Temporal Restricted Boltzmann Machines for Dependency Parsing

Nikhil Garg
Department of Computer Science
University of Geneva
Switzerland
nikhil.garg@unige.ch

James Henderson
Department of Computer Science
University of Geneva
Switzerland
james.henderson@unige.ch

Abstract

We propose a generative model based on Temporal Restricted Boltzmann Machines for transition based dependency parsing. The parse tree is built incrementally using a shift-reduce parse and an RBM is used to model each decision step. The RBM at the current time step induces latent features with the help of temporal connections to the relevant previous steps which provide context information. Our parser achieves labeled and unlabeled attachment scores of 88.72% and 91.65% respectively, which compare well with similar previous models and the state-of-the-art.

1 Introduction

There has been significant interest recently in machine learning methods that induce generative models with high-dimensional hidden representations, including neural networks (Bengio et al., 2003; Collobert and Weston, 2008), Bayesian networks (Titov and Henderson, 2007a), and Deep Belief Networks (Hinton et al., 2006). In this paper, we investigate how these models can be applied to dependency parsing. We focus on Shift-Reduce transition-based parsing proposed by Nivre et al. (2004). In this class of algorithms, at any given step, the parser has to choose among a set of possible actions, each representing an incremental modification to the partially built tree. To assign probabilities to these actions, previous work has proposed *memory-based classifiers* (Nivre et al., 2004), SVMs (Nivre et al., 2006b), and Incremental Sigmoid Belief Networks (ISBN) (Titov and Henderson, 2007b). In a related earlier work, Ratnaparkhi (1999) proposed a maximum entropy model for transition-based constituency parsing. Of these approaches, only ISBNs induce high-dimensional latent representations to encode parse history, but suffer from either very approximate or slow inference procedures.

We propose to address the problem of inference in a high-dimensional latent space by using an undirected graphical model, Restricted Boltzmann Machines (RBMs), to model the individual parsing decisions. Unlike the Sigmoid Belief Networks (SBNs) used in ISBNs, RBMs have tractable inference procedures for both forward and backward reasoning, which allows us to efficiently infer both the probability of the decision given the latent variables and vice versa. The key structural difference between the two models is that the directed connections between latent and decision vectors in SBNs become undirected in RBMs. A complete parsing model consists of a sequence of RBMs interlinked via directed edges, which gives us a form of Temporal Restricted Boltzmann Machines (TRBM) (Taylor et al., 2007), but with the incrementally specified model structure required by parsing. In this paper, we analyze and contrast ISBNs with TRBMs and show that the latter provide an accurate and theoretically sound model for parsing with high-dimensional latent variables.

2 An ISBN Parsing Model

Our TRBM parser uses the same history-based probability model as the ISBN parser of Titov and Henderson (2007b): $P(\text{tree}) = \Pi_t P(\mathbf{v}^t|\mathbf{v}^1, ..., \mathbf{v}^{t-1})$, where each

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 11–17,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

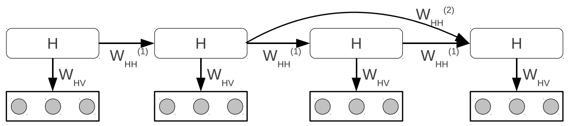

Figure 1: An ISBN network. Shaded nodes represent decision variables and 'H' represents a vector of latent variables. $W_{HH}^{(c)}$ denotes the weight matrix for directed connection of type c between two latent vectors.

$\mathbf{v}^t$ is a parser decision of the type *Left-Arc*, *Right-Arc*, *Reduce* or *Shift*. These decisions are further decomposed into sub-decisions, as for example $P(\textit{Left-Arc}|\mathbf{v}^1, ..., \mathbf{v}^{t-1})P(\text{Label}|\textit{Left-Arc}, \mathbf{v}^1, ..., \mathbf{v}^{t-1})$. The TRBMs and ISBNs model these probabilities.

In the ISBN model shown in Figure 1, the decisions are shown as boxes and the sub-decisions as shaded circles. At each decision step, the ISBN model also includes a vector of latent variables, denoted by 'H', which act as latent features of the parse history. As explained in (Titov and Henderson, 2007b), the temporal connections between latent variables are constructed to take into account the *structural locality* in the partial dependency structure. The model parameters are learned by backpropagating likelihood gradients.

