Towards Accurate and Efficient Chinese Part-of-Speech Tagging

Weiwei Sun*
Peking University

Xiaojun Wan*
Peking University

From the perspective of structural linguistics, we explore paradigmatic and syntagmatic lexical relations for Chinese POS tagging, an important and challenging task for Chinese language processing. Paradigmatic lexical relations are explicitly captured by word clustering on large-scale unlabeled data and are used to design new features to enhance a discriminative tagger. Syntagmatic lexical relations are implicitly captured by syntactic parsing in the constituency formalism, and are utilized via system combination. Experiments on the Penn Chinese Treebank demonstrate the importance of both paradigmatic and syntagmatic relations. Our linguistically motivated, hybrid approaches yield a relative error reduction of 18% in total over state-of-the-art baselines. Despite the effectiveness to boost accuracy, computationally expensive parsers make hybrid systems inappropriate for many realistic NLP applications. In this article, we are also concerned with improving tagging efficiency at test time. In particular, we explore unlabeled data to transfer the predictive power of hybrid models to simple sequence models. Specifically, hybrid systems are utilized to create large-scale pseudo training data for cheap models. Experimental results illustrate that the re-compiled models not only achieve high accuracy with respect to per token classification, but also serve as a front-end to a parser well.

1. Introduction

In grammar, a part-of-speech (POS) is a linguistic category of words, generally defined by the syntactic or morphological behavior of the word in question. Automatically assigning POS tags to words plays an important role in parsing, word sense disambiguation, as well as many other NLP applications. Many successful tagging algorithms developed for English have been applied to many other languages as well. In some cases, the methods work well without large modifications, such as for German. But a number of augmentations and changes become necessary when dealing with highly inflected or agglutinative languages, as well as analytic languages, of which Chinese is the focus of this article. The Chinese language is characterized by the lack of formal

* The authors are with the Institute of Computer Science and Technology, the MOE Key Laboratory of Computational Linguistics, Peking University, Beijing 100871, China.
E-mail: {ws,wanxiaojun}@pku.edu.cn. Xiaojun Wan is the corresponding author.

Submission received: 24 August 2014; revised submission received: 10 December 2015; accepted for publication: 15 February 2016.

doi:10.1162/COLI_a_00253

devices such as morphological tense and number that often provide important clues for syntactic processing tasks. Although state-of-the-art tagging systems have achieved accuracies above 97% on English, Chinese POS tagging has proven to be more challenging and result in accuracies of about 93–94% (Ng and Low 2004; Tseng, Jurafsky, and Manning 2005; Huang, Harper, and Wang 2007; Huang, Eidelman, and Harper 2009; Li et al. 2011).

It is generally accepted that Chinese POS tagging often requires more sophisticated language processing techniques that are capable of drawing inferences from more subtle linguistic knowledge. From a linguistic point of view, meaning arises from the differences between linguistic units, including words, phrases, and so on, and these differences are of two kinds: **paradigmatic** (concerning substitution) and **syntagmatic** (concerning positioning). The distinction is a key one in structuralist semiotic analysis. Whereas syntagmatic relations are possibilities of combination, paradigmatic relations are functional contrasts—they involve differentiation. Both paradigmatic and syntagmatic lexical relations have a great impact on POS tagging, because the *value* of a word is determined by the two relations. For example, the Penn Chinese Treebank (CTB) (Xue et al. 2005)-style POS tags capture both paradigmatic and syntagmatic relations among words, given that its annotation criterion is the syntactic distribution of words.

With a linguistic motivation, we examine the impact of paradigmatic and syntagmatic lexical relations on Chinese POS tagging. Our study is motivated by the key language-specific property that Chinese is an analytic language and encodes lexical categorial information in a highly configurational rather than morphological way. This implies that capturing paradigmatic and syntagmatic relations must leverage on clues from a wider range of sources rather than surface strings. On the contrary, expressive morphological information can be found based on the word strings themselves. We argue that different strategies should be employed for designing tagging models for Chinese and other morphologically rich languages.

We present an error analysis of two state-of-the-art sequential taggers. The first one uses a generative hidden Markov model (HMM) that is enhanced by using latent annotations. This model is also known as symbol-refined HMM (SR-HMM). The second one is a discriminative tagger that uses linear-chain global linear models (LGLM) with rich contextual word features. Both achieve state-of-the-art performance. Our error analysis of both taggers shows that the lack of both paradigmatic and syntagmatic lexical knowledge accounts for a large part of tagging errors.

Our research is concerned with capturing paradigmatic and syntagmatic lexical relations to advance the state-of-the-art of Chinese POS tagging. Chinese, as an analytic language, encodes lexical categorial information in a highly configurational rather than morphological way. This language-specific property implies that capturing paradigmatic and syntagmatic relations must leverage clues from a wider range of sources rather than surface strings. To improve tagging performance, first, we use unsupervised word clustering to explore paradigmatic relations that are encoded in large-scale unlabeled data. Using unsupervised algorithms to acquire rich word representations, such as word clustering and word similarity calculation, is a very practical way to achieve wide-coverage lexical resources. To enhance the discriminative tagger, word clusters are explicitly utilized as new features. We are relying on the ability of discriminative learning to explore informative features that play a central role in boosting tagging performance.

Second, we study the possible impact of syntagmatic relations on POS tagging by comparatively analyzing (syntax-free) sequential tagging models and (syntax-based) parsing models in the constituency formalism. Inspired by the analysis, we use a full parser to implicitly capture syntagmatic relations and propose a simple yet effective stacking model to combine the complementary strengths of sequential taggers and parsers.

We conduct experiments on the CTB and Chinese Gigaword. We implement a discriminative sequential classification model for POS tagging that achieves state-of-the-art accuracy. Experiments show that this model is significantly improved by word cluster features in accuracy across a wide range of conditions. This confirms the importance of the paradigmatic relations. We then present a comparative study of our tagger and a constituency parser and a dependency parser, and show that the combination of heterogeneous models can significantly improve tagging accuracy. Our experiments show that stacking is a very effective method to combine the complementary strengths of heterogeneous models. This demonstrates the importance of the syntagmatic relations. Cluster-based features and the stacking model result in a relative error reduction of 18% in terms of the word classification accuracy.

Although predictive powers of hybrid systems are significantly better than individual systems, they are not suitable for large-scale real word applications that have stringent time requirements. The best performing model is slow and large, and fast and compact models are less accurate, because either they are not expressive enough or they overfit to the limited training data. To improve POS tagging efficiency without loss of accuracy, we explore unlabeled data to transfer the predictive power of complex, inefficient models to simple, efficient models. Specifically, hybrid systems are utilized to create large-scale pseudo training data for cheap sequence models. For the SR-HMM tagger, pseudo training data are able to estimate finer-grained latent variables, and for the discriminative tagger, tagging accuracy can be improved by extending the context for feature extraction.

