Semi-supervised Convex Training for Dependency Parsing

Qin Iris Wang
Department of Computing Science
University of Alberta
Edmonton, AB, Canada, T6G 2E8
wqin@cs.ualberta.ca

Dale Schuurmans
Department of Computing Science
University of Alberta
Edmonton, AB, Canada, T6G 2E8
dale@cs.ualberta.ca

Dekang Lin
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA, USA, 94043
lindek@google.com

Abstract

We present a novel semi-supervised training algorithm for learning dependency parsers. By combining a supervised large margin loss with an unsupervised least squares loss, a discriminative, convex, semi-supervised learning algorithm can be obtained that is applicable to large-scale problems. To demonstrate the benefits of this approach, we apply the technique to learning dependency parsers from combined labeled and unlabeled corpora. Using a stochastic gradient descent algorithm, a parsing model can be efficiently learned from semi-supervised data that significantly outperforms corresponding supervised methods.

1 Introduction

Supervised learning algorithms still represent the state of the art approach for inferring dependency parsers from data (McDonald et al., 2005a; McDonald and Pereira, 2006; Wang et al., 2007). However, a key drawback of supervised training algorithms is their dependence on labeled data, which is usually very difficult to obtain. Perceiving the limitation of supervised learning—in particular, the heavy dependence on annotated corpora—many researchers have investigated *semi-supervised* learning techniques that can take both labeled and unlabeled training data as input. Following the common theme of "more data is better data" we also use both a limited labeled corpora and a plentiful unlabeled data resource. Our goal is to obtain better performance than a purely supervised approach without unreasonable computational effort. Unfortunately, although significant recent progress has been made in the area of semi-supervised learning, the performance of semi-supervised learning algorithms still fall far short of expectations, particularly in challenging real-world tasks such as natural language parsing or machine translation.

A large number of distinct approaches to semi-supervised training algorithms have been investigated in the literature (Bennett and Demiriz, 1998; Zhu et al., 2003; Altun et al., 2005; Mann and McCallum, 2007). Among the most prominent approaches are self-training, generative models, semi-supervised support vector machines (S3VM), graph-based algorithms and multi-view algorithms (Zhu, 2005).

Self-training is a commonly used technique for semi-supervised learning that has been ap-

plied to several natural language processing tasks (Yarowsky, 1995; Charniak, 1997; Steedman et al., 2003). The basic idea is to bootstrap a supervised learning algorithm by alternating between inferring the missing label information and retraining. Recently, McClosky et al. (2006a) successfully applied self-training to parsing by exploiting available unlabeled data, and obtained remarkable results when the same technique was applied to parser adaptation (McClosky et al., 2006b). More recently, Haffari and Sarkar (2007) have extended the work of Abney (2004) and given a better mathematical understanding of self-training algorithms. They also show connections between these algorithms and other related machine learning algorithms.

Another approach, generative probabilistic models, are a well-studied framework that can be extremely effective. However, generative models use the EM algorithm for parameter estimation in the presence of missing labels, which is notoriously prone to getting stuck in poor local optima. Moreover, EM optimizes a marginal likelihood score that is not discriminative. Consequently, most previous work that has attempted semi-supervised or unsupervised approaches to parsing have not produced results beyond the state of the art supervised results (Klein and Manning, 2002; Klein and Manning, 2004). Subsequently, alternative estimation strategies for unsupervised learning have been proposed, such as *Contrastive Estimation* (CE) by Smith and Eisner (2005). Contrastive Estimation is a generalization of EM, by defining a notion of learner guidance. It makes use of a set of examples (its *neighborhood*) that are similar in some way to an observed example, requiring the learner to move probability mass to a given example, taking only from the example's neighborhood. Nevertheless, CE still suffers from shortcomings, including local minima.

In recent years, SVMs have demonstrated state of the art results in many supervised learning tasks. As a result, many researchers have put effort on developing algorithms for semi-supervised SVMs (S3VMs) (Bennett and Demiriz, 1998; Altun et al., 2005). However, the standard objective of an S3VM is non-convex on the unlabeled data, thus requiring sophisticated global optimization heuristics to obtain reasonable solutions. A number of researchers have proposed several efficient approximation algorithms for S3VMs (Bennett and Demiriz, 1998; Chapelle and Zien, 2005; Xu and Schuurmans, 2005). For example, Chapelle and Zien (2005) propose an algorithm that smoothes the objective with a Gaussian function, and then performs a gradient descent search in the primal space to achieve a local solution. An alternative approach is proposed by Xu and Schuurmans (2005) who formulate a semi-definite programming (SDP) approach. In particular, they present an algorithm for multiclass unsupervised and semi-supervised SVM learning, which relaxes the original non-convex objective into a close convex approximation, thereby allowing a global solution to be obtained. However, the computational cost of SDP is still quite expensive.