Because decision probabilities are conditioned on the history, once a decision is made the corresponding variable becomes observed, or visible. In an ISBN, the directed edges to these visible variables and the large numbers of heavily inter-connected latent variables make exact inference of decision probabilities intractable. Titov and Henderson (2007a) proposed two approximation procedures for inference. The first was a feed forward approximation where latent variables were allowed to depend only on their parent variables, and hence did not take into account the current or future observations. Due to this limitation, the authors proposed to make latent variables conditionally dependent also on a set of explicit features derived from the parsing history, specifically, the base features defined in (Nivre et al., 2006b). As shown in our experiments, this addition results in a big improvement for the parsing task.

The second approximate inference procedure, called the incremental mean field approximation, extended the feed-forward approximation by updating the current time step's latent variables after each sub-decision. Although this approximation is more accurate than the feed-forward one, there is no analytical way to maximize likelihood w.r.t. the means of the latent variables, which requires an iterative numerical method and thus makes inference very slow, restricting the model to only shorter sentences.

3 Temporal Restricted Boltzmann Machines

In the proposed TRBM model, RBMs provide an analytical way to do exact inference within each time step. Although information passing between time steps is still approximated, TRBM inference is more accurate than the ISBN approximations.

3.1 Restricted Boltzmann Machines (RBM)

An RBM is an undirected graphical model with a set of binary visible variables $\mathbf{v}$, a set of binary latent variables $\mathbf{h}$, and a weight matrix $\mathbf{W}$ for bipartite connections between $\mathbf{v}$ and $\mathbf{h}$. The probability of an RBM configuration is given by: $p(\mathbf{v}, \mathbf{h}) = (1/Z)e^{-E(\mathbf{v},\mathbf{h})}$ where Z is the partition function and E is the energy function defined as:

$$E(\mathbf{v}, \mathbf{h}) = -\Sigma_i a_i v_i - \Sigma_j b_j h_j - \Sigma_{i,j} v_i h_j w_{ij}$$

where a_i and b_j are biases for corresponding visible and latent variables respectively, and w_{ij} is the symmetric weight between v_i and h_j. Given the visible variables, the latent variables are conditionally independent of each other, and vice versa:

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \Sigma_i v_i w_{ij}) \qquad (1)$$

$$p(v_i = 1|\mathbf{h}) = \sigma(a_i + \Sigma_j h_j w_{ij}) \qquad (2)$$

where $\sigma(x) = 1/(1 + e^{-x})$ (the logistic sigmoid).

RBM based models have been successfully used in image and video processing, such as Deep Belief Networks (DBNs) for recognition of hand-written digits (Hinton et al., 2006) and TRBMs for modeling motion capture data (Taylor et al., 2007). Despite their success, RBMs have seen limited use in the NLP community. Previous work includes RBMs for topic modeling in text documents (Salakhutdinov and Hinton, 2009), and *Temporal Factored RBM* for language modeling (Mnih and Hinton, 2007).

3.2 Proposed TRBM Model Structure

TRBMs (Taylor et al., 2007) can be used to model sequences where the decision at each step requires some context information from the past. Figure 2

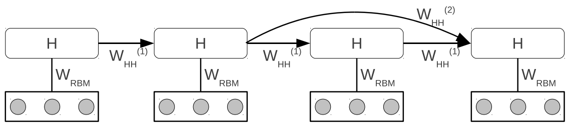

Figure 2: Proposed TRBM Model. Edges with no arrows represent undirected RBM connections. The directed temporal connections between time steps contribute a bias to the latent layer inference in the current step.

shows our proposed TRBM model with latent to latent connections between time steps. Each step has an RBM with weights W_{RBM} composed of smaller weight matrices corresponding to different sub-decisions. For instance, for the action *Left-Arc*, W_{RBM} consists of RBM weights between the latent vector and the sub-decisions: "Left-Arc" and "Label". Similarly, for the action *Shift*, the sub-decisions are "Shift", "Part-of-Speech" and "Word". The probability distribution of a TRBM is:

$$p(\mathbf{v}_1^T, \mathbf{h}_1^T) = \Pi_{t=1}^T p(\mathbf{v}^t, \mathbf{h}^t | \mathbf{h}^{(1)}, ..., \mathbf{h}^{(C)})$$

where $\mathbf{v}_1^T$ denotes the set of visible vectors from time steps 1 to T i.e. $\mathbf{v}^1$ to $\mathbf{v}^T$. The notation for latent vectors $\mathbf{h}$ is similar. $\mathbf{h}^{(c)}$ denotes the latent vector in the past time step that is connected to the current latent vector through a connection of type c. To simplify notation, we will denote the past connections $\{\mathbf{h}^{(1)}, ..., \mathbf{h}^{(C)}\}$ by $history^t$. The conditional distribution of the RBM at each time step is given by:

$$p(\mathbf{v}^t, \mathbf{h}^t | history^t) = (1/Z) exp(\Sigma_i a_i v_i^t + \Sigma_{i,j} v_i^t h_j^t w_{ij} + \Sigma_j (b_j + \Sigma_{c,l} w^{(c)}_{HH_{lj}} h_l^{(c)}) h_j^t)$$

where v_i^t and h_j^t denote the ith visible and jth latent variable respectively at time step t. $h_l^{(c)}$ denotes a latent variable in the past time step, and $w^{(c)}_{HH_{lj}}$ denotes the weight of the corresponding connection.