Experiments on the CTB and Gigaword demonstrate that unlabeled data are effective to transfer the predictive power of hybrid models to simple models, including both latent variable generative models and global linear classifiers. On one hand, the precision in terms of word classification is improved to 95.34%, which is equivalent to the parser-integrated hybrid model. On the other hand, re-compiled models are adapted based on parsing results, and as a result the ability to capture syntagmatic lexical relations is improved, too. Different from the purely supervised sequence models, re-compiled models also serve as a front-end to a parser well.

Our study has been partially published in Sun and Uszkoreit (2012) and Sun, Peng, and Wan (2013). For this iteration, we re-implement all models, and therefore experimental results are not exactly the same. We also release our implementation for research purposes. The related resources can be downloaded at `www.icst.pku.edu.cn/lcwm/lexer`.

2. Motivating Analysis

Many algorithms have been applied to computationally assigning POS labels to English words, including hand-written rules, generative HMM tagging, and discriminative sequence labeling. Such methods have been applied to many other languages as well. In some cases, the methods work well without large modifications, such as for German POS tagging. But a number of augmentations and changes became necessary when dealing with Chinese, a language that has little, if any, inflectional morphology. Whereas state-of-the-art tagging systems have achieved accuracies above 97% on English, Chinese POS tagging has proven to be more challenging and obtains accuracies of about 93–94% (Tseng, Jurafsky, and Manning 2005; Huang, Harper, and Wang 2007;

Huang, Eidelman, and Harper 2009; Li et al. 2011). In this section, we give a brief introduction and a comparative analysis to several models that have been recently designed to resolve the Chinese POS tagging problem.

2.1 State-of-the-Art Tagging Models

2.1.1 Linear-Chain Global Linear Model (LGLM). All state-of-the-art English POS taggers are based on discriminative sequence labeling models—for example, maximum entropy (Toutanova et al. 2003), support vector machines (Giménez and Màrquez 2004), structure perceptron (Collins 2002; Shen, Satta, and Joshi 2007; Huang, Fayong, and Guo 2012), and conditional random fields (Sun 2014). A discriminative learner can be easily extended with arbitrary features and is therefore suitable to recognize more new words. Moreover, a majority of the POS tags are locally dependent on each other, so the Markov assumption can well capture the syntactic relations among words. A majority of discriminative POS taggers utilize Global Linear Models (GLMs) for learning and prediction. A GLM represents the sequence labeling task through a feature-vector representation of the whole observation and tag sequence pair. We use Φ to denote a map from $(x_{[1:n]}, y_{[1:n]})$ pairs to d-dimensional feature vectors, where $x_{[1:n]}$ and $y_{[1:n]}$ are the observation and the tag sequences. Φ is often referred to as a **global representation function**. Using this feature-vector representation, the conditional probability of the label sequence given the observation sequence is modeled as:

$$P(y_{[1:n]}|x_{[1:n]}) \propto e^{w\Phi(x_{[1:n]},y_{[1:n]})} \tag{1}$$

By adopting the global feature-vector representation, we can flexibly incorporate rich context features. The global feature-vector can be further decomposed into the sum of local feature-vectors with smaller granularity:

$$\Phi(x_{[1:n]}, y_{[1:n]}) = \sum_{i=1}^{n} \phi(h_i, y_i) \tag{2}$$

where h_i is the history correlated with y_i. Using this local representation, we can use the Viterbi algorithm for inference, which finds the optimal tag sequence $\hat{y}_{[1:n]}$ that maximizes the following score:

$$\hat{y}_{[1:n]} = \arg\max_{y_{[1:n]}} \sum_{i=1}^{n} w\phi(h_i, y_i) \tag{3}$$

Discriminative learning is also an appropriate solution for Chinese POS tagging, because of its flexibility to include knowledge from multiple linguistic sources. Tseng, Jurafsky, and Manning (2005) introduced a maximum entropy–based model, which includes morphological features for unknown word recognition, and Sun (2011) studied the joint word segmentation and POS tagging problem and developed a fully discriminative method. However, they did not deeply analyze the problem from a linguistic view.

The global linear algorithm we adopt in this article is averaged perceptron (Collins 2002).

2.1.2 Symbol-Refined Hidden Markov Model (SR-HMM). Generative models with latent annotations (LAs) obtain state-of-the-art performance for a number of NLP tasks. For example, both context-free Grammar (CFG) and tree-substitution grammar (TSG) with refined latent variables achieve excellent results for syntactic parsing (Matsuzaki, Miyao, and Tsujii 2005; Shindo et al. 2012). For Chinese POS tagging, Huang, Eidelman, and Harper (2009) described and evaluated a bigram HMM tagger that utilizes latent annotations. The use of latent annotations substantially improves the performance of a simple generative bigram tagger, outperforming a trigram HMM tagger with sophisticated smoothing.

An HMM POS tagger models the joint distribution of the observation sequence $x_{[1:n]}$ and the tag sequence $y_{[1:n]}$. Under the first-order Markov assumption, the inference problem can be computed as:

$$\begin{aligned}\hat{y}_{[1:n]} &= \arg\max_{y_{[1:n]}} P(x_{[1:n]}, y_{[1:n]}) \\ &= \arg\max_{y_{[1:n]}} \prod_{i=1}^{n} P(y_i|y_{i-1})P(x_i|y_i)\end{aligned} \tag{4}$$

where the set $\{P(y_i|y_{i-1})\}$ are transition parameters, which model the transition from tag y_{i-1} to tag y_i, and the set $\{P(x_i|y_i)\}$ are emission parameters, which model the generation of word x_i from tag y_i. However, the first-order Markov independence assumption of a bigram tagger is too strong in many cases. Huang, Eidelman, and Harper (2009) introduces using latent annotation to refine the tags of a bigram HMM model. For example, the NR tag may be split into NR-1 and NR-2, and the corresponding symbol-refined tag sequence for "Mr./NR Smith/NR saw/VV Ms./NR Smith/NR" can be denoted as "Mr./NR-2 Smith/NR-1 saw/VV-2 Ms./NR-2 Smith/NR-1."

The objective of training a symbol-refined bigram tagger is to solve the LA-involved emission and transition parameters by maximizing the likelihood of the training data. In contrast with a non-symbol-refined HMM tagger, where the POS tags are observed, the latent annotations are unseen variables. In order to learn these parameters, a variant of EM algorithm is used. The objective function used for decoding is:

$$\hat{y}_{[1:n]} = \arg\max_{y_{[1:n]}} \prod_{i=1}^{n-1} P(y_i, y_{i+1}|x_{[1:n]}) \tag{5}$$

This goal function is a variant of the *MAX-RULE-PRODUCT* algorithm in Petrov and Klein (2007), which maximizes the product of rule posteriors. This algorithm is not probabilistically correct but follows the instinct of choosing the tree with the greatest chance of having all rules correct. Similarly, the goal function used in the SR-HMM POS tagger tries to find the tag sequence with the greatest chance of having all bigrams correct. The bigram tag posterior is calculated by marginalizing out the latent annotations in the bigram latent tag posterior.

$$P(y_i = t, y_{i+1} = s|x_{[1:n]}) = \sum_{t_a \in S(t)} \sum_{s_b \in S(s)} P(t_a, s_b|x_{[1:n]}) \tag{6}$$

Table 1
Training, development, and test data on CTB 6.0.