Instead of devising various techniques for coping with non-convex loss functions, we approach the problem from a different perspective. We simply replace the non-convex loss on unlabeled data with an alternative loss that is jointly convex with respect to both the model parameters and (the encoding of) the self-trained prediction targets. More specifically, for the loss on the unlabeled data part, we substitute the original unsupervised structured SVM loss with a least squares loss, but keep constraints on the inferred prediction targets, which avoids trivialization. Although using a least squares loss function for classification appears misguided, there is a precedent for just this approach in the early pattern recognition literature (Duda et al., 2000). This loss function has the advantage that the entire training objective on both the labeled and unlabeled data now becomes convex, since it consists of a convex structured large margin loss on labeled data and a convex least squares loss on unlabeled data. As we will demonstrate below, this approach admits an efficient training procedure that can find a global minimum, and, perhaps surprisingly, can systematically improve the accuracy of supervised training approaches for learning dependency parsers.

Thus, in this paper, we focus on *semi-supervised* language learning, where we can make use of both labeled and unlabeled data. In particular, we investigate a semi-supervised approach for structured large margin training, where the objective is a combination of two convex functions, the structured large margin loss on labeled data and the least squares loss on unlabeled data. We apply the result-

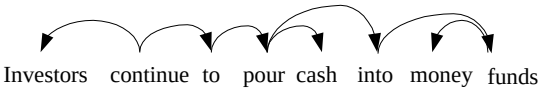

Figure 1: A dependency tree

ing semi-supervised convex objective to dependency parsing, and obtain significant improvement over the corresponding supervised structured SVM. Note that our approach is different from the self-training technique proposed in (McClosky et al., 2006a), although both methods belong to semi-supervised training category.

In the remainder of this paper, we first review the supervised structured large margin training technique. Then we introduce the standard semi-supervised structured large margin objective, which is non-convex and difficult to optimize. Next we present a new semi-supervised training algorithm for structured SVMs which is convex optimization. Finally, we apply this algorithm to dependency parsing and show improved dependency parsing accuracy for both Chinese and English.

2 Dependency Parsing Model

Given a sentence $X = (x_1, ..., x_n)$ (x_i denotes each word in the sentence), we are interested in computing a directed dependency tree, Y, over X. As shown in Figure 1, in a dependency structure, the basic units of a sentence are the syntactic relationships (aka. head-child or governor-dependent or regent-subordinate relations) between two individual words, where the relationships are expressed by drawing links connecting individual words (Manning and Schutze, 1999). The direction of each link points from a head word to a child word, and each word has one and only one head, except for the head of the sentence. Thus a dependency structure is actually a rooted, directed tree. We assume that a directed dependency tree Y consists of ordered pairs $(x_i \rightarrow x_j)$ of words in X such that each word appears in at least one pair and each word has in-degree at most one. Dependency trees are assumed to be projective here, which means that if there is an arc $(x_i \rightarrow x_j)$, then x_i is an ancestor of all the words between x_i and x_j.[1] Let $\Phi(X)$ denote the set of all the directed, projective trees that span on X. The parser's goal is then to find the most preferred parse; that is, a projective tree, $Y \in \Phi(X)$, that obtains the highest "score". In particular, one would assume that the score of a complete spanning tree Y for a given sentence, whether probabilistically motivated or not, can be decomposed as a sum of local scores for each link (a word pair) (Eisner, 1996; Eisner and Satta, 1999; McDonald et al., 2005a). Given this assumption, the parsing problem reduces to find

$$\begin{aligned} Y^* &= \arg \max_{Y \in \Phi(X)} \textit{score}(Y|X) \qquad (1) \\ &= \arg \max_{Y \in \Phi(X)} \sum_{(x_i \rightarrow x_j) \in Y} \textit{score}(x_i \rightarrow x_j) \end{aligned}$$

where the $\textit{score}(x_i \rightarrow x_j)$ can depend on any measurable property of x_i and x_j within the sentence X. This formulation is sufficiently general to capture most dependency parsing models, including probabilistic dependency models (Eisner, 1996; Wang et al., 2005) as well as non-probabilistic models (McDonald et al., 2005a).