3.3 TRBM Likelihood and Inference

Section 3.1 describes an RBM where visible variables can take binary values. In our model, similar to (Salakhutdinov et al., 2007), we have multi-valued visible variables which we represent as one-hot binary vectors and model via a softmax distribution:

$$p(v_k^t = 1 | \mathbf{h}^t) = \frac{exp(a_k + \sum_j h_j^t w_{kj})}{\sum_i exp(a_i + \sum_j h_j^t w_{ij})} \qquad (3)$$

Latent variable inference is similar to equation 1 with an additional bias due to the temporal connections.

$$\begin{aligned} \mu_j^t &= p(h_j^t = 1 | \mathbf{v}^t, history^t) \\ &= \langle \sigma(b_j + \Sigma_{c,l} w^{(c)}_{HH_{lj}} h_l^{(c)} + \Sigma_i v_i^t w_{ij}) \rangle \\ &\approx \sigma(b_j^{'} + \Sigma_i v_i^t w_{ij}), \qquad (4) \\ b_j^{'} &= b_j + \Sigma_{c,l} w^{(c)}_{HH_{lj}} \mu_l^{(c)}. \end{aligned}$$

Here, μ denotes the mean of the corresponding latent variable. To keep inference tractable, we do not do any backward reasoning across directed connections to update $\mu^{(c)}$. Thus, the inference procedure for latent variables takes into account both the parse history and the current observation, but no future observations.

The limited set of possible values for the visible layer makes it possible to marginalize out latent variables in linear time to compute the exact likelihood. Let $\mathbf{v}^t(k)$ denote a vector with $v_k^t = 1$ and $v_{i(i \neq k)}^t = 0$. The conditional probability of a sub-decision is:

$$\begin{aligned} p(\mathbf{v}^t(k) | history^t) &= (1/Z) \Sigma_{\mathbf{h}^t} e^{-E(\mathbf{v}^t(k), \mathbf{h}^t)} \qquad (5) \\ &= (1/Z) e^{a_k} \Pi_j (1 + e^{b_j^{'} + w_{kj}}), \end{aligned}$$

where $Z = \Sigma_{i \in visible} e^{a_i} \Pi_{j \in \text{latent}} (1 + e^{b_j^{'} + w_{ij}})$.

We actually perform this calculation once for each sub-decision, ignoring the future sub-decisions in that time step. This is a slight approximation, but avoids having to compute the partition function over all possible combinations of values for all sub-decisions.[1]

The complete probability of a derivation is:

$$p(\mathbf{v}_1^T) = p(\mathbf{v}^1).p(\mathbf{v}^2 | history^2)...p(\mathbf{v}^T | history^T)$$

3.4 TRBM Training

The gradient of an RBM is given by:

$$\partial \log p(\mathbf{v}) / \partial w_{ij} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \qquad (6)$$

where $\langle \rangle_d$ denotes the expectation under distribution d. In general, computing the exact gradient is intractable and previous work proposed a Contrastive Divergence (CD) based learning procedure that approximates the above gradient using only *one step reconstruction* (Hinton, 2002). Fortunately, our model has only a limited set of possible visible values, which allows us to use a better approximation by taking the derivative of equation 5:

[1] In cases where computing the partition function is still not feasible (for instance, because of a large vocabulary), sampling methods could be used. However, we did not find this to be necessary.

$$\frac{\partial \log p(\mathbf{v}^t(k)|history^t)}{\partial w_{ij}} = \\ (\delta_{ki} - p(\mathbf{v}^t(i)|history^t)) \;\; \sigma(b_j^{'} + w_{ij}) \qquad (7)$$

Further, the weights on the temporal connections are learned by back-propagating the likelihood gradients through the directed links between steps. The back-proped gradient from future time steps is also used to train the current RBM weights. This back-propagation is similar to the Recurrent TRBM model of Sutskever et al. (2008). However, unlike their model, we do not use CD at each step to compute gradients.