	#Sentence	#Word
Training	22,277	609,060
Development	1,763	49,620
Test	2,556	73,153

Huang, Harper, and Wang (2007) and Huang, Eidelman, and Harper (2009) present empirical studies of generative Chinese POS tagging. In particular, evaluation of the SR-HMM model obtains state-of-the-art performance. In this article, we adopt their tagger for experiments.

2.1.3 Local Classification. A very simple approach to POS tagging is to formulate it as a local word classification problem. Various features can be drawn upon information sources such as word forms and characters that constitute words. Previous study on many languages shows that local classification is inadequate to capture structural information of output labels, and thus does not perform as well as structured models. The local classification algorithm we adopt in this article is linear SVM.[1] Because it is a local linear model, we denote it as LLM.

2.2 Evaluation

2.2.1 Setting. Penn Chinese Treebank (CTB) (Xue et al. 2005) is a popular data set to evaluate a number of Chinese NLP tasks, including word segmentation (Sun and Xu 2011), POS tagging (Huang, Harper, and Wang 2007; Huang, Eidelman, and Harper 2009), constituency parsing (Wang, Sagae, and Mitamura 2006; Zhang and Clark 2009), and dependency parsing (Zhang and Clark 2008; Huang and Sagae 2010; Li et al. 2011). We use CTB 6.0 as the labeled data for the study. The corpus was collected during different time periods from different sources with a diversity of topics. In order to obtain a representative split of data sets, we conduct experiments following the setting of the CoNLL 2009 shared task. The setting is provided by the principal organizer of the CTB project, and considers many annotation details. This setting is more robust for evaluating Chinese language processing algorithms. Table 1 shows the statistics of our experimental settings.

To deeply analyze the POS tagging problem for Chinese, we implement a linear-chain global linear model. A majority of state-of-the-art English POS taggers are based on LGLMs, for example, structured perceptron (Collins 2002) and conditional random fields (Lafferty, McCallum, and Pereira 2001). We choose structured perceptron (Collins 2002) to estimate parameters.

2.2.2 Features for LLM and LGLM. In our experiments, we use a feature set that draws upon information sources such as word forms and characters that constitute

[1] `www.csie.ntu.edu.tw/~cjlin/liblinear/`.

words. To conveniently illustrate, we denote a word in focus with a fixed window $w_{-2}w_{-1}ww_{+1}w_{+2}$, where w is the current token. Our features includes:

- Word unigrams: w_{-2}, w_{-1}, w, w_{+1}, w_{+2}
- Word bigrams: $w_{-2}\_w_{-1}$, w_{-1}_w, w_w_{+1}, $w_{+1}\_w_{+2}$
- In order to better handle unknown words, we extract morphological features: character n-gram prefixes and suffixes for n up to 3

That means 15 features are used to represent a given word token. When a different amount of data is available, the best configuration of feature template varies. Normally, a larger window of context leads to improved accuracy when more labeled data is available. This setting can be tuned on the development data. In our experiments on the CTB 6.0, the window size is tuned to 2.

2.2.3 Overall Performance. Table 2 summarizes the performance in terms of per word classification of different supervised models on the development data. We present the results of both first- and second-order LGLMs. There is only a slight gap between the local classification model and various structured models. Although the local classifier achieves comparable results when applied to Chinese data, there is a much more significant gap between the corresponding structured models. Similarly, the gap between the first- and second-order LGLMs is very modest too.

2.3 Error Analysis

2.3.1 Correlating Tagging Accuracy with Word Frequency. Table 3 summarizes the prediction accuracy on the development data with respect to the word frequency on the training data. To avoid overestimating the tagging accuracy, these statistics exclude all punctuation that can be easily recognized. From this table, we can see that words with low frequency, especially the out-of-vocabulary (OOV) words, are hard to label. Compared with a generative model, one major advantage of a discriminative model is its ability to utilize flexible features for disambiguation. This is quite important for predicting an unknown word. When a word is very frequently used, its behavior is complicated and therefore hard to predict. A typical example of such words is the language-specific function word "的." This analysis suggests that a main topic to enhance Chinese POS tagging is to bridge the gap between the infrequent words and frequent words.

2.3.2 Correlating Tagging Accuracy with Span Length. In this work, we define the maximal projection of a word x as the span of words below x in the dependency tree. The key

Table 2
Tagging accuracies on the development data. LGLM_1 and LGLM_2 denote first- and second-order global linear model respectively.

System	Accuracy (%)
LLM	93.61
LGLM_1	94.30
LGLM_2	94.42
SR-HMM	94.08

Table 3
Tagging accuracies (%) relative to word frequency.

Freq.	LLM	LGLM$_1$	LGLM$_2$	SR-HMM
0	78.72	79.77	80.66	77.49
1–5	87.75	87.95	88.13	87.57
6–10	90.04	91.04	91.28	90.69
11–100	94.49	94.94	94.80	94.60
101–1000	95.68	96.08	96.12	96.23
1001–	91.81	93.62	93.94	93.41

property is that a word projects its grammatical property to its maximal projection and it syntactically governs all words under the span of its maximal projection. Though maximal projection is traditionaly defined on deep structure by transformational generative grammaticians, we can empirically borrow the idea that a word in a sentence only governs a limited domain. Measuring the area governed by a word is helpful for error analysis. Sometimes modeling such an observation can even improve practical NLP systems such as a semantic role labeller (Sun, Sui, and Wang 2008). The concept of maximal projection used here is adopted from our early work on semantic role labeling (Sun, Sui, and Wang 2008).

Table 4 shows the tagging accuracies relative to the length of the spans. The spans are calculated according to the corresponding dependency annotations converted from CTB and provided by the CoNLL shared task. We can see that with the increase of the number of words governed by the token, the difficulty of its POS prediction increases. Especially, higher-order models make better predictions for words governing larger spans. This analysis suggests that syntagmatic lexical relations play a significant role in POS tagging, and sometimes words located far from the current token significantly affect its tagging.

An interesting phenomenon is that the performance decline stops when the length is greater than 7. The main reason is that these words usually have clear collocation words nearby but the collocated words govern a very large area. Typical examples are words that take a clause as its complement, such as "说/say." It is relatively easy to label this word, but its complement could be of a large size. In other words, the usage of a word that is complex from one particular view is not necessarily complex from another.

Table 4
Tagging accuracies (%) relative to length. The length is defined as one plus the number of words that are dominated by the target word.

Len.	LLM	LGLM$_1$	LGLM$_2$	SR-HMM
1–2	92.77	93.51	93.55	93.37
3–4	91.97	92.94	93.13	92.50
5–6	91.21	92.29	92.51	91.62
7–	93.37	94.17	94.58	93.77

Table 5
Tagging F1 scores relative to POS types.

