non-terminals in a standard PCFG model are augmented with latent variables. The best model from this class uses the split-and-merged approach to discover the appropriate split of non-terminals [39, 40], it demonstrated state-of-the-art results on constituent parsing. This approach was very recently extended to dependency parsing [32], the authors proposed transforms of dependency grammars to CFG grammars annotated with latent variables. However, the resulting parser does not achieve state-of-the-art accuracy. In [23], an undirected graphical model for constituent parse reranking uses dependency relations to define the edges. Thus, it should be easy to apply a similar method to reranking dependency trees.

Undirected graphical models, in particular Conditional Random Fields, are the standard tools for shallow parsing [45]. However, shallow parsing is effectively a sequence labeling problem and therefore differs significantly from full parsing. As discussed in [46], undirected graphical models do not seem to be suitable for history-based parsing models.

Sigmoid Belief Networks (SBNs) were used originally for character recognition tasks, but later a dynamic modification of this model was applied to the reinforcement learning task [44]. However, their graphical model, approximation method, and learning method differ significantly from those of this paper. The extension of dynamic SBNs with incrementally specified model structure (i.e. Incremental Sigmoid Belief Networks, used in this paper) was proposed and applied to constituent parsing in [46].

Recent work has extended this ISBN model of dependency parsing to joint dependency parsing and semantic role labeling [19]. There, ISBNs are used to model synchronous derivations of syntactic and semantic structures. Semantic role label (SRL) structures are specified with a less constrained version of the derivations discussed here. The incremental nature of these derivations is exploited by synchronising the two derivations at each word. An ISBN is designed which induces features which capture statistical dependencies both within each derivation and between derivations. The ISBN is trained as a generative model of these synchronous derivations, thereby providing an estimate of the joint probability of the syntactic and semantic structures. The model achieves competitive results despite minimal feature engineering.

8 Conclusions

We proposed a latent variable dependency parsing model based on Incremental Sigmoid Belief Networks. Unlike state-of-the-art dependency parsers, it uses a generative history-based model. We demonstrated that it achieves state-of-the-art results on a selection of languages from the CoNLL-2006 shared task, and achieves competitive accuracy on all 10 languages in the CoNLL-2007 shared task. This parser was ranked third best overall in the CoNLL-2007 shared task, and the best single-model system. It uses a vector of latent variables to represent an intermediate state and uses relations defined on the output structure to construct the edges between latent state vectors. These properties make it a powerful feature induction method for dependency parsing, and it achieves competitive results even with very simple explicit features.

The ISBN model is especially accurate at modeling long dependencies, achieving average improvement of 5.7% over the state-of-the-art baseline on dependences longer than 6 words. Empirical evaluation demonstrates that competitive results are achieved mostly because of the ability of the model to induce complex features and not because of the use of a generative probability model or a specific search method. This automatic feature induction means that the proposed model requires minimal design effort, which is highly desirable when using new treebanks or languages. The parsing time needed to achieve high accuracy is also quite small, making this model a good candidate for use in practical applications. As with other generative models, it can be further improved by the application of discriminative reranking techniques. Discriminative methods are likely to allow it to significantly improve over the current state-of-the-art in dependency parsing.[10]

Acknowledgements This work was funded by Swiss NSF grant 200020-109685, Swiss NSF Fellowship PBGE22-119276, UK EPSRC grant EP/E019501/1, EU FP6 grant 507802 (TALK project), and EU FP7 grant 216594 (CLASSiC project).

References

1. Abeillé, A. (ed.): Treebanks: Building and Using Parsed Corpora. Kluwer (2003)

[10] The ISBN dependency parser is downloadable from the authors' web-page.

2. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Atutxa, A., de Ilarraza, A.D., Garmendia, A., Oronoz, M.: Construction of a Basque dependency treebank. In: Proc. of the 2nd Workshop on Treebanks and Linguistic Theories (TLT), pp. 201–204 (2003)
3. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques and Tools. Addison Wesley (1986)
4. van der Beek, L., Bouma, G., Daciuk, J., Gaustad, T., Malouf, R., van Noord, G., Prins, R., Villada, B.: The Alpino dependency treebank. Computational Linguistic in the Netherlands (CLIN) (2002)
5. Böhmová, A., Hajič, J., Hajičová, E., Hladká, B.: The PDT: a 3-level annotation scenario. In: Abeillé [1], chap. 7, pp. 103–127
6. Bottou, L.: Une approche théoretique de l'apprentissage connexionniste: Applications à la reconnaissance de la parole. Ph.D. thesis, Université de Paris XI, Paris, France (1991)
7. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proc. of the Tenth Conference on Computational Natural Language Learning. New York, USA (2006)
8. Charniak, E.: A maximum-entropy-inspired parser. In: Proc. 1st Meeting of North American Chapter of Association for Computational Linguistics, pp. 132–139. Seattle, Washington (2000)
9. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In: Proc. 43rd Meeting of Association for Computational Linguistics, pp. 173–180. Ann Arbor, MI (2005)
10. Chen, K., Luo, C., Chang, M., Chen, F., Chen, C., Huang, C., Gao, Z.: Sinica treebank: Design criteria, representational issues and implementation. In: Abeillé [1], chap. 13, pp. 231–248
11. Collins, M.: Head-driven statistical models for natural language parsing. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA (1999)
12. Collins, M.: Discriminative reranking for natural language parsing. In: Proc. 17th Int. Conf. on Machine Learning, pp. 175–182. Stanford, CA (2000)
13. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. Springer (2005)
14. Dzeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Zabokrtsky, Z., Zele, A.: Towards a Slovene dependency treebank. In: Proc. Int. Conf. on Language Resources and Evaluation (LREC). Genoa, Italy (2006)
15. Hajič, J., Smrž, O., Zemánek, P., Šnaidauf, J., Beška, E.: Prague Arabic dependency treebank: Development in data and tools. In: Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools, pp. 110–117 (2004)
16. Hall, J., Nilsson, J., Nivre, J., Eryigit, G., Megyesi, B., Nilsson, M., Saers, M.: Single malt or blended? a study in multilingual parser optimization. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 933–939 (2007)
17. Henderson, J.: Inducing history representations for broad coverage statistical parsing. In: Proc. joint meeting of North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conf., pp. 103–110. Edmonton, Canada (2003)
18. Henderson, J.: Discriminative training of a neural network statistical parser. In: Proc. 42nd Meeting of Association for Computational Linguistics. Barcelona, Spain (2004)
19. Henderson, J., Merlo, P., Musillo, G., Titov, I.: A latent variable model of synchronous parsing for syntactic and semantic dependencies. In: Proc. CoNLL-2008 Shared Task. Manchester, UK (2008)
20. Henderson, J., Titov, I.: Data-defined kernels for parse reranking derived from probabilistic models. In: Proc. 43rd Meeting of Association for Computational Linguistics, pp. 181–188. Ann Arbor, MI (2005)
21. Johansson, R., Nugues, P.: Extended constituent-to-dependency conversion for English. In: Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA) (2007)
22. Jordan, M.I., Z.Ghahramani, Jaakkola, T.S., Saul., L.K.: An introduction to variational methods for graphical models. In: M.I. Jordan (ed.) Learning in Graphical Models. MIT Press, Cambridge, MA (1999)
23. Koo, T., Collins, M.: Hidden-variable models for discriminative reranking. In: Proc. Conf. on Empirical Methods in Natural Language Processing. Vancouver, B.C., Canada (2005)

24. Kromann, M.T.: The Danish dependency treebank and the underlying linguistic theory. In: Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT). Vaxjo, Sweden (2003)
25. Liang, P., Petrov, S., Jordan, M., Klein, D.: The infinite PCFG using hierarchical dirichlet processes. In: Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Prague, Czech Republic (2007)
26. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics **19**(2), 313–330 (1993)
27. Martí, M.A., Taulé, M., Màrquez, L., Bertran, M.: CESS-ECE: A multilingual and multilevel annotated corpus. Available for download from: http://www.lsi.upc.edu/∼mbertran/cess-ece/ (2007)
28. Matsuzaki, T., Miyao, Y., Tsujii, J.: Probabilistic CFG with latent annotations. In: Proceedings of the 43rd Annual Meeting of the ACL. Ann Arbor, MI (2005)
29. McDonald, R., Lerman, K., Pereira, F.: Multilingual dependency analysis with a two-stage discriminative parser. In: Proc. of the Tenth Conference on Computational Natural Language Learning. New York, USA (2006)
30. Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzari, O., Lenci, A., Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M.T., Saracino, D., Zanzotto, F., Nana, N., Pianesi, F., Delmonte, R.: Building the Italian Syntactic-Semantic Treebank. In: Abeillé [1], chap. 11, pp. 189–210
31. Murphy, K.P.: Dynamic belief networks: Representation, inference and learning. Ph.D. thesis, University of California, Berkeley, CA (2002)
32. Musillo, G., Merlo, P.: Unlexicalised hidden variable models of split dependency grammars. In: Proc. 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Columbus, Ohio, USA (2008)
33. Nakagawa, T.: Multilingual dependency parsing using global features. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 952–956 (2007)
34. Neal, R.: Connectionist learning of belief networks. Artificial Intelligence **56**, 71–113 (1992)
35. Nivre, J., Hall, J., Nilsson, J.: Memory-based dependency parsing. In: Proc. of the Eighth Conference on Computational Natural Language Learning, pp. 49–56. Boston, USA (2004)
36. Nivre, J., Hall, J., Nilsson, J., Eryigit, G., Marinov, S.: Pseudo-projective dependency parsing with support vector machines. In: Proc. of the Tenth Conference on Computational Natural Language Learning, pp. 221–225. New York, USA (2006)
37. Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G.: Building a Turkish treebank. In: Abeillé [1], chap. 15, pp. 261–277
38. Peshkin, L., Savova, V.: Dependency parsing with dynamic Bayesian network. In: AAAI, 20th National Conference on Artificial Intelligence. Pittsburgh, Pennsylvania (2005)
39. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proc. of the Annual Meeting of the ACL and the International Conference on Computational Linguistics. Sydney, Australia (2006)
40. Petrov, S., Klein, D.: Improved inference for unlexicalized parsing. In: Proceedings of the Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics (HLT-NAACL 2007). Rochester, USA (2007)
41. Prescher, D.: Head-driven PCFGs with latent-head statistics. In: Proc. 9th Int. Workshop on Parsing Technologies. Vancouver, Canada (2005)
42. Prokopidis, P., Desypri, E., Koutsombogera, M., Papageorgiou, H., Piperidis, S.: Theoretical and practical issues in the construction of a Greek dependency treebank. In: Proc. of the 4th Workshop on Treebanks and Linguistic Theories (TLT), pp. 149–160 (2005)
43. Riezler, S., King, T.H., Kaplan, R.M., Crouch, R., Maxwell, J.T., Johnson, M.: Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In: Proc. 40th Meeting of Association for Computational Linguistics. Philadelphia, PA (2002)
44. Sallans, B.: Reinforcement learning for factored markov decision processes. Ph.D. thesis, University of Toronto, Toronto, Canada (2002)

45. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proc. joint meeting of North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conf. Edmonton, Canada (2003)
46. Titov, I., Henderson, J.: Constituent parsing with incremental sigmoid belief networks. In: Proc. 45th Meeting of Association for Computational Linguistics (ACL). Prague, Czech Republic (2007)