For standard scoring functions, particularly those used in non-generative models, one further assumes that the score of each link in (1) can be decomposed into a weighted linear combination of features

$$\textit{score}(x_i \rightarrow x_j) = \boldsymbol{\theta} \cdot \mathbf{f}(x_i \rightarrow x_j) \qquad (2)$$

where $\mathbf{f}(x_i \rightarrow x_j)$ is a feature vector for the link $(x_i \rightarrow x_j)$, and $\boldsymbol{\theta}$ are the weight parameters to be estimated during training.

3 Supervised Structured Large Margin Training

Supervised structured large margin training approaches have been applied to parsing and produce promising results (Taskar et al., 2004; McDonald et al., 2005a; Wang et al., 2006). In particular, structured large margin training can be expressed as minimizing a regularized loss (Hastie et al., 2004), as shown below:

[1] We assume all the dependency trees are projective in our work (just as some other researchers do), although in the real word, most languages are non-projective.

$$\min_{\boldsymbol{\theta}} \quad \frac{\beta}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \sum_i \max_{L_{i,k}} (\Delta(L_{i,k}, Y_i) - \textit{diff}(\boldsymbol{\theta}, Y_i, L_{i,k})) \tag{3}$$

where Y_i is the target tree for sentence X_i; $L_{i,k}$ ranges over all possible alternative k trees in $\Phi(X_i)$; $\textit{diff}(\boldsymbol{\theta}, Y_i, L_{i,k}) = \textit{score}(\boldsymbol{\theta}, Y_i) - \textit{score}(\boldsymbol{\theta}, L_{i,k})$; $\textit{score}(\boldsymbol{\theta}, Y_i) = \sum_{(x_m \to x_n) \in Y_i} \boldsymbol{\theta} \cdot \mathbf{f}(x_m \to x_n)$, as shown in Section 2; and $\Delta(L_{i,k}, Y_i)$ is a measure of distance between the two trees $L_{i,k}$ and Y_i. This is an application of the structured large margin training approach first proposed in (Taskar et al., 2003) and (Tsochantaridis et al., 2004).

Using the techniques of Hastie et al. (2004) one can show that minimizing the objective (3) is equivalent to solving the quadratic program

$$\begin{aligned} \min_{\boldsymbol{\theta}, \boldsymbol{\xi}} \quad & \frac{\beta}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \mathbf{e}^\top\boldsymbol{\xi} \quad \text{subject to} \\ & \xi_{i,k} \geq \Delta(L_{i,k}, Y_i) - \textit{diff}(\boldsymbol{\theta}, Y_i, L_{i,k}) \\ & \xi_{i,k} \geq 0 \\ & \text{for all } i, L_{i,k} \in \Phi(X_i) \end{aligned} \tag{4}$$

where $\mathbf{e}$ denotes the vector of all 1's and $\boldsymbol{\xi}$ represents slack variables. This approach corresponds to the training problem posed in (McDonald et al., 2005a) and has yielded the best published results for English dependency parsing.

To compare with the new semi-supervised approach we will present in Section 5 below, we re-implemented the supervised structured large margin training approach in the experiments in Section 7. More specifically, we solve the following quadratic program, which is based on Equation (3)

$$\min_{\boldsymbol{\theta}} \ \frac{\alpha}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \sum_i \max_L \sum_{m=1}^{k}\sum_{n=1}^{k} \Delta(L_{i,m,n}, Y_{i,m,n}) - \textit{diff}(\boldsymbol{\theta}, Y_{i,m,n}, L_{i,m,n}) \tag{5}$$

where $\textit{diff}(\boldsymbol{\theta}, Y_{i,m,n}, L_{i,m,n}) = \textit{score}(\boldsymbol{\theta}, Y_{i,m,n}) - \textit{score}(\boldsymbol{\theta}, L_{i,m,n})$ and k is the sentence length. We represent a dependency tree as a $k \times k$ adjacency matrix. In the adjacency matrix, the value of $Y_{i,m,n}$ is 1 if the word m is the head of the word n, 0 otherwise. Since both the distance function $\Delta(L_i, Y_i)$ and the score function decompose over links, solving (5) is equivalent to solve the original constrained quadratic program shown in (4).