Type	LLM	LGLM$_1$	LGLM$_2$	SR-HMM
NN	94.51	94.77	94.82	94.32
NR	93.94	94.37	94.90	94.42
NT	97.13	97.41	97.26	97.56
DEC	78.72	81.17	81.89	79.25
DEG	82.35	85.59	86.61	84.38

2.3.3 Correlating Tagging Accuracy with POS Type. Table 5 presents F-scores of several POS types, including nouns and functional words. The POS types *NR*, *NT*, and *NN*, respectively, represent proper nouns, temporal nouns, and other common nouns. We can clearly see that models that only explore local dependencies are good enough to deal with nouns. Superisingly, the local classifier that does not directly define features of possible POS tags of other surrounding words performs even better than structured models for proper nouns and other common nouns.

The tag *DEC* denotes a complementizer or a nominalizer, and the tag *DEG* denotes a genitive marker and an associative marker. These two types only include two words: "的" and "之." The latter is mainly used in ancient Chinese. About 5.19% of words appearing in the training data set is *DEC*/*DEG*. In addition to the high frequency, "的" takes much functional information that is very important for syntactic processing. The pattern of the *DEC* recognition is *clause/verb phrase+DEC+noun phrase*, and the pattern of the *DEG* recognition is *nominal modifier+DEC+noun phrase*. To distinguish the sentential/verbal and nominal modification phrases, the *DEC* and *DEG* words usually need long-range syntactic information for accurate disambiguation. We claim that the prediction performance of the two specific types is a good clue to how well a tagging model resolves long-distance dependencies. We can see that though these taggers work relatively well on predicting content words, they cannot handle function words satisfyingly. The significant performance gap between content words and function words again suggests that syntagmatic lexical relations plays an important role in POS tagging.

3. Capturing Paradigmatic Relations via Word Clustering

To bridge the gap between high- and low-frequency words, we use word clustering to acquire the knowledge about paradigmatic lexical relations from large-scale texts. Our work is also inspired by the successful application of word clustering to named entity recognition (Miller, Guinness, and Zamanian 2004) and dependency parsing (Koo, Carreras, and Collins 2008).

3.1 Word Clustering

Word clustering is a technique for partitioning sets of words into subsets of syntactically or semantically similar words. It is a useful technique to capture paradigmatic or substitutional similarity among words.

3.1.1 Clustering Algorithms. Various clustering techniques have been proposed, some of which, for example, perform automatic word clustering optimizing a maximum-likelihood criterion with iterative clustering algorithms. In this article, we focus on distributional word clustering that is based on the assumption that words that appear in similar contexts (especially surrounding words) tend to have similar syntactic distributions. Note that syntactic rather than morphological distributions are the key evidence to determine the grammatical categories of Chinese words, given that Chinese is an analytic language. Automatic word clustering has been successfully applied to many NLP problems, such as language modeling.

The main problem is that we cannot expect these independently optimized classes to be correspondent with syntactic structures. In the feature induction framework, this problem is partially resolved by exploring the ability of discriminative learning to automatically identify the correspondence between the two types of "word classes." In the literature, contexts have been defined as subjective and objective relations involving the word, as the documents containing the word, or as search engine snippets for the word as a query. We derive new features for POS tagging by applying two distributional clustering methods, which both take into account surrounding words as contexts.

Brown Clustering. Our first choice is the bottom–up agglomerative word clustering algorithm of Brown et al. (1992), which derives a hierarchical clustering of words from unlabeled data. This algorithm generates a hard clustering—each word belongs to exactly one cluster. The input to the algorithm is sequences of words $w_1, ..., w_n$. Initially, the algorithm starts with each word in its own cluster. As long as there are at least two clusters left, the algorithm merges the two clusters that maximize the quality of the resulting clustering. The quality is defined based on a class-based bigram language model as follows.

$$P(w_i|w_1, ...w_{i-1}) \approx p(C(w_i)|C(w_{i-1}))p(w_i|C(w_i)) \tag{7}$$

where the function C maps a word w to its class $C(w)$. We use a publicly available package[2] (Liang, Collins, and Liang 2005) to train this model.

MKCLS Clustering. We also do experiments by using another popular clustering method based on the exchange algorithm (Kneser and Ney 1993). The objective function is maximizing the likelihood $\prod_{i=1}^{n} P(w_i|w_1, ..., w_{i-1})$ of the training data given a partially class-based bigram model of the form

$$P(w_i|w_1, ...w_{i-1}) \approx p(C(w_i)|w_{i-1})p(w_i|C(w_i)) \tag{8}$$

We use the publicly available implementation MKCLS[3] (Och 1999) to train this model.

One downside of both Brown and MKCLS clustering is that they are based solely on bigram statistics, and do not consider word usage in a wider context. We choose to work with these two algorithms considering their prior success in other NLP applications. However, we expect that our approach can function with other clustering algorithms.

[2] `http://cs.stanford.edu/~pliang/software/brown-cluster-1.2.zip`.

[3] `http://code.google.com/p/giza-pp/`.

3.1.2 Data. Chinese Gigaword is a comprehensive archive of newswire text data that has been acquired over several years by the Linguistic Data Consortium (LDC). The large-scale unlabeled data we use in our experiments come from the Chinese Gigaword (LDC2005T14). We choose the Mandarin news text, that is, Xinhua newswire. These data cover all news published by Xinhua News Agency (the largest news agency in China) from 1991 to 2004, which contains over 473 million characters.

3.1.3 Pre-processing: Word Segmentation. Different from English and other Western languages, Chinese is written without explicit word delimiters such as space characters. To find the basic language units (i.e., words), segmentation is a necessary pre-processing step for word clustering. Our previous research showed that character-based segmentation models trained on labeled data are reasonably accurate (Sun 2010). In this work, we use a supervised segmenter introduced in Sun and Xu (2011) to process raw texts.

3.2 Improving Tagging with Cluster Features

Our discriminative sequential tagger is easy to be extended with arbitrary features and therefore suitable to explore additional features derived from other sources. We propose using word clusters as substitutes for word forms to assist the POS tagger. We are relying on the ability of the discriminative learning method to explore informative features, which play central role to boost the tagging performance. Five clustering-based features are added:

- Cluster unigrams: wc_{-1}, wc, wc_{+1}
- Cluster bigrams: wc_{-1}_wc, wc_wc_{+1}

where wc_i denotes the clustering index of word w_i.

Table 6
Tagging accuracies (%) with different feature configurations.

Features	#Sent	Brown			MKCLS		
		LLM	LGLM$_1$	LGLM$_2$	LLM	LGLM$_1$	LGLM$_2$
Baseline		93.61	94.30	94.42	93.61	94.30	94.42
+c100	2.39M	94.30	94.70	94.76	94.48	94.75	94.88
+c500	2.39M	94.41	94.71	94.66	94.59	94.67	94.87
+c1000	2.39M	94.37	94.59	94.77	94.50	94.84	94.86
+c100	7.17M	94.31	94.68	94.88	94.46	94.73	94.80
+c500	7.17M	94.51	94.74	94.86	94.68	94.82	94.95
+c1000	7.17M	94.51	94.74	94.89	94.62	94.77	94.94
+c100	11.96M	94.53	94.81	94.87	94.62	94.83	95.00
+c500	11.96M	94.55	94.75	94.79	94.60	94.87	94.94
+c1000	11.96M	94.66	94.87	94.91	94.66	94.79	94.96

3.3 Evaluation

Table 6 summarizes the tagging results on the development data with different feature configurations. In this table, the symbol "+" in the *Features* column means that the current configuration contains both the baseline features and new cluster-based features; the number is the total number of the clusters; the number in the #*Sent* column means how many millions of raw sentences are used to cluster words. From this table, we can clearly see the impact of word clustering features on POS tagging. The new features lead to substantial improvements over the strong supervised baseline. In particular, the word clustering information bridges the gap between the local classifier and structured prediction models much. Moreover, these increases are consistent regardless of the clustering algorithms. Both clustering algorithms contribute to the overall performance equivalently. A natural strategy for extending current experiments is to include both clustering results together. However, we find no further improvement. For each clustering algorithm, there are not many differences among different sizes of the total clustering numbers. When a small size of unlabeled data is added, the semi-supervised learning only yields minor improvements. When a comparable amount of unlabeled data are used, the further increase of the unlabeled data for clustering does not lead to much changes of the tagging performance.

3.4 Learning Curves

We do additional experiments to evaluate the effect of the derived features as the amount of labeled training data is varied. We use the LGLM_1 model and the clustering results with "MKCLS+11.96M" setting for these experiments. Table 7 summarizes the accuracies of the systems when trained on smaller portions of the labeled data. We can see that the new features obtain consistent gains regardless of the size of the training set. The error is reduced significantly on all data sets. In other words, the word cluster features can significantly reduce the amount of labeled data required by the learning algorithm. The relative reduction is greatest when smaller amounts of the labeled data are used, and the effect lessens as more labeled data are added. This result gives a rough impression of the amount by which derived features reduce the need for supervised data, given a desired level of accuracy.

Table 7
Tagging accuracies (%) relative to sizes of training data. Size = number of sentences in the labeled training corpus. **Bold** identifies best performance at the given size.

Size	Supervised	+c100	+c500	+c1000
100	66.94	**76.78**	71.91	68.59
500	77.89	**84.19**	82.17	81.14
1000	83.70	87.68	**87.74**	86.49
5000	90.57	91.84	91.93	**91.95**
10000	93.17	94.00	93.91	**94.02**
15000	93.76	94.42	**94.47**	94.39
20000	94.11	**94.76**	94.58	94.73

Table 8
Tagging accuracies (%) with IV clustering.

Clusters	+c100	+c500	+c1000
IV	94.37 (↑0.07)	94.41 (↑0.11)	94.40 (↑0.10)
All	94.83 (↑0.46)	94.87 (↑0.46)	94.79 (↑0.39)

Table 9
The tagging recall (%) of OOV words.

Type	#Words	Baseline	+c100	+c500	+c1000
AD	21	42.86	47.62 (↑)	52.38 (↑)	52.38 (-)
CD	237	98.73	98.31 (↓)	99.16 (↑)	98.73 (↑)
JJ	86	26.74	37.21 (↑)	31.40 (↑)	23.26 (↓)
NN	1012	85.47	87.06 (↑)	88.44 (↑)	86.86 (↑)
NR	863	81.23	88.30 (↑)	85.86 (↑)	89.92 (↑)
NT	21	57.14	57.14 (-)	61.90 (↑)	66.67 (↑)
VA	15	40.00	73.33 (↑)	80.00 (↑)	73.33 (↑)
VV	402	69.15	72.14 (↑)	72.89 (↑)	76.37 (↑)

3.5 Analysis

Word clustering derives paradigmatic relational information from unlabeled data by grouping words into different sets. As a result, the contribution of word clustering to POS tagging is two-fold. On the one hand, word clustering captures and abstracts context information. This new linguistic *knowledge* is thus helpful to better correlate a word in a certain context to its POS tag. On the other hand, the clustering of the OOV words to some extent fights the sparse data problem by correlating an OOV word with in-vocabulary (IV) words through their classes. To evaluate the two contributions of the word clustering, we limit entries of the clustering lexicon to only contain IV words, that is, words appearing in the training corpus. Using this constrained lexicon, we train new first-order LGLMs with "+MKCLS+11.96M" clustering and report its prediction power in Table 8. The gap between the baseline and *+IV* models can be viewed as the contribution of the first effect, and the gap between the *+IV* and *+All* models can be viewed as the second contribution. This result indicates that the improved predictive power partially comes from the new interpretation of a POS tag through clustering, and mainly comes from its memory of OOV words that appear in the unlabeled data.

Table 9 shows the recall of OOV words on the development data set. Only the word types appearing more than 10 times are reported. For more information about the definition POS tags, refer to the guideline[4] provided by the CTB project. We give a brief illustration of the POS tags in Appendix A. The results are evaluated using the first-order LGLM tagger. The recall of almost all OOV words is improved with any kind of clustering results, especially of proper nouns (NR) and common verbs (VV). Another

[4] http://www.cs.brandeis.edu/~clp/ctb/posguide.3rd.ch.pdf.

interesting fact is that almost all of them are content words. This table is also helpful to understand the impact of the clustering information on the prediction of OOV words.

4. Capturing Syntagmatic Relations via Parsing

To capture syntagmatic relations among words, a trivial idea is to use higher order Markov models. However, the empirical evaluation on the CTB data indicates that the second-order model does not benefit much, especially when word clustering features are added. This result suggests that a linear-chain structure is relatively weak to capture complex syntagmatic lexical relations. Different from lexical analysis, syntactic analysis, especially the full and deep one, reflects syntagmatic relations of words and phrases of sentences. We present a series of empirical studies of the tagging results of the two syntax-free sequential taggers and a state-of-the-art syntax-based parser, aiming at illuminating more precisely the impact of information about phrase-structures as well as dependency structures on POS tagging. The analysis is helpful to understand the role of syntagmatic lexical relations in POS prediction.

4.1 CFG-Based Parsing

POS tags can be taken as pre-terminals of a constituency parse tree, so a constituency parser can also provide POS information. The majority of the state-of-the-art constituent parsers are based on generative probabilistic CGF (PCFG) learning, with lexicalized (Charniak 2000; Collins 2003) or latent annotation (Matsuzaki, Miyao, and Tsujii 2005; Petrov et al. 2006) refinements. Compared with complex lexicalized parsers, the symbol-refined PCFG (SR-PCFG) parsers leverage on an automatic procedure to learn refined grammars and are more robust to parse many non-English languages that are not well studied. For Chinese, a SR-PCFG parser achieves the state-of-the-art performance and outperforms many other types of parsers (Zhang and Clark 2009). In our work, the Berkeley parser,[5] an open source implementation of the SR-PCFG model, is used for experiments.

4.2 Comparing Tagging and Parsing

From a linguistic view, we can distinguish syntax-free and syntax-based models. In a syntax-based model, POS tagging is integrated into parsing, and thus (to some extent) is capable of capturing a considerable amount of long range syntactic information. From a machine learning view, we can distinguish generative and discriminative models. Compared with generative models, discriminative models define expressive features to classify words. Note that the two generative models use latent variables to refine the output spaces, which significantly boost the accuracy and increase the robustness of simple generative models.

Table 10 shows their overall and detailed performance with respect to representative types. In the following, we present a comparative analysis.

4.2.1 Content Words vs. Function Words. Table 10 gives a detailed comparison regarding different word types. For each type of word, we report the accuracy of both solvers and compare the difference. The majority of the words that are better labeled by the

[5] `code.google.com/p/berkeleyparser/`.

Table 10
Tagging F1 scores of relative to word classes.

Type	LLM	$LGLM_1$	$LGLM_2$	SR-HMM	Parser
NN	94.51	94.77	94.82	94.32	93.46
NR	93.94	94.37	94.90	94.42	89.76
NT	97.13	97.41	97.26	97.56	96.80
CD	97.26	97.57	97.63	97.57	95.50
VA	79.34	83.25	84.49	80.57	81.47
VC	97.10	97.20	97.00	96.90	96.01
AD	93.47	94.53	94.59	94.81	94.13
JJ	82.19	83.80	83.18	82.54	81.38
CC	90.52	91.99	91.98	92.91	94.00
P	93.51	94.52	94.35	95.10	96.19
DEC	78.72	81.17	81.89	79.25	85.69
DEG	82.35	85.59	86.61	84.38	88.94
DER	75.86	77.42	75.00	83.33	78.05
DEV	57.73	74.38	74.14	76.81	84.89
Overall	93.61%	94.30%	94.42%	94.08%	93.69%

tagger are content words, including nouns (NN, NR, NT), numbers (CD), predicates (VA, VC), adverbs (AD), nominal modifiers (JJ), and so on. It is worth noting that both discriminative and generative sequential taggers consistently outperform the parser. In contrast, most of the words that are better predicted by the parser are function words, including most particles (DEC, DEG, DER, DEV, AS, MSP), prepositions (P), and coordinating conjunctions (CC).

4.2.2 Open Classes vs. Close Classes. POS can be divided into two broad supercategories: closed class types and open class types. Open classes accept the addition of new morphemes (words), through such processes as compounding, derivation, inflection, coining, and borrowing. On the other hand closed classes are those that have relatively fixed membership. For example, nouns and verbs are open classes because new nouns and verbs are continually coined or borrowed from other languages, whereas *DEC/DEG* are two closed classes because only the function word "的" is assigned to them. The discriminative model can conveniently include many features, especially features related to the word formation, which are important to predict words of open classes. Table 11 summarizes the tagging accuracies relative to IV and OOV words. These statistics exclude all punctuations that can be trivially recognized. On the whole, the Berkeley parser processes IV words slightly better than our tagger, but processes OOV words significantly worse. The numbers in this table clearly show that the main weakness of the Berkeley parser is the the predictive power of the OOV words.

4.2.3 Local Disambiguation vs. Global Disambiguation. Closed class words are generally function words that tend to occur frequently and often have structuring uses in grammar. These words have little lexical meaning or have ambiguous meaning, but instead

Table 11
Tagging accuracies (%) of the IV and OOV words.

	IV	OOV
LLM	93.56	78.72
LGLM$_1$	94.35	79.77
LGLM$_2$	94.43	80.66
SR-HMM	94.22	77.49
Parser	94.61	64.43

serve to express grammatical relationships with other words within a sentence. They signal the structural relationships that words have to one another and are the glue that holds sentences together. Thus, they serve as important elements to the structures of sentences. The disambiguation of these words normally requires more syntactic clues, which are very hard and inappropriate for a sequential tagger to capture. Based on global grammatical inference of the whole sentence, the full parser is relatively good at dealing with structure-related ambiguities.

We conclude that a discriminative sequential tagging model can better capture local syntactic and morphological information, and the full parser can better capture global syntactic structural information. The discriminative tagging models are limited by the Markov assumption and are inadequate to correctly label structure-related words.

4.3 Impact on Parsing

The weak ability for non-local disambiguation also imposes restrictions on using a sequence POS tagging model as a front module for parsing. To evaluate the impact, we use the Berkeley parser to parse a sentence based on the POS tags provided by sequence models. Table 12 shows the parsing performance. Labeled bracketing precision, recall, and F-score (LP, LR, and LF) are listed. Note that the overall tagging performance of the Berkeley parser is significantly worse than the sequence models. However, better POS tagging does not lead to better parsing. Our experiments suggest that sequence models

Table 12
Parsing accuracies (%) on the development data.

Devel.	LP	LR	LF
Berkeley(ALL)	82.44	80.31	81.36
LLM(ALL)	79.17	78.46	78.82 (↓)
LGLM$_1$(ALL)	80.28	79.52	79.90 (↓)
LGLM$_2$(ALL)	79.59	80.58	80.08 (↓)
SR-HMM(ALL)	80.59	79.35	79.96 (↓)
LLM(non-De)+Berkeley(De)	81.12	79.18	79.64 (↓)
LGLM$_1$(non-De)+Berkeley(De)	80.82	79.94	80.38 (↓)
LGLM$_2$(non-De)+Berkeley(De)	81.14	80.07	80.60 (↓)
SR-HMM(non-De)+Berkeley(De)	81.32	80.01	80.66 (↓)

propagate too many errors to the parser. Moreover, the parser is very sensitive to errors of prediction of some specific categories. The numbers presented in the bottom block of Table 12 give a rough illustration. The results are obtained by providing the parser *mixed* POS tagging analysis: The tags of "的/得" predicted by the Berkeley parser and the tags of other words are utilized. We can see that though the overall parsing quality is still worse than Berkeley parser, it is better than sequence models. The performance change demonstrates the importance of prediction of these two particular words. Our linguistic analysis can also better explain the poor performance of Chinese CCG parsing when applying the C&C parser (Tse and Curran 2012). We think the failure is mainly due to overplaying sequence models in both POS tagging and supertagging.

4.4 Enhancing Tagging via Stacking

We study a simple way of integrating multiple heterogeneous models in order to exploit their complementary strengths and thereby improve tagging accuracy beyond what is possible by either model in isolation. The method integrates the heterogeneous models by allowing the outputs of SR-HMM and the parser to define features for the LLM/LGLM. Similar to our work on combining a sequence model and a parser, Rush et al. (2010) proposed a principled decoding technique based on dual decomposition to take advantages of heterogeneous models. There are two differences between their model and ours. First, the base models for combining are separately trained in their solution. In other words, one key difference is whether to allow integration of base models at learning time. Second, the application of the decomposition technique is dependent on the solvability of sub-problems. This technique, therefore, is not as flexible as stacking.

4.4.1 Stacked Learning. **Stacked generalization** is a meta-learning algorithm that was first proposed in Wolpert (1992) and Breiman (1996). Stacked learning has been applied as a system ensemble method in several NLP tasks, such as joint word segmentation and POS tagging (Sun 2011), and dependency parsing (Nivre and McDonald 2008). The idea is to include two "levels" of predictors. The first level includes one or more predictors $g_1, ..., g_K : \mathbb{R}^d \rightarrow \mathbb{R}$; each receives input $\mathbf{x} \in \mathbb{R}^d$ and outputs a prediction $g_k(\mathbf{x})$. The second level consists of a single function $h : \mathbb{R}^{d+K} \rightarrow \mathbb{R}$ that takes as input $\langle \mathbf{x}, g_1(\mathbf{x}), ..., g_K(\mathbf{x}) \rangle$ and outputs a final prediction $\hat{y} = h(\mathbf{x}, g_1(\mathbf{x}), ..., g_K(\mathbf{x}))$. The predictor, then, combines an ensemble (the g_k's) with a meta-predictor (h).

Training is done as follows. The training data $S = \{(\mathbf{x}_t, \mathbf{y}_t) : t \in [1, T]\}$ are split into L equal-sized disjoint subsets $S_1, ..., S_L$. Then functions $\mathbf{g}_1, ..., \mathbf{g}_L$ (where $\mathbf{g}_l = \langle g_1^l, ..., g_K^l \rangle$) are separately trained on $S - S_l$, and are used to construct the augmented data set $\hat{S} = \{(\langle \mathbf{x}_t, \hat{\mathbf{y}}_t^1, ..., \hat{\mathbf{y}}_t^K \rangle, \mathbf{y}_t) : \hat{\mathbf{y}}_t^k = g_k^l(\mathbf{x}_t) \text{ and } \mathbf{x}_t \in S_l\}$. Finally, each g_k is trained on the original data set and the second level predictor h is trained on $\hat{S}$. The intent of the *cross-validation* scheme is that $\mathbf{y}_t^k$ is similar to the prediction produced by a predictor which is learned on a sample that does not include $\mathbf{x}_t$.

This framework is also explored as a solution for learning *long range* features in Torres Martins et al. (2008). Torres Martins et al. explored a stacked framework for learning *long range* features for dependency parsing. In machine learning research, stacked learning has been applied to structured prediction (Cohen and Carvalho 2005). In this work, stacked learning is used to acquire extended training data for sub-word tagging. For example, Cohen and Carvalho (2005) described a sequential learning scheme called "stacked sequential learning." In that meta-learning algorithm, an arbitrary base learner is augmented so as to make it aware of the labels of nearby examples.

4.4.2 Applying Stacking to POS Tagging. We use the LLMs or LGLMs (as h) for the level-1 processing, and other models (as g_k) for the level-0 processing. The characteristic of discriminative learning makes LLMs/LGLMs very easy to integrate into the outputs of other models as new features. T is set to 5 to generate augmented training data for estimating h. We are relying on the ability of discriminative learning to explore informative features, which play a central role in boosting the tagging accuracy. For output labels produced by each auxiliary model, five new *label uni/bigram* features are added: w_{-1}, w, w_{+1}, w_{-1}_w, w_w_{+1}. This choice is tuned on the development data.

4.4.3 Evaluation. Table 13 summarizes the tagging accuracy of different stacking models. From this table, we can clearly see that the new features derived from the outputs of other models lead to substantial improvements over the baseline LLM/LGLM. The output structures provided by the SR-PCFG model is most effective in improving the LLM/LGLM baseline systems. Among different stacking models, the syntax-free hybrid one (i.e., stacking LLM/LGLM with SR-HMM) does not need any treebank to train their systems. For the situations in which parsers are not available, this is a good solution. Moreover, the decoding algorithms for linear-chain Markov models are very fast. Therefore the syntax-free hybrid system is more appealing for many NLP applications.

Table 14 shows the F1 scores of the DEC/DEG prediction obtained by different stacking models. Compared with Table 10, we can see that the hybrid sequence model is still not good at handling long-distance ambiguities. As a result, it still does not serve the parser well, though it achieves higher overall precision. On the other hand, the syntax-based hybrid model can refine the POS tags returned by the same parser, and therefore improve the final parsing results. In other words, by parsing twice, we can obtain better phrase-structure trees.

Table 13
Tagging accuracies (%) of different stacking models on the development data.

	LLM	LGLM$_1$	LGLM$_2$
+SR-HMM	94.59	94.80	94.83
+SR-PCFG	94.83	95.06	95.04
+Word clustering+SR-HMM	95.03	95.11	95.12
+Word clustering+SR-PCFG	95.40	95.45	95.54

Table 14
F1 score of the *DEC/DEG* prediction and parsing performance of different stacking models on the development data.

	DEC	DEG	LP	LR	LF
LGLM$_1$(SR-HMM)	82.72	86.99	81.13%	80.12%	80.63 (↓)
LGLM$_1$(SR-PCFG)	87.05	90.06	82.66%	81.20%	81.92 (↑)

4.5 Combining Both

We have introduced two separate improvements for Chinese POS tagging that capture different types of lexical relations. We therefore expect further improvement by combining both enhancements, since their contributions to the task is different. We still use the stacking model to integrate the discriminative tagger and the Berkeley parser. The only difference between current experiment and the previous experiment is that the discriminative models are trained with the help of word clustering features. The last line of Table 13 also shows the performance of the new hybrid models on the development data set. We can see that the improvements that come from two methods, namely, capturing syntagmatic and paradigmatic relations, do not overlap much and their combination yields a better combined result.

5. Reducing Hybrid Models to Sequence Models

We have shown that higher accuracy can be achieved by applying learning techniques to capture deep lexical relations. Especially, syntagmatic lexical relations have been shown playing an essential role in Chinese POS tagging. To capture such relations, we utilize hybrid models that obtain such information from a syntactic parser. However, it is inappropriate to use computationally expensive parsers to improve POS tagging for many realistic NLP applications, mainly because of efficiency considerations. In this section, we investigate the feasibility of capturing some longer-distance dependencies in a sequence model.

5.1 The Idea

We explore unlabeled data to transfer the predictive power of hybrid models to sequence models. The main idea behind this is to use a fast model to approximate the function learned by a slower, larger, but better-performing ensemble model. Unlike the true function that is unknown, the function learned by a high-performing model is available and can be used to label large amounts of pseudo data. A fast and expressive model trained on large-scale pseudo data will not overfit and will approximate the function learned by the high performing model well. This allows a slow, complex model such as a massive ensemble to be compressed into a fast sequence model such as a first-order LGLM with very little loss in performance.

This idea to use unlabeled data to transfer the predictive power of one model to another has been investigated in many areas, for example, from high accuracy neural networks to more interpretable decision trees (Craven 1996), from high accuracy ensembles to faster and more compact neural networks (Bucila, Caruana, and Niculescu-Mizil 2006), from structured prediction models to local classification models (Liang, Daumé, and Klein 2008), or from complicated parsing models to simpler ones (Petrov et al. 2010).

5.2 Applying Structured Compilation to POS Tagging

We do some experiments to explore the feasibility of reducing hybrid tagging models to a SR-HMM or LGLM for Chinese POS tagging. The large-scale unlabeled data we use in our experiments come from the Chinese Gigaword. We choose the Mandarin news text (i.e., Xinhua newswire). We tag Gigaword sentences by applying a successful model, namely, the stacked second order LGLMs with Berkeley parser. According to our evaluation, the automatically annotated texts obtained in this step is of relatively high quality. Note that the process of this step is somewhat time-consuming, given that a chart parser is used. We viewed the annotated results as **pseudo training data**, which is imperfect but still of high quality. Such pseudo training data could be of very large-scale theoretically and practically. Together with gold standard training data, large-scale pseudo training data can be used to train SR-HMMs and LGLMs. We expect that the SR-HMM tagger can be improved by exploring better latent variables, and that the discriminative taggers can be improved by using features in a larger context.

5.3 Beam Decoding for LGLM

For a number of NLP tasks, including tagging and parsing, the generic beam-search algorithmic technique has been shown to be very powerful to build efficient systems with comparable accuracy (Zhang and Clark 2011). In our model re-compilation case, to train a second order LGLM on a very large data set is quite time-consuming. Rather than the Viterbi algorithm, we here use the beam search algorithm for decoding. Beyond simple beam decoding that essentially implements the greedy search strategy, Huang and Sagae (2010) discuss how the state-merging strategy that is used by dynamic programming methods can be applied to enhance a beam decoder. Considering that the total number of possible tags is much larger than conventional tagging, we implement a beam-search algorithm with state merging for our discriminative tagger.

In a second-order model, the basic factor contains three consecutive tags, but only the last two influence future decoding. That means that all partial tag sequences with the same last two tags can be merged together. Specifically, at each decoding step, our decoder first generates all new partial tag sequences by labeling the next word, then the top-*b* sequences with different last two tags are collected for future prediction while others are thrown away. With the state-merging strategy, our beam decoder can perform dynamic programming too. Note that when the beam width is large enough, our decoding algorithm actually searches the whole space and is exactly a Viterbi decoder.

5.4 Multi-View Learning with Unlabeled Data

The key for the success of hybrid tagging models is the existence of a large diversity among learners. Zhou (2009) argued that when there are many labeled training examples, unlabeled instances are still helpful for hybrid models because they can help to increase the diversity among the base learners. The author also briefly introduced a preliminary theoretical study. In this work, we also combine the re-trained models to see if we can benefit more. The final combination is very simple: We utilize voting as the strategy for final combination. In the tagging phase, the re-trained LGLM and SR-HMM systems with different settings output multiple tagging results, in which each word is assigned one POS label. The final tagging is the voting result of these labels.

5.5 Evaluation

5.5.1 Reducing Hybrid Models to SR-HMMs. With the increase of (pseudo) training data, a SR-HMM may learn better latent variables to subcategorize POS tags, which could significantly improve a purely supervised SR-HMM. In our experiments, SR-HMM models are trained with six, seven, and eight iterations of split, merge, smooth. Table 15 shows the performance of the re-trained SR-HMMs. The first column is the number of sentences of pseudo sentences, and the second column lists the number of words. The pseudo sentences are selected from the Xinhua news section of the Chinese Gigaword. We can clearly see that the idea to leverage unlabeled data to transfer the predictive

Table 15
Tagging accuracies (%) of re-compiled SR-HMM models on the development data. "I-x" denotes the number (x) of split-merge-smooth iterations for training. **Bold** identifies best performance results.

		I-6	I-7			I-8		
#Sent	#Word	Overall	Overall	DEC	DEG	Overall	DEC	DEG
100K	2.36M	94.27	94.46	80.76	85.82	94.64	81.27	86.34
200K	4.72M	94.51	94.65	80.21	85.29	94.70	81.72	86.39
500K	11.82M	94.57	94.75	80.30	85.26	94.78	81.62	86.57
1000K	23.63M	94.79	94.87	81.26	86.32	**94.96**	**82.09**	**88.95**

ability of the hybrid model works. Self-training can also slightly improve a SR-HMM (Huang, Eidelman, and Harper 2009). Our auxiliary experiments show that self-training is not as effective as our structure compilation method.

With the increase of training iterations, finer-grained latent variables are estimated and they can enhance tagging. Note that the training procedure on the purely supervised setting obtains the best tagging results at iteration 6. More training data, even if it is not perfect, can improve the generative learning process. The table also presents the performance with respect to DEC/DEG disambiguation. The results suggest that finer-grained latent variables lead to better long-range disambiguation.

5.5.2 Reducing Hybrid Models to LGLMs. To increase the expressive power of a discriminative classification model, we extend the feature templates. This strategy is proposed by Liang, Daumé, and Klein (2008). In our experiments, we increase the window size of word uni-/bigram features to approximate longer distance dependencies. For window size 3, we will add w_{-3}, w_3, $w_{-3}w_{-2}$, and w_2w_3 as new features; for size 4, we will add w_{-4}, w_{-3}, w_3, w_4, $w_{-4}w_{-3}$, $w_{-3}w_{-2}$, w_2w_3, and w_3w_4. Using features derived from a longer window is harmful when only limited labeled data are available. That is why

Table 16
Tagging accuracies (%) of re-compiled LGLM$_1$ and LGLM$_2$ models on the development data. The beam size is set to 4. "win=x" denotes the window size (x) of word uni-/bigrams for feature extraction. **Bold** identifies best performance results.

			win=2	win=3			win=4		
#Sent	#Word	Model	Overall	Overall	DEC	DEG	Overall	DEC	DEG
100K	2.36M	LGLM$_1$	94.98	95.15	83.99	87.53	95.14	84.05	87.68
200K	4.72M		95.03	95.21	84.70	88.25	95.19	85.12	88.46
500K	11.82M		95.14	95.24	85.17	88.94	95.27	86.02	89.31
1000K	23.63M		95.20	95.24	84.89	89.03	95.30	86.50	**89.91**
100K	2.36M	LGLM$_2$	95.09	95.18	84.84	88.10	95.09	84.70	88.14
200K	4.72M		95.13	95.15	84.96	87.96	95.24	86.08	89.76
500K	11.82M		95.22	95.23	85.51	88.79	95.27	85.51	89.01
1000K	23.63M		95.24	95.30	85.28	89.04	**95.40**	**86.59**	89.85

we only use these features in the structure compilation setting. Table 16 shows the performance of the re-compiled first- and second-order LGLMs. The "+MKCLS+11.96M" algorithm is used to provide word clustering information, and the number of total clusters is 500. Similar to the generative model, the discriminative LGLM tagger can be improved too. The second-order model performs slightly better than the first-order one. Considering the decoding time is equivalent because of the fixed beam width, the second-order model is a better choice for application.

In these experiments, we set the beam width for decoding to be 4. Our auxiliary experