A Multilingual Dependency Analysis System using Online Passive-Aggressive Learning

Le-Minh Nguyen, Akira Shimazu, and Phuong-Thai Nguyen
Japan Advanced Institute of Science and Technology (JAIST)
Asahidai 1-1, Nomi, Ishikawa, 923-1292 Japan
{nguyenml,shimazu,thai}@jaist.ac.jp

Xuan-Hieu Phan
Tohoku University
Aobayama 6-3-09, Sendai, 980-8579, Japan
hieuxuan@ecei.tohoku.ac.jp

Abstract

This paper presents an online algorithm for dependency parsing problems. We propose an adaptation of the passive and aggressive online learning algorithm to the dependency parsing domain. We evaluate the proposed algorithms on the 2007 CONLL Shared Task, and report errors analysis. Experimental results show that the system score is better than the average score among the participating systems.

1 Introduction

Research on dependency parsing is mainly based on machine learning methods, which can be called history-based (Yamada and Matsumoto, 2003; Nivre et al., 2006) and discriminative learning methods (McDonald et al., 2005a; Corston-Oliver et al., 2006). The learning methods using in discriminative parsing are Perceptron (Collins, 2002) and online large-margin learning (MIRA) (Crammer and Singer, 2003).

The difference of MIRA-based parsing in comparison with history-based methods is that the MIRA-based parser were trained to maximize the accuracy of the overall tree. The MIRA based parsing is close to maximum-margin parsing as in Taskar et al. (2004) and Tsochantaridis et al. (2005) for parsing. However, unlike maximum-margin parsing, it is not limited to parsing sentences of 15 words or less due to computation time. The performance of MIRA based parsing achieves the state-of-the-art performance in English data (McDonald et al., 2005a; McDonald et al., 2006).

In this paper, we propose a new adaptation of online larger-margin learning to the problem of dependency parsing. Unlike the MIRA parser, our method does not need an optimization procedure in each learning update, but users only an update equation. This might lead to faster training time and easier implementation.

The contributions of this paper are two-fold: First, we present a training algorithm called PA learning for dependency parsing, which is as easy to implement as Perceptron, yet competitive with large margin methods. This algorithm has implications for anyone interested in implementing discriminative training methods for any application. Second, we evaluate the proposed algorithm on the multilingual data task as well as the domain adaptation task (Nivre et al., 2007).

The remaining parts of the paper are organized as follows: Section 2 proposes our dependency parsing with Passive-Aggressive learning. Section 3 discusses some experimental results and Section 4 gives conclusions and plans for future work.

2 Dependency Parsing with Passive-Aggressive Learning

This section presents the modification of Passive-Aggressive Learning (PA) (Crammer et al., 2006) for dependency parsing. We modify the PA algorithm to deal with structured prediction, in which our problem is to learn a discriminant function that maps an input sentence x to a dependency tree y. Figure 1 shows an example of dependency parsing which depicts the relation of each word to another word within a sentence. There are some algorithms

Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 1149–1155,
Prague, June 2007.

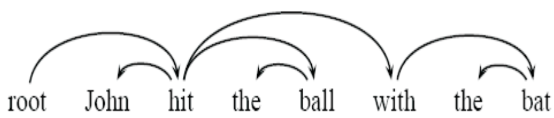

Figure 1: This is an example of dependency tree

to determine these relations of each word to another words, for instance, the modified CKY algorithm (Eisner, 1996) is used to define these relations for a given sentence.

2.1 Parsing Algorithm

Dependency-tree parsing as the search for the maximum spanning tree (MST) in a graph was proposed by McDonald et al. (2005b). In this subsection, we briefly describe the parsing algorithms based on the first-order MST parsing. Due to the limitation of participation time, we only applied the first-order decoding parsing algorithm in CONLL-2007. However, our algorithm can be used for the second order parsing.

Let the generic sentence be denoted by x ; the ith word of x is denoted by w_i. The generic dependency tree is denoted by y. If y is a dependency tree for sentence x, we write $(i, j) \in y$ to indicate that there is a directed edge from word xw_i to word xw_j in the tree, that is, xw_i is the parent of xw_j . $T = \{(x_t, y_t)\}_{t=1}^{n}$ denotes the training data. We follow the edge based factorization method of Eisner (Eisner, 1996) and define the score of a dependency tree as the sum of the score of all edges in the tree,

$$s(x, y) = \sum_{(i,j)\in y} s(i, j) = \sum_{(i,j)\in y} \mathbf{w} \cdot \Phi(i, j) \quad (1)$$

where $\Phi(i, j)$ is a high-dimensional binary feature representation of the edge from xw_i to xw_j . For example in Figure 1, we can present an example $\Phi(i, j)$ as follows;

$$\Phi(i, j) = \begin{cases} 1 \,\text{if}\, xw_i =' hit' \, and \, xw_j =' ball' \\ 0 \,\text{otherwise} \end{cases}$$

The basic question must be answered for models of this form: how to find the dependency tree y with the highest score for sentence x? The two algorithms we employed in our dependency parsing model are the Eisner parsing (Eisner, 1996) and Chu-Liu's algorithm (Chu and Liu, 1965). The algorithms are commonly used in other online-learning dependency parsing, such as in (McDonald et al., 2005a).

In the next subsection we will address the problem of how to estimate the weight w_i associated with a feature Φ_i in the training data using an online PA learning algorithm.

2.2 Online PA Learning

This section presents a modification of PA algorithm for structured prediction, and its use in dependency parsing. The Perceptron style for natural language processing problems as initially proposed by (Collins, 2002) can provide state of the art results on various domains including text chunking, syntactic parsing, etc. The main drawback of the Perceptron style algorithm is that it does not have a mechanism for attaining the maximize margin of the training data. It may be difficult to obtain high accuracy in dealing with hard learning data. The structured support vector machine (Tsochantaridis et al., 2005) and the maximize margin model (Taskar et al., 2004) can gain a maximize margin value for given training data by solving an optimization problem (i.e quadratic programming). It is obvious that using such an optimization algorithm requires much computational time. For dependency parsing domain, McDonald et al (2005a) modified the MIRA learning algorithm (McDonald et al., 2005a) for structured domains in which the optimization problem can be solved by using Hidreth's algorithm (Censor and Zenios, 1997), which is faster than the quadratic programming technique. In contrast to the previous method, this paper presents an online algorithm for dependency parsing in which we can attain the maximize margin of the training data without using optimization techniques. It is thus much faster and easier to implement. The details of PA algorithm for dependency parsing are presented below.

Assume that we are given a set of sentences x_i and their dependency trees y_i where $i = 1, ..., n$. Let the feature mapping between a sentence x and a tree y be: $\Phi(x, y) = \Phi_1(x, y), \Phi_2(x, y), ..., \Phi_d(x, y)$ where each feature mapping Φ_j maps (x, y) to a real value. We assume that each feature $\Phi(x, y)$ is associated with a weight value. The goal of PA learning for dependency parsing is to obtain a parameter w that minimizes the hinge-loss function and the margin of learning data.

1 Input: $S = \{(x_i; y_i), i = 1, 2, ..., n\}$ in which x_i is the sentence and y_i is a dependency tree
2 Aggressive parameter C
3 Output: the PA learning model
4 Initialize: $w_1 = (0, 0, ..., 0)$
5 **for** *t=1, 2...* **do**
6 Receive an sentence x_t
7 Predict $y_t^* = \arg\max_{y \in Y}(\mathbf{w_t} \cdot \Phi(x_t, y_t))$
8 Suffer loss: $l_t = \mathbf{w_t} \cdot \Phi(x_t, y_t^*) - \mathbf{w_t} \cdot \Phi(x_t, y_t) + \sqrt{\rho(y_t, y_t^*)}$
9 Set:

PA: $\tau_t = \frac{l_t}{||\Phi(x_t, y_t^*) - \Phi(x_t, y_t)||^2}$

PA-I: $\tau_t = \min\{C, \frac{l_t}{||\Phi(x_t, y_t^*) - \Phi(x_t, y_t)||^2}\}$

PA-II: $\tau_t = \frac{l_t}{||\Phi(x_t, y_t^*) - \Phi(x_t, y_t)||^2 + \frac{1}{2C}}$

Update:
$w_{t+1} = w_t + \tau_t(\Phi(x_t, y_t) - \Phi(x_t, y_t^*))$
10 **end**

Algorithm 1: The Passive-Aggressive algorithm for dependency parsing.

Algorithm 1 shows the PA learning algorithm for dependency parsing in which its three variants are different only in the update formulas. In Algorithm 1, we employ two kinds of argmax algorithms: The first is the decoding algorithm for projective language data and the second one is for non-projective language data. Algorithm 1 shows (line 8) $p(y, y_t)$ is a real-valued loss for the tree y_t relative to the correct tree y. We define the loss of a dependency tree as the number of words which have an incorrect parent. Thus, the largest loss a dependency tree can have is the length of the sentence. The similar loss function is designed for the dependency tree with labeled. Algorithm 1 returns an averaged weight vector: an auxiliary weight vector v is maintained that accumulates the values of w after each iteration, and the returned weight vector is the average of all the weight vectors throughout training. Averaging has been shown to help reduce overfitting (McDonald et al., 2005a; Collins, 2002). It is easy to see that the main difference between the PA algorithms and the Perceptron algorithm (PC) (Collins, 2002) as well as the MIRA algorithm (McDonald et al., 2005a) is in line 9. As we can see in the PC algorithm, we do not need the value τ_t and in the MIRA algorithm we need an optimization algorithm to compute τ_t. We also have three updated formulations for obtaining τ_t in Line 9. In the scope of this paper, we only focus on using the second update formulation (PA-I method) for training dependency parsing data.

2.3 Feature Set

We denote p-word: word of parent node in dependency tree. c-word: word of child node. p-pos: POS of parent node. c-pos: POS of child node. p-pos+1: POS to the right of parent in sentence. p-pos-1: POS to the left of parent. c-pos+1: POS to the right of child. c-pos-1: POS to the left of child. b-pos: POS of a word in between parent and child nodes. The features used in our system are described below.

p-word,p-pos
p-word
p-pos
c-word, c-pos
c-word
c-pos

Table 1: Feature Set 1: Basic Unit-gram features

p-word, p-pos, c-word, c-pos
p-pos, c-word, c-pos
p-word, c-word, c-pos
p-word, p-pos, c-pos
p-word, p-pos, c-word
p-word, c-word
p-pos, c-pos

Table 2: Feature Set 2: Basic bi-gram features

p-pos, b-pos, c-pos
p-pos, p-pos+1, c-pos-1, c-pos
p-pos-1, p-pos, c-pos-1, c-pos
p-pos, p-pos+1, c-pos, c-pos+1
p-pos-1, p-pos, c-pos, c-pos+1

Table 3: Feature Set 3: In Between POS Features and Surrounding Word POS Features

- Tables 1 and 2 show our basic features. These features are added for entire words as well as for the 5-gram prefix if the word is longer than 5 characters.

- In addition to these features shown in Table 1, the morphological information for each pair of words p-word and c-word are represented. In addition, we also add the conjunction morphological information of p-word and c-word. We do not use the LEMMA and CPOSTAG information in our set features. The morphological information are obtained from FEAT information.

- Table 3 shows our Feature set 3 which take the form of a POS trigram: the POS of the parent, of the child, and of a word in between, for all words linearly between the parent and the child. This feature was particularly helpful for nouns identifying their parent (McDonald et al., 2005a).

- Table 3 also depicts these features taken the form of a POS 4-gram: The POS of the parent, child, word before/after parent and word before/after child. The system also used back-off features with various trigrams where one of the local context POS tags was removed.

- All features are also conjoined with the direction of attachment, as well as the distance between the two words being attached.

3 Experimental Results and Discussion

We test our parsing models on the CONLL-2007 (Hajič et al., 2004; Aduriz et al., 2003; Martí et al., 2007; Chen et al., 2003; Böhmová et al., 2003; Marcus et al., 1993; Johansson and Nugues, 2007; Prokopidis et al., 2005; Csendes et al., 2005; Montemagni et al., 2003; Oflazer et al., 2003) data set on various languages including Arabic, Basque, Catalan, Chinese, English, Italian, Hungarian, and Turkish. Each word is attached by POS tags for each sentence in both the training and the testing data. Table 4 shows the number of training and testing sentences for these languages. The table shows that the sentence length in Arabic data is largest and its size of training data is smallest. These factors might be affected to the accuracy of our proposed algorithm as we will discuss later.

The training and testing were conducted on a pentium IV at 4.3 GHz. The detailed information about the data are shown in the CONLL-2007 shared task. We applied non-projective and projective parsing along with PA learning for the data in CONLL-2007.

Table 5 reports experimental results by using the first order decoding method in which an MST parsing algorithm (McDonald et al., 2005b) is applied for non-projective parsing and the Eisner's method is used for projective language data. In fact, in our method we applied non-projective parsing for the Italian data, the Turkish data, and the Greek data. This was because we did not have enough time to train all training data using both projective and non-projective parsing. This is the problem of discriminative learning methods when performing on a large set of training data. In addition, to save time in training we set the number of best trees k to 1 and the parameter C is set to 0.05.

Table 5 shows the comparison of the proposed method with the average, and three top systems on the CONLL-2007. As a result, our method yields results above the average score on the CONLL-2007 shared task (Nivre et al., 2007).

Table 5 also indicates that the Basque results obtained a lower score than other data. We obtained 69.11 UA score and 58.16 LA score, respectively. These are far from the results of the Top3 scores (81.13 and 75.49). We checked the outputs of the Basque data to understand the main reason for the errors. We see that the errors in our methods are usually mismatched with the gold data at the labels "ncmod" and "ncsubj". The main reason might be that the application of projective parsing for this data in both training and testing is not suitable. This was because the number of sentences with at least 1 non projective relation in the data is large (26.1).

The Arabic score is lower than the scores of other data because of some difficulties in our method as follows. Morphological and sentence length problems are the main factors which affect the accuracy of parsing Arabic data. In addition, the training size in the Arabic is also a problem for obtaining a good result. Furthermore, since our tasks was focused on improving the accuracy of English data, it might be unsuitable for other languages. This is an imbalance

Languages	Training size	Tokens size	tokens-per-sent	% of NPR	% of-sentence AL-1-NPR
Arabic	2,900	112,000	38.3	0.4	10.1
Basque	3,200	51,000	15.8	2.9	26.2
Catalan	15,000	431,000	28.8	0.1	2.9
Chinese	57,000	337,000	5.9	0.0	0.0
Czech	25,400	432,000	17.0	1.9	23.2
English	18,600	447,000	24.0	0.3	6.7
Greek	2,700	65,000	24.2	1.1	20.3
Hungarian	6,000	132,000	21.8	2.9	26.4
Italian	3,100	71,000	22.9	0.5	7.4
Turkish	5,600	65,000	11.6	0.5	33.3

Table 4: The data used in the multilingual track (Nivre et al., 2007). NPR means non-projective-relations. AL-1-NPR means at-least-least 1 non-projective relation.

problem in our method. Table 5 also shows the comparison of our system to the average score and the Top3 scores. It depicts that our system is accurate in English data, while it has low accuracy in Basque and Arabic data.

We also evaluate our models in the domain adaptation tasks. This task is to adapt our model trained on PennBank data to the test data in the Biomedical domain. The pchemtb-closed shared task (Marcus et al., 1993; Johansson and Nugues, 2007; Kulick et al., 2004) is used to illustrate our models. We do not use any additional unlabeled data in the Biomedical domain. Only the training data in the PennBank is used to train our model. Afterward, we selected carefully a suitable parameter using the development test set. We set the parameter C to 0.01 and select the non projective parsing for testing to obtain the highest result in the development data after performing several experiments. After that, the trained model was used to test the data in Biomedical domain. The score (UA=82.04; LA=79.50) shows that our method yields results above the average score (UA=76.42; LA=73.03). In addition, it is officially coming in 4th place out of 12 teams and within 1.5% of the top systems.

The good result of performing our model in another domain suggested that the PA learning seems sensitive to noise. We hope that this problem is solved in future work.

4 Conclusions

This paper presents an online algorithm for dependency parsing problem which have tested on various language data in CONLL-2007 shared task. The performance in English data is close to the Top3 score. We also perform our algorithm on the domain adaptation task, in which we only focus on the training of the source data and select a suitable parameter using the development set. The result is very good as it is close to the Top3 score of participating systems. Future work will also be focused on extending our method to a version of using semi-supervised learning that can efficiently be learnt by using labeled and unlabeled data. We hope that the application of the PA algorithm to other NLP problems such as semantic parsing will be explored in future work.

Acknowledgments

We would like to thank D. Yuret for his helps in checking errors of my parser's outputs. We would like to thank Vinh-Van Nguyen his helps during the revision process and Mary Ann Mooradian for correcting the paper.

We would like to thank to anonymous reviewers for helpful discussions and comments on the manuscript. Thank also to Sebastian Riedel for checking the issues raised in the reviews.

The work on this paper was supported by a Monbukagakusho 21st COE Program.

References

A. Abeillé, editor. 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer.

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Diaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank. In *Proc. of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 201–204.

Languages	Unlabled Accuracy					Labeled Accuracy					NTeams
	PA-I	Average	Top3	Top2	Top1	PA-I	Average	Top3	Top2	Top1	
Arabic	73.46	78.84	84.21	85.81	86.09	68.34	74.75	83.0	75.08	76.52	20
Basque	69.11	75.15	81.13	81.93	81.13	58.16	68.06	75.49	75.73	76.92	20
Catalan	88.12	87.98	93.12	93.34	93.40	83.23	79.85	87.90	88.16	88.70	20
Chinese	84.05	81.98	87.91	88.88	88.94	79.77	76.59	83.51	83.84	84.69	21
Czech	80.91	77.56	84.19	85.16	86.28	72.54	70.12	77.98	78.60	80.19	20
English	88.01	82.67	89.87	90.13	90.63	86.73	80.95	88.41	89.01	89.61	23
Greek	77.56	77.78	81.37	82.04	84.08	70.42	70.22	74.42	74.65	76.31	20
Hungarian	78.13	76.34	82.49	83.51	83.55	68.12	71.49	78.09	79.53	80.27	21
Italian	80.40	82.45	87.68	87.77	87.91	75.06	78.06	78.09	79.53	80.27	20
Turkish	80.19	73.19	85.77	85.77	86.22	67.63	73.19	79.24	79.79	79.81	20
Multilingual-average	79.99	71.13	85.62	85.71	86.55	72.52	65.77	79.90	80.28	80.32	23
pchemtb-closed	82.04	76.42	83.08	83.38	83.42	79.50	73.03	80.22	80.40	81.06	8

Table 5: Dependency accuracy in the CONLL-2007 shared task.

A. Böhmová, J. Hajič, E. Hajičová, and B. Hladká. 2003. The PDT: a 3-level annotation scenario. In Abeillé (Abeillé, 2003), chapter 7, pages 103–127.

Y. Censor and S.A. Zenios. 1997. Parallel optimization: theory, algorithms, and applications. In *Oxford University Press*.

K. Chen, C. Luo, M. Chang, F. Chen, C. Chen, C. Huang, and Z. Gao. 2003. Sinica treebank: Design criteria, representational issues and implementation. In Abeillé (Abeillé, 2003), chapter 13, pages 231–248.

Y.J. Chu and T.H. Liu. 1965. On the shortest arborescence of a directed graph. In *Science Sinica*.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

S. Corston-Oliver, A. Aue, K. Duh, , and E. Ringger. 2006. Multilingual dependency parsing using bayes point machines. In *Proceedings of HLT/NAACL*.

K. Crammer and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

K. Crammer, O. Dekel, J. Keshet, S.Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:581–585.

D. Csendes, J. Csirik, T. Gyimóthy, and A. Kocsor. 2005. *The Szeged Treebank*. Springer.

J. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of COLING 1996*, pages 340–345.

J. Hajič, O. Smrž, P. Zemánek, J. Šnaidauf, and E. Beška. 2004. Prague Arabic dependency treebank: Development in data and tools. In *Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*, pages 110–117.

R. Johansson and P. Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA)*.

S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein, and L. Ungar. 2004. Integrated annotation for biomedical information extraction. In *Proc. of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

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

M. A. Martí, M. Taulé, L. Màrquez, and M. Bertran. 2007. CESS-ECE: A multilingual and multilevel annotated corpus. Available for download from: http://www.lsi.upc.edu/∼mbertran/cess-ece/.

R. McDonald, K. Cramer, and F. Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of ACL*.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of the Human Language Technology Conf. and the Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530.

R. McDonald, K. Crammer, and F. Pereira. 2006. Multilingual dependency parsing with a two-stage discriminative parser. In *Conference on Natural Language Learning*.

S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Lenci, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Nana, F. Pianesi, and

R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Abeillé (Abeillé, 2003), chapter 11, pages 189–210.

J. Nivre, J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proc. of the Tenth Conf. on Computational Natural Language Learning (CoNLL)*, pages 221–225.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

K. Oflazer, B. Say, D. Zeynep Hakkani-Tür, and G. Tür. 2003. Building a Turkish treebank. In Abeillé (Abeillé, 2003), chapter 15, pages 261–277.

P. Prokopidis, E. Desypri, M. Koutsombogera, H. Papageorgiou, and S. Piperidis. 2005. Theoretical and practical issues in the construction of a Greek dependency treebank. In *Proc. of the 4th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 149–160.

B. Taskar, D. Klein, M. Collins, D. Koller, and C.D. Manning. 2004. Max-margin parsing. In *proceedings of EMNLP*.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2005. Support vector machine learning for interdependent and structured output spaces. In *proceedings ICML 2004*.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proc. 8th International Workshop on Parsing Technologies (IWPT)*, pages 195–206.