3.5 Prediction

We use the same beam-search decoding strategy as used in (Titov and Henderson, 2007b). Given a derivation prefix, its partial parse tree and associated TRBM, the decoder adds a step to the TRBM for calculating the probabilities of hypothesized next decisions using equation 5. If the decoder selects a decision for addition to the candidate list, then the current step's latent variable means are inferred using equation 4, given that the chosen decision is now visible. These means are then stored with the new candidate for use in subsequent TRBM calculations.

4 Experiments & Results

We used syntactic dependencies from the English section of the CoNLL 2009 shared task dataset (Hajič et al., 2009). Standard splits of training, development and test sets were used. To handle word sparsity, we replaced all the *(POS, word)* pairs with frequency less than 20 in the training set with *(POS, UNKNOWN)*, giving us only 4530 tag-word pairs. Since our model can work only with projective trees, we used MaltParser (Nivre et al., 2006a) to projectivize/deprojectivize the training input/test output.

4.1 Results

Table 1 lists the labeled (LAS) and unlabeled (UAS) attachment scores. Row a shows that a simple ISBN model without features, using feed forward inference procedure, does not work well. As explained in section 2, this is expected since in the absence of explicit features, the latent variables in a given layer do not take into account the observations in the previous layers. The huge improvement in performance on adding the features (row b) shows that the feed forward inference procedure for ISBNs relies heavily on these feature connections to compensate for the lack of backward inference.

	Model	**LAS**	**UAS**
$a.$	ISBN w/o features	38.38	54.52
$b.$	ISBN w/ features	88.65	91.44
$c.$	TRBM w/o features	86.01	89.78
$d.$	TRBM w/ features	88.72	91.65
$e.$	MST (McDonald et al., 2005)	87.07	89.95
$f.$	Malt$_{\overrightarrow{\text{AE}}}$ (Hall et al., 2007)	85.96	88.64
$g.$	MST$_{\text{Malt}}$ (Nivre and McDonald, 2008)	87.45	90.22
$h.$	CoNLL 2008 #1 (Johansson and Nugues, 2008)	90.13	92.45
$i.$	ensemble$^{3}_{100\%}$ (Surdeanu and Manning, 2010)	88.83	91.47
$j.$	CoNLL 2009 #1 (Bohnet, 2009)	89.88	unknown

Table 1: LAS and UAS for different models.

The TRBM model avoids this problem as the inference procedure takes into account the current observation, which makes the latent variables much more informed. However, as row c shows, the TRBM model without features falls a bit short of the ISBN performance, indicating that features are indeed a powerful substitute for backward inference in sequential latent variable models. TRBM models would still be preferred in cases where such feature engineering is difficult or expensive, or where the objective is to compute the latent features themselves. For a fair comparison, we add the same set of features to the TRBM model (row d) and the performance improves by about 2% to reach the same level (non-significantly better) as ISBN with features. The improved inference in TRBM does however come at the cost of increased training and testing time. Keeping the same likelihood convergence criteria, we could train the ISBN in about 2 days and TRBM in about 5 days on a 3.3 GHz Xeon processor. With the same beam search parameters, the test time was about 1.5 hours for ISBN and about 4.5 hours for TRBM. Although more code optimization is possible, this trend is likely to remain.

We also tried a Contrastive Divergence based training procedure for TRBM instead of equation 7, but that resulted in about an absolute 10% lower LAS. Further, we also tried a very simple model without latent variables where temporal connections are between decision variables themselves. This model gave an LAS of only 60.46%, which indicates that without latent variables, it is very difficult to capture the parse history.

For comparison, we also include the performance numbers for some state-of-the-art dependency parsing systems. Surdeanu and Manning (2010) compare different parsing models using CoNLL 2008 shared task dataset (Surdeanu et al., 2008), which is the same as our dataset. Rows $e - i$ show the performance numbers of some systems as mentioned in their paper. Row j shows the best syntactic model in CoNLL 2009 shared task. The TRBM model has only 1.4% lower LAS and 0.8% lower UAS compared to the best performing model.