4 Semi-supervised Structured Large Margin Objective

The objective of standard semi-supervised structured SVM is a combination of structured large margin losses on both labeled and unlabeled data. It has the following form:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \ & \frac{\alpha}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \sum_{i=1}^{N} \textit{structured\_loss}\,(\boldsymbol{\theta}, X_i, Y_i) \\ & + \min_{\mathbf{Y}_j} \sum_{j=1}^{U} \textit{structured\_loss}\,(\boldsymbol{\theta}, X_j, Y_j) \end{aligned} \tag{6}$$

where

$$\begin{aligned} & \textit{structured\_loss}\,(\boldsymbol{\theta}, X_i, Y_i) \\ = \ & \max_L \sum_{m=1}^{k}\sum_{n=1}^{k} \Delta(L_{i,m,n}, Y_{i,m,n}) \\ & \qquad - \textit{diff}(\boldsymbol{\theta}, Y_{i,m,n}, L_{i,m,n}) \end{aligned} \tag{7}$$

N and U are the number of labeled and unlabeled training sentences respectively.

In the second term of the above objective shown in (6), both $\boldsymbol{\theta}$ and $\mathbf{Y}_j$ are variables. The resulting loss function has a hat shape (usually called hat-loss), which is non-convex. Therefore the objective as a whole is non-convex, making the search for global optimal difficult. Note that the root of the optimization difficulty for S3VMs is the non-convex property of the second term in the objective function. We will propose a novel approach which can deal with this problem. We introduce an efficient approximation—least squares loss—for the structured large margin loss on unlabeled data below.

5 Semi-supervised Convex Training for Structured SVM

Although semi-supervised structured SVM learning has been an active research area, semi-supervised structured SVMs have not been used in many real applications to date. The main reason is that most available semi-supervised large margin learning approaches are non-convex or computationally expensive (e.g. (Xu and Schuurmans, 2005)). These techniques are difficult to implement and extremely hard to scale up. We present a semi-supervised algorithm for structured large margin training, whose objective

is a combination of two convex terms: the supervised structured large margin loss on labeled data and the cheap least squares loss on unlabeled data. The combined objective is still convex, easy to optimize and much cheaper to implement.

5.1 Least Squares Convex Objective

Before we introduce the new algorithm, we first introduce a convex loss which we apply it to unlabeled training data for the semi-supervised structured large margin objective which we will introduce in Section 5.2 below. More specifically, we use a *structured* least squares loss to approximate the structured large margin loss on unlabeled data. The corresponding objective is:

$$\min_{\boldsymbol{\theta},\mathbf{Y}_j} \ \frac{\alpha}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \frac{\lambda}{2}\sum_{j=1}^{U}\sum_{m=1}^{k}\sum_{n=1}^{k}\left(\boldsymbol{\theta}^\top\mathbf{f}(X_{j,m}\rightarrow X_{j,n}) - Y_{j,m,n}\right)^2 \tag{8}$$

subject to constraints on $\mathbf{Y}$ (explained below).

The idea behind this objective is that for each possible link $(X_{j,m} \rightarrow X_{j,n})$, we intend to minimize the difference between the link and the corresponding estimated link based on the learned weight vector. Since this is conducted on unlabeled data, we need to estimate both $\boldsymbol{\theta}$ and $\mathbf{Y}_j$ to solve the optimization problem. As mentioned in Section 3, a dependency tree $\mathbf{Y}_j$ is represented as an adjacency matrix. Thus we need to enforce some constraints in the adjacency matrix to make sure that each $\mathbf{Y}_j$ satisfies the dependency tree constraints. These constraints are critical because they prevent (8) from having a trivial solution in $\mathbf{Y}$. More concretely, suppose we use rows to denote heads and columns to denote children. Then we have the following constraints on the adjacency matrix:

- (1) All entries in $\mathbf{Y}_j$ are between 0 and 1 (convex relaxation of discrete directed edge indicators);
- (2) The sum over all the entries on each column is equal to one (one-head rule);
- (3) All the entries on the diagonal are zeros (no self-link rule);
- (4) $Y_{j,m,n} + Y_{j,n,m} \leq 1$ (anti-symmetric rule), which enforces directedness.

One final constraint that is sufficient to ensure that a directed tree is obtained, is connectedness (i.e. acyclicity), which can be enforced with an additional semidefinite constraint. Although convex, this constraint is more expensive to enforce, therefore we drop it in our experiments below. (However, adding the semidefinite connectedness constraint appears to be feasible on a sentence by sentence level.)

Critically, the objective (8) is *jointly* convex in both the weights $\boldsymbol{\theta}$ and the edge indicator variables $\mathbf{Y}$. This means, for example, that there are no local minima in (8)—*any* iterative improvement strategy, if it converges at all, must converge to a global minimum.

5.2 Semi-supervised Convex Objective

By combining the convex structured SVM loss on labeled data (shown in Equation (5)) and the convex least squares loss on unlabeled data (shown in Equation (8)), we obtain a semi-supervised structured large margin loss

$$\min_{\boldsymbol{\theta},\mathbf{Y}_j} \ \frac{\alpha}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \sum_{i=1}^{N}\textit{structured\_loss}\left(\boldsymbol{\theta}, X_i, Y_i\right) + \sum_{j=1}^{U}\textit{least\_squares\_loss}\left(\boldsymbol{\theta}, X_j, Y_j\right) \tag{9}$$

subject to constraints on $\mathbf{Y}$ (explained above).

Since the summation of two convex functions is also convex, so is (9). Replacing the two losses with the terms shown in Equation (5) and Equation (8), we obtain the final convex objective as follows:

$$\min_{\boldsymbol{\theta},\mathbf{Y}_j} \ \frac{\alpha}{2N}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \sum_{i=1}^{N}\max_{L}\sum_{m=1}^{k}\sum_{n=1}^{k}\Delta(L_{i,m,n}, Y_{i,m,n}) - \textit{diff}(\boldsymbol{\theta}, Y_{i,m,n}, L_{i,m,n}) + \frac{\alpha}{2U}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \frac{\lambda}{2}\sum_{j=1}^{U}\sum_{m=1}^{k}\sum_{n=1}^{k}\left(\boldsymbol{\theta}^\top\mathbf{f}(X_{j,m}\rightarrow X_{j,n}) - Y_{j,m,n}\right)^2 \tag{10}$$

subject to constraints on $\mathbf{Y}$ (explained above), where $\textit{diff}(\boldsymbol{\theta}, Y_{i,m,n}, L_{i,m,n}) = \textit{score}(\boldsymbol{\theta}, Y_{i,m,n}) -$

score$(\boldsymbol{\theta}, L_{i,m,n})$, N and U are the number of labeled and unlabeled training sentences respectively, as we mentioned before. Note that in (10) we have split the regularizer into two parts; one for the supervised component of the objective, and the other for the unsupervised component. Thus the semi-supervised convex objective is regularized proportionally to the number of labeled and unlabeled training sentences.

6 Efficient Optimization Strategy

To solve the convex optimization problem shown in Equation (10), we used a gradient descent approach which simply uses stochastic gradient steps. The procedure is as follows.

- Step 0, initialize the $\mathbf{Y}_j$ variables of each unlabeled sentence as a right-branching (left-headed) chain model, i.e. the head of each word is its left neighbor.
- Step 1, pass through all the labeled training sentences one by one. The parameters $\boldsymbol{\theta}$ are updated based on each labeled sentence.
- Step 2, based on the learned parameter weights from the labeled data, update $\boldsymbol{\theta}$ and $\mathbf{Y}_j$ on each unlabeled sentence alternatively:
  - treat $\mathbf{Y}_j$ as a constant, update $\boldsymbol{\theta}$ on each unlabeled sentence by taking a local gradient step;
  - treat $\boldsymbol{\theta}$ as a constant, update $\mathbf{Y}_j$ by calling the optimization software package CPLEX to solve for an optimal local solution.
- Repeat the procedure of step 1 and step 2 until maximum iteration number has reached.

This procedure works efficiently on the task of training a dependency parser. Although $\boldsymbol{\theta}$ and $\mathbf{Y}_j$ are updated locally on each sentence, progress in minimizing the total objective shown in Equation (10) is made in each iteration. In our experiments, the objective usually converges within 30 iterations.

7 Experimental Results

Given a convex approach to semi-supervised structured large margin training, and an efficient training algorithm for achieving a global optimum, we now investigate its effectiveness for dependency parsing. In particular, we investigate the accuracy of the results it produces. We applied the resulting algorithm to learn dependency parsers for both English and Chinese.

7.1 Experimental Design

Data Sets

Since we use a semi-supervised approach, both labeled and unlabeled training data are needed. For experiment on English, we used the English Penn Treebank (PTB) (Marcus et al., 1993) and the constituency structures were converted to dependency trees using the same rules as (Yamada and Matsumoto, 2003). The standard training set of PTB was spit into 2 parts: labeled training data—the first $30k$ sentences in section 2-21, and unlabeled training data—the remaining sentences in section 2-21. For Chinese, we experimented on the Penn Chinese Treebank 4.0 (CTB4) (Palmer *et al.*, 2004) and we used the rules in (Bikel, 2004) for conversion. We also divided the standard training set into 2 parts: sentences in section 400-931 and sentences in section 1-270 are used as labeled and unlabeled data respectively. For both English and Chinese, we adopted the standard development and test sets throughout the literature.

As listed in Table 1 with greater detail, we experimented with sets of data with different sentence length: PTB-10/CTB4-10, PTB-15/CTB4-15, PTB-20/CTB4-20, CTB4-40 and CTB4, which contain sentences with up to 10, 15, 20, 40 and all words respectively.

Features

For simplicity, in current work, we only used two sets of features—word-pair and tag-pair indicator features, which are a subset of features used by other researchers on dependency parsing (McDonald et al., 2005a; Wang et al., 2007). Although our algorithms can take arbitrary features, by only using these simple features, we already obtained very promising results on dependency parsing using both the supervised and semi-supervised approaches. Using the full set of features described in (McDonald et al., 2005a; Wang et al., 2007) and comparing the corresponding dependency parsing

English	PTB-10	Training(l/ul)	3026/1016
		Dev	163
		Test	270
	PTB-15	Training	7303/2370
		Dev	421
		Test	603
	PTB-20	Training	12519/4003
		Dev	725
		Test	1034
Chinese	CTB4-10	Training(l/ul)	642/347
		Dev	61
		Test	40
	CTB4-15	Training	1262/727
		Dev	112
		Test	83
	CTB4-20	Training	2038/1150
		Dev	163
		Test	118
	CTB4-40	Training	4400/2452
		Dev	274
		Test	240
	CTB4	Training	5314/2977
		Dev	300
		Test	289

Table 1: Size of Experimental Data (# of sentences)

Training	Test length	Supervised	Semi-sup
Train-10	≤ 10	82.98	**84.50**
Train-15	≤ 10	84.80	**86.93**
	≤ 15	76.96	**80.79**
Train-20	≤ 10	84.50	**86.32**
	≤ 15	78.77	**80.57**
	≤ 20	74.89	**77.85**
Train-40	≤ 10	84.19	**85.71**
	≤ 15	78.03	**81.21**
	≤ 20	76.25	**77.79**
	≤ 40	68.17	**70.90**
Train-all	≤ 10	82.67	**84.80**
	≤ 15	77.92	**79.30**
	≤ 20	77.30	77.24
	≤ 40	70.11	**71.90**
	all	66.30	**67.35**

Table 2: Supervised and Semi-supervised Dependency Parsing Accuracy on Chinese (%)

results with previous work remains a direction for future work.

Dependency Parsing Algorithms

For simplicity of implementation, we use a standard CKY parser in the experiments, although Eisner's algorithm (Eisner, 1996) and the Spanning Tree algorithm (McDonald et al., 2005b) are also applicable.

7.2 Results

We evaluate parsing accuracy by comparing the directed dependency links in the parser output against the directed links in the treebank. The parameters α and λ which appear in Equation (10) were tuned on the development set. Note that, during training, we only used the raw sentences of the unlabeled data. As shown in Table 2 and Table 3, for each data set, the semi-supervised approach achieves a signifi cant improvement over the supervised one in dependency parsing accuracy on both Chinese and English. These positive results are somewhat surprising since a very simple loss function was used on

Training	Test length	Supervised	Semi-sup
Train-10	≤ 10	87.77	**89.17**
Train-15	≤ 10	88.06	**89.31**
	≤ 15	81.10	**83.37**
Train-20	≤ 10	88.78	**90.61**
	≤ 15	83.00	**83.87**
	≤ 20	77.70	**79.09**

Table 3: Supervised and Semi-supervised Dependency Parsing Accuracy on English (%)

the unlabeled data. A key benefit of the approach is that a straightforward training algorithm can be used to obtain global solutions. Note that the results of our model are not directly comparable with previous parsing results shown in (McClosky et al., 2006a), since the parsing accuracy is measured in terms of dependency relations while their results are f-score of the bracketings implied in the phrase structure.

8 Conclusion and Future Work

In this paper, we have presented a novel algorithm for semi-supervised structured large margin training. Unlike previous proposed approaches, we introduce a convex objective for the semi-supervised learning algorithm by combining a convex structured SVM loss and a convex least square loss. This new semi-supervised algorithm is much more computationally efficient and can easily scale up. We have proved our hypothesis by applying the algorithm to the significant task of dependency parsing. The experimental results show that the proposed semi-supervised large margin training algorithm outperforms the supervised one, without much additional computational cost.

There remain many directions for future work. One obvious direction is to use the whole Penn Treebank as labeled data and use some other unannotated data source as unlabeled data for semi-supervised training. Next, as we mentioned before, a much richer feature set can be used in our model to get better dependency parsing results. Another direction is to apply the semi-supervised algorithm to other natural language problems, such as machine translation, topic segmentation and chunking. In these areas, there are only limited annotated data available. Therefore semi-supervised approaches are necessary to achieve better performance. The proposed semi-supervised convex training approach can be easily applied to these tasks.

Acknowledgments

We thank the anonymous reviewers for their useful comments. Research is supported by the Alberta Ingenuity Center for Machine Learning, NSERC, MITACS, CFI and the Canada Research Chairs program. The first author was also funded by the Queen Elizabeth II Graduate Scholarship.

References

S. Abney. 2004. Understanding the yarowsky algorithm. *Computational Linguistics*, 30(3):365–395.

Y. Altun, D. McAllester, and M. Belkin. 2005. Maximum margin semi-supervised learning for structured variables. In *Proceedings of Advances in Neural Information Processing Systems 18*.

K. Bennett and A. Demiriz. 1998. Semi-supervised support vector machines. In *Proceedings of Advances in Neural Information Processing Systems 11*.

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

O. Chapelle and A. Zien. 2005. Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Inteligence and Statistics*.

E. Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 598–603.

R. Duda, P. Hart, and D. Stork. 2000. *Pattern Classification*. Wiley, second edition.

J. Eisner and G. Satta. 1999. Efficient parsing for bilexical context-free grammars and head-automaton grammars. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

J. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the International Conference on Computational Linguistics*.

G. Haffari and A. Sarkar. 2007. Analysis of semi-supervised learning with the yarowsky algorithm. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. 2004. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415.

D. Klein and C. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

D. Klein and C. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedingsof the Annual Meeting of the Association for Computational Linguistics*.

G. S. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of International Conference on Machine Learning*.

C. Manning and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

D. McClosky, E. Charniak, and M. Johnson. 2006a. Effective self-training for parsing. In *Proceedings of the Human Language Technology: the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

D. McClosky, E. Charniak, and M. Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics*.

R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of European Chapter of the Annual Meeting of the Association for Computational Linguistics*.

R. McDonald, K. Crammer, and F. Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technologies and Conference on Empirical Methods in Natural Language Processing*.

M. Palmer *et al.* 2004. *Chinese Treebank 4.0*. Linguistic Data Consortium.

N. Smith and J. Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the European Chapter of the Annual Meeting of the Association for Computational Linguistics*, pages 331–338.

B. Taskar, C. Guestrin, and D. Koller. 2003. Max-margin Markov networks. In *Proceedings of Advances in Neural Information Processing Systems 16*.

B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning. 2004. Max-margin parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of International Conference on Machine Learning*.

Q. Wang, D. Schuurmans, and D. Lin. 2005. Strictly lexical dependency parsing. In *Proceedings of the International Workshop on Parsing Technologies*, pages 152–159.

Q. Wang, C. Cherry, D. Lizotte, and D. Schuurmans. 2006. Improved large margin dependency parsing via local constraints and Laplacian regularization. In *Proceedings of The Conference on Computational Natural Language Learning*, pages 21–28.

Q. Wang, D. Lin, and D. Schuurmans. 2007. Simple training of dependency parsers via structured boosting. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1756–1762.

L. Xu and D. Schuurmans. 2005. Unsupervised and semi-supervised multi-class support vector machines. In *Proceedings the Association for the Advancement of Artificial Intelligence*.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the International Workshop on Parsing Technologies*.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts.

X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of International Conference on Machine Learning*.

X. Zhu. 2005. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison.