Improving Chinese POS Tagging with Dependency Parsing

Zhenghua Li, Wanxiang Che, Ting Liu*
Research Center for Social Computing and Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology, China
{lzh,car,tliu}@ir.hit.edu.cn

Abstract

Recent research usually models POS tagging as a sequential labeling problem, in which only local context features can be used. Due to the lack of morphological inflections, many tagging ambiguities in Chinese are difficult to handle unless consulting larger contexts. In this paper, we try to improve Chinese POS tagging by using long-distance dependencies produced by a statistical dependency parser. Experimental results show that, despite error propagation, the syntactic features can significantly improve the tagging accuracy from 93.88% to 94.41% ($p < 10^{-5}$). Detailed analysis shows that these features are helpful for ambiguous pairs like {NN,VV} and {DEC,DEG}.[1]

1 Introduction

Part-of-speech (POS) tagging is a necessary and important step for many natural language tasks, for example, named entity recognition, parsing and sentence boundary detection. In the current literature, POS tagging is treated as a typical sequence labeling problem, to which many models have been successfully applied, such as maximum-entropy (Ratnaparkhi, 1996), conditional random fields (CRF) (Lafferty et al., 2001) and perceptron (Collins, 2002). To facilitate fast decoding, these models make the Markovian independence assumption that the current tag depends on one or two previous tags. In addition, they can merely consider local context features, e.g. two words in both sides of the focus word. This works quite well for English, because inflections are useful and strong clues for POS tags. However, due to the lack of morphological inflections, Chinese POS tagging has proven to be much more challenging than English. With a typical sequential labeling model such as Conditional Random Fields (CRF), the tagging accuracy is about 97% for English, while less than 94% for Chinese (Huang et al., 2009).

NN-VV ambiguities are one of the most notorious difficulties for Chinese POS tagging. Figure 1 gives two examples. We can see that the POS tagger can effortlessly assign the right tags to both "development" and "develop" in the English side. However, it is very difficult in the Chinese side since no word form inflection is available and the context features may be too sparse or uninformative. However, the introduction of long-distance dependencies can largely reduce this difficulty. In the upper example of Figure 1, the coordinate relation between "贸易/NN" and "发展" is a strong clue to "发展/NN". In the lower example of Figure 1, it is also easy to tag "发展" as "VV" if its coordination with "维护/VV" is known.

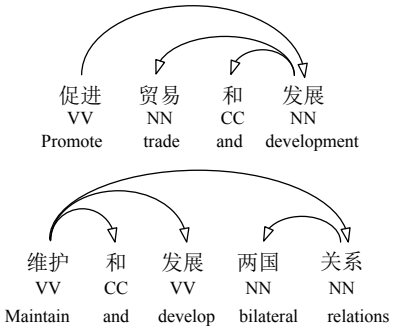

Figure 1: Examples of NN-VV ambiguities with dependency structures. The focus word is "发展".

As far as we know, there has been few research that try to improve POS tagging with dependency

*Correspondence author: tliu@ir.hit.edu.cn

[1]DEG and DEC are the two POS tags for the frequently used auxiliary word "的" (dē, of) in Chinese. The associative "的" is tagged as DEG, such as "父亲/father 的 眼睛/eyes (eyes of the father)"; while the one in a relative clause is tagged as DEC, such as "他/he 取得/made 的 进步/progress (progress that he made)".

parsing . The reason for this may be three-fold. Firstly, POS tags are indispensable features for dependency parsing since pure lexical features lead to severe sparseness problem. Therefore, POS tagging is traditionally considered as a supporting task for dependency parsing. Secondly, Chinese dependency parsing performs not well. The accuracy is about 85% when gold-standard POS tags are given, and quickly drops to about 79% when using automatically assigned POS tags. Therefore, error propagation may be an obstacle to research on this idea. Thirdly, inefficiency of syntactic parsing may be another concern. However, we believe that this problem can be relieved in the case of dependency parsing, since efficient cubic-time or even linear-time parsing models have been proposed for dependency parsing (McDonald, 2006; Nivre and Hall, 2005).

In this paper, we propose several kinds of syntactic features based on the output of a statistical dependency parser. And we use these features to enhance a traditional POS tagging model so that long-distance information can be explored. Experimental results show that this effort is rewarding, and the tagging accuracy is significantly improved. Detailed error analysis confirms the usefulness of these syntactic features.

2 Baseline POS Taggers

Given an input sentence $\mathbf{x} = w_1...w_n$, we denote its *POS tag sequence* by $\mathbf{t} = t_1...t_n$, where $t_i \in \mathcal{T}$, $1 \leq i \leq n$, and $\mathcal{T}$ is the POS tag set. The goal of POS tagging is to find the highest-scoring sequence:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \mu(\mathbf{x}, \mathbf{t})$$

We implement two baseline taggers, i.e., a Perceptron-based tagger and a CRF-based tagger. As a linear model, Perceptron defines the score of a tag sequence to be

$$\mu(\mathbf{x}, \mathbf{t}) = \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{t})$$

where $\mathbf{f}(\mathbf{x}, \mathbf{t})$ refers to the feature vector and $\mathbf{w}$ is the corresponding weight vector. We use standard *averaged perceptron* to learn the weight vector (Collins, 2002).

As a probabilistic model, CRF defines the probability of a sequence to be

$$\mu(\mathbf{x}, \mathbf{t}) = P(\mathbf{t}|\mathbf{x}) = \frac{e^{\mathbf{w}\cdot\mathbf{f}(\mathbf{x},\mathbf{t})}}{\sum_{\mathbf{t}'} e^{\mathbf{w}\cdot\mathbf{f}(\mathbf{x},\mathbf{t}')}}$$

We adopt the *exponentiated gradient* algorithm to learn the weight vector (Collins et al., 2008).

For POS tagging features $\mathbf{f}(\mathbf{x}, \mathbf{t})$, we follow the work of Zhang and Clark (2008a). Besides standard *POS unigram* ($\underline{w_i\ t_i}$), *bigram* ($\underline{t_{i-1}\ t_i}$) and *trigram* ($\underline{t_{i-2}\ t_{i-1}\ t_i}$) features, they explore many features composed of Chinese characters, such as $\underline{c_{i,0}\ t_i}$ and $\underline{c_{i,-1}\ t_i}$, where $c_{i,0}$ and $c_{i,-1}$ denote the start and end characters of w_i. These character-based features are very helpful for tagging accuracy. Due to space limitation, we refer to Zhang and Clark (2008a) for the complete feature description. In order to distinguish these features from our proposed syntactic features, we refer to them as the *basic features* and denote them as $\mathbf{f}_b(\mathbf{x}, \mathbf{t})$. Given $\mathbf{w}$, we adopt the Viterbi algorithm to get the optimal tagging sequence.

3 POS Tagging with Syntactic Features

The framework of our method is shown in Figure 2. Given an input sentence $\mathbf{x}$, we first use the CRF-based model to produce a tagging sequence $\mathbf{t}^C$. Then, based on $\mathbf{t}^C$, we use a statistical dependency parser to obtain the syntactic tree $\mathbf{d}^A$. Finally, both $\mathbf{t}^C$ and $\mathbf{d}^A$ are used as additional features in the enhanced Perceptron-based model. We use Perceptron to build our model because it is competitive to CRF in tagging accuracy but requires much less training time. During training phase, we adopt the 10-fold cross validation strategy to produce both $\mathbf{t}^C$ and $\mathbf{d}^A$ for the training set.

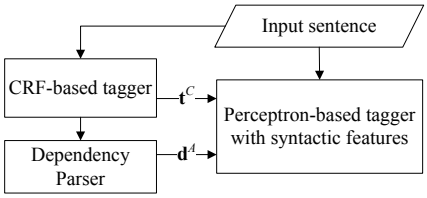

Figure 2: Framework of our method.

Based on a guide POS sequence $\mathbf{t}'$ ($\mathbf{t}^C$ in this paper) and a syntactic tree $\mathbf{d}$, we propose three kinds of features, as shown in Table 1. Our use of guide POS Features $\mathbf{f}_g(\mathbf{x}, \mathbf{t}', \mathbf{t})$ is mainly inspired by *stacked learning*, in which results of the first-level predicator are used to guide the second (Cohen and de Carvalho, 2005; Nivre and McDonald, 2008; Martins et al., 2008).

Syntactic features $\mathbf{f}_s(\mathbf{x}, \mathbf{d}, \mathbf{t})$ explore features related with the head and children of the focus word. Syntactic features with guide POS tags

Guide POS Features: $\mathbf{f}_g(\mathbf{x}, \mathbf{t}', \mathbf{t})$		Syntactic Features: $\mathbf{f}_s(\mathbf{x}, \mathbf{d}, \mathbf{t})$		Syntactic Features with Guide POS: $\mathbf{f}_{sg}(\mathbf{x}, \mathbf{t}', \mathbf{d}, \mathbf{t})$			
$t'_i\ t_i$	$t'_{i-1}\ t'_i\ t_i$	$\#lc(i)\ t_i$	$w_{h(i)}\ t_i$	$\#lc(i)\ t'_i\ t_i$	$\#rc(i)\ t'_i\ t_i$	$t'_{h(i)}\ t_i$	$t'_i\ d(i)\ t_i$
$t'_{i-1}\ t_i$	$t'_i\ t'_{i+1}\ t_i$	$w_{lc(i,k)}\ t_i$	$d(i)\ t_i$	$t'_{lc(i,k)}\ t'_i\ t_i$	$t'_{rc(i,k)}\ t'_i\ t_i$	$t'_{h(i)}\ t'_i\ t_i$	$t'_{h(i)}\ d(i)\ t_i$
$t'_{i+1}\ t_i$	$t'_{i-1}\ t'_{i+1}\ t_i$	$\#rc(i)\ t_i$	$w_{h(i)}\ d(i)\ t_i$	$t'_{lc(i,k)}\ t_i$	$t'_{rc(i,k)}\ t_i$	$t'_{h(i)}\ w_i\ t_i$	$t'_{h(i)}\ t'_i\ d(i)\ t_i$
$t'_i \sim \mathbf{f}_b(\mathbf{x})$	$t'_{i-1}\ t'_i\ t'_{i+1}\ t_i$	$w_{rc(i,k)}\ t_i$	$w_i\ d(i)\ t_i$				$t'_{h(i)}\ w_i\ d(i)\ t_i$

Table 1: Feature templates for our enhanced Perceptron-based tagger. $\mathbf{t}'$ denotes a guide POS sequence, which is $\mathbf{t}^C$ in this paper. $t'_i \sim \mathbf{f}_b(\mathbf{x})$ means that we concatenate t'_i and each feature in $\mathbf{f}_b(\mathbf{x}, \mathbf{t})$ to obtain a new one. $h(i)$ denotes the index of the head of i in the syntactic tree $\mathbf{d}$; while $d(i)$ means the distance and direction of the dependency $h(i) \rightarrow i$. $\#lc(i)$ means the number of left-side children of i, and $lc(i, k)$ is the index of the k^{th} left child of i. Analogously, $\#rc(i)$ and $rc(i, k)$ considers right-side children of i.

$\mathbf{f}_{sg}(\mathbf{x}, \mathbf{t}', \mathbf{d}, \mathbf{t})$ further make use of the POS tags of the head and children of the focus word. The effectiveness of these features will be examined in the experiments.

4 Experiments and Analysis

The Penn Chinese Treebank 5.1 (CTB5) is used as the labeled data (Xue et al., 2005). We follow the setup of Duan et al. (2007) and split CTB5 into training (secs 001-815 and 1001-1136), development (secs 886-931 and 1148-1151), and test (secs 816-885 and 1137-1147) sets. Head-finding rules are used to turn the bracketed sentences into dependency structures (Zhang and Clark, 2008b).

We adopt the second-order graph-based model of McDonald and Pereira (2006) for our statistical dependency parser. Its time complexity for decoding is $O(n^3)$. On the test set, its parsing accuracy is 85.01% when using gold-standard POS tags, and is 78.82% when using automatic POS tags produced by the baseline CRF tagger.

4.1 Main Results

Table 2 gives the final results. The first row contains two baseline tagging models which only use the basic features $\mathbf{f}_b(\mathbf{x}, \mathbf{t})$. We can see that the Perceptron-based and CRF-based models achieve comparable accuracies.

From the results in the second row, we can find that using guide POS features only modestly (but significantly) improve the accuracy. This model can be regarded as the integrated model of both Perceptron-based and CRF-based models.

In the third row, we explore syntactic features based on gold-standard trees and aim to find out the usefulness of syntactic features without error propagation. Obviously, correct syntactic features can greatly help resolve tagging ambiguities. Using all the features leads to the best accuracy.

Method	Token	Known	Unknown
Perceptron with $\mathbf{f}_b(\mathbf{x}, \mathbf{t})$	93.82	94.65	81.32
CRF with $\mathbf{f}_b(\mathbf{x}, \mathbf{t})$	93.88	94.70	81.51
+$\mathbf{f}_g(\mathbf{t}^C)$	94.02	94.84	81.67
+$\mathbf{f}_s(\mathbf{d}^G)$	96.02	96.85	83.51
+$\mathbf{f}_s(\mathbf{d}^G)$+$\mathbf{f}_{sg}(\mathbf{t}^C, \mathbf{d}^G)$	96.19	96.99	84.27
+$\mathbf{f}_s(\mathbf{d}^G)$+$\mathbf{f}_{sg}(\mathbf{t}^C, \mathbf{d}^G)$+$\mathbf{f}_g(\mathbf{t}^C)$	**96.26**	**97.05**	**84.37**
+$\mathbf{f}_s(\mathbf{d}^A)$	94.06	94.91	81.44
+$\mathbf{f}_s(\mathbf{d}^A)$+$\mathbf{f}_{sg}(\mathbf{t}^C, \mathbf{d}^A)$	**94.41**	**95.26**	81.67
+$\mathbf{f}_s(\mathbf{d}^A)$+$\mathbf{f}_{sg}(\mathbf{t}^C, \mathbf{d}^A)$+$\mathbf{f}_g(\mathbf{t}^C)$	94.37	95.20	**81.95**

Table 2: Tagging accuracy on the test set (%). $\mathbf{t}^C$ denotes the tagging sequence of the baseline CRF model. $\mathbf{d}^G$ refers to the gold-standard tree; while $\mathbf{d}^A$ denotes the automatically parsed tree. Note that we omit $\mathbf{x}$ and $\mathbf{t}$ in $\mathbf{f}_{s/g/sg}(.)$ for brevity.

In the fourth row, we examine our method in the realistic scenario. The syntactic tree is automatically produced by the parser trained on the training set. The accuracy improvement is modest but significant when only adding pure syntactic features $\mathbf{f}_s(\mathbf{x}, \mathbf{d}^A, \mathbf{t})$ ($p < 0.01$).[2] Using syntactic features with guide POS tags, i.e., $\mathbf{f}_{sg}(\mathbf{x}, \mathbf{t}^C, \mathbf{d}^A, \mathbf{t})$, can boost the accuracy by a large margin. Compared with the baseline models, the improvement is significant ($p < 10^{-5}$). Then, adding guide POS features $\mathbf{f}_g(\mathbf{x}, \mathbf{t}^C, \mathbf{t})$ slightly decreases the accuracy, but somehow improves the accuracy of unknown words.

4.2 Error Analysis

To find out how the syntactic features help tagging, we conduct detailed error analysis through comparing the results of different models, as shown in Table 3. We choose the most frequent error pat-

[2] We adapt Dan Bikel's randomized parsing evaluation comparator to do significant test for POS tagging. http://www.cis.upenn.edu/~bikel/software.html

terns made by the baseline CRF-based model, and presents them in descending order of frequency.

error pattern	CRF	Gold Parse	Auto Parse
VV → NN	456	-197	-15
NN → VV	341	-180	-30
DEC → DEG	227	-222	-66
NR → NN	224	+1	-5
DEG → DEC	191	-187	-57
JJ → NN	135	+10	0
NN → NR	84	-3	0
NN → JJ	63	0	+1

Table 3: The number of error patterns made by different models. An error pattern "X → Y" means that the focus word, whose true tag is 'X', is assigned a tag 'Y'. "CRF" refers to the baseline CRF-based model. "Gold Parse" and "Auto Parse" are two perceptron-based models augmented with syntactic features, and correspond to the best models in the third and fourth rows of Table 2, respectively. The signed numbers in the last two columns present the change of error number.

From the column of "Gold Parse" we can see that using correct syntactic features can greatly reduce the errors for ambiguous pairs {NN, VV} and {DEC, DEG}. Especially, nearly all ambiguities of {DEC, DEG} are correctly resolved.However, syntactic features are not helpful for ambiguities like {NN,NR} and {NN,JJ}. One common characteristic of these two pairs are that the two POS tags play similar roles from syntactic view. In other words, their syntactic contexts are usually similar, which naturally explains why the gold-standard syntactic features fail to help. In contrast, "NN" and "VV" (or "DEC" and "DEG") usually have completely different syntactic structures. This demonstrates that our proposed syntactic features are very effective.

Using automatic syntactic features still help resolve {NN, VV} and {DEC, DEG}. However, the error reduction is much less than that of using correct parse trees, which is obviously due to error propagation. Likewise, the errors over {NN, NR} and {NN,JJ} are not influenced.

5 Related Work

Recently, extensive research on Chinese POS tagging has been done. Tseng et al. (2005) enhance the tagging accuracy of unknown words by using rich morphological features. Huang et al. (2009) improve a bigram HMM POS tagger by latent annotation and self-training. Several methods are proposed to handle joint word segmentation and POS tagging of Chinese (Jiang et al., 2008; Zhang and Clark, 2008a; Kruengkrai et al., 2009).

The most closely related work to our approach is the one of Huang et al. (2007), which also explores syntactic features to boost the tagging accuracy. In stead of directly using syntactic features in a discriminative POS tagger, they adopt the RankBoost-based algorithm to rerank the N-best output of a sophisticated HMM tagger (Collins and Koo, 2005). As a discriminative model, the reranker can make use of rich features including morphological features, word/tag n-grams and syntactic features. Another difference from our work is that their syntactic tree is produced by the constituent parser of Charniak (2000) which jointly solves POS tagging and parsing. In this way, they might obtain higher-quality syntactic features since error propagation can be alleviated to some extent. Their reranking approach lead to an improvement of about 1% in tagging accuracy over the HMM tagger. In this paper, we propose another way to incorporate long-distance information for POS tagging. In another perspective, our approach may be more promising in real applications, since dependency parsing is simpler and potentially more efficient than constituent parsing.

6 Conclusions

In this paper, we show that the accuracy of a discriminative sequential POS tagger can be substantially improved by exploring syntactic features. We also show that the syntactic features can help resolve ambiguities like {NN,VV} and {DEC,DEG}, which are difficult to handle when only local contexts are considered. In the future, we will investigate joint POS tagging and dependency parsing models to further improve tagging accuracy.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 61133012, the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2009069) and the Fundamental Research Funds for the Central Universities (HIT.KLOF.2010064).

References

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *ANLP'00*, pages 132–139.

William W. Cohen and Vitor Rocha de Carvalho. 2005. Stacked sequential learning. In *IJCAI'05*, pages 671–676.

Michael J. Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, pages 25–70.

Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter Bartlett. 2008. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *JMLR*, 9:1775–1822.

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

Xiangyu Duan, Jun Zhao, , and Bo Xu. 2007. Probabilistic models for action-based Chinese dependency parsing. In *Proceedings of ECML/ECPPKDD*.

Zhongqiang Huang, Mary Harper, and Wen Wang. 2007. Mandarin part-of-speech tagging and discriminative reranking. In *EMNLP-CoNLL07*, pages 1093–1102.

Zhongqiang Huang, Vladimir Eidelman, and Mary Harper. 2009. Improving a simple bigram hmm part-of-speech tagger by latent annotation and self-training. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 213–216, Boulder, Colorado, June. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL-08: HLT*, pages 897–904.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 513–521.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, pages 282–289.

André F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *EMNLP'08*, pages 157–166.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL06*.

Ryan McDonald. 2006. *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.

Joakim Nivre and Johan Hall. 2005. Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 13–95.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL 2008*, pages 950–958.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP 1996*.

Huihsin Tseng, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. In *SIGHAN Workshop on Chinese Language Processing*.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. In *Natural Language Engineering*, volume 11, pages 207–238.

Yue Zhang and Stephen Clark. 2008a. Joint word segmentation and POS tagging using a single perceptron. In *ACL08*, pages 888–896.

Yue Zhang and Stephen Clark. 2008b. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *EMNLP08*, pages 562–571.