4.2 Latent Layer Analysis

We analyzed the latent layers in our models to see if they captured semantic patterns. A latent layer is a vector of 100 latent variables. Every *Shift* operation gives a latent representation for the corresponding word. We took all the verbs in the development set[2] and partitioned their representations into 50 clusters using the k-means algorithm. Table 2 shows some partitions for the TRBM model. The partitions look semantically meaningful but to get a quantitative analysis, we computed pairwise semantic similarity between all word pairs in a given cluster and aggregated this number over all the clusters. The semantic similarity was calculated using two different similarity measures on the wordnet corpus (Miller et al., 1990): *path* and *lin*. *path* similarity is a score between 0 and 1, equal to the inverse of the shortest path length between the two word senses. *lin* similarity (Lin, 1998) is a score between 0 and 1 based on the *Information Content* of the two word senses and of the Least Common Subsumer. Table 3 shows the similarity scores.[3] We observe that TRBM latent representations give a slightly better clustering than ISBN models. Again, this is because of the fact that the inference procedure in TRBMs takes into account the current observation. However, at the same time, the similarity numbers for ISBN with features are not very low, which shows that features are a powerful way to compensate for the lack of backward inference. This is in agreement with their good performance on the parsing task.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
says	needed	pressing	renewing
contends	expected	bridging	cause
adds	encouraged	curing	repeat
insists	allowed	skirting	broken
remarked	thought	tightening	extended

Table 2: K-means clustering of words according to their TRBM latent representations. Duplicate words in the same cluster are not shown.

Model	path	lin
ISBN w/o features	0.228	0.381
ISBN w/features	0.366	0.466
TRBM w/o features	0.386	0.487
TRBM w/ features	0.390	0.489

Table 3: Wordnet similarity scores for clusters given by different models.

5 Conclusions & Future Work

We have presented a Temporal Restricted Boltzmann Machines based model for dependency parsing. The model shows how undirected graphical models can be used to generate latent representations of local parsing actions, which can then be used as features for later decisions.

The TRBM model for dependency parsing could be extended to a Deep Belief Network by adding one more latent layer on top of the existing one (Hinton et al., 2006). Furthermore, as done for unlabeled images (Hinton et al., 2006), one could learn high-dimensional features from unlabeled text, which could then be used to aid parsing. Parser latent representations could also help other tasks such as Semantic Role Labeling (Henderson et al., 2008).

A free distribution of our implementation is available at `http://cui.unige.ch/~garg`.

Acknowledgments

This work was partly funded by Swiss NSF grant 200021_125137 and European Community FP7 grant 216594 (CLASSiC, www.classic-project.org).

[2] Verbs are words corresponding to POS tags: VB, VBD, VBG, VBN, VBP, VBZ. We selected verbs as they have good coverage in Wordnet.

[3] To account for randomness in k-means clustering, the clustering was performed 10 times with random initializations, similarity scores were computed for each run and a mean was taken.

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

B. Bohnet. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 67–72. Association for Computational Linguistics.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M.A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

J. Hall, J. Nilsson, J. Nivre, G. Eryigit, B. Megyesi, M. Nilsson, and M. Saers. 2007. Single malt or blended? A study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939. Association for Computational Linguistics.

J. Henderson, P. Merlo, G. Musillo, and I. Titov. 2008. A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 178–182. Association for Computational Linguistics.

G.E. Hinton, S. Osindero, and Y.W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

G.E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

R. Johansson and P. Nugues. 2008. Dependency-based syntactic-semantic analysis with PropBank and NomBank. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187. Association for Computational Linguistics.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296–304.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.

G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Introduction to wordnet: An on-line lexical database. *International Journal of lexicography*, 3(4):235.

A. Mnih and G. Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM.

J. Nivre and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. *Proceedings of ACL-08: HLT*, pages 950–958.

J. Nivre, J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of CoNLL*, pages 49–56.

J. Nivre, J. Hall, and J. Nilsson. 2006a. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6.

J. Nivre, J. Hall, J. Nilsson, G. Eryiit, and S. Marinov. 2006b. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 221–225. Association for Computational Linguistics.

A. Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1):151–175.

R. Salakhutdinov and G. Hinton. 2009. Replicated softmax: an undirected topic model. *Advances in Neural Information Processing Systems*, 22.

R. Salakhutdinov, A. Mnih, and G. Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, page 798. ACM.

M. Surdeanu and C.D. Manning. 2010. Ensemble models for dependency parsing: cheap and good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 649–652. Association for Computational Linguistics.

M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177. Association for Computational Linguistics.

I. Sutskever, G. Hinton, and G. Taylor. 2008. The recurrent temporal restricted boltzmann machine. In *NIPS*, volume 21, page 2008.

G.W. Taylor, G.E. Hinton, and S.T. Roweis. 2007. Modeling human motion using binary latent variables. *Advances in neural information processing systems*, 19:1345.

I. Titov and J. Henderson. 2007a. Constituent parsing with incremental sigmoid belief networks. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, volume 45, page 632.

I. Titov and J. Henderson. 2007b. Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 947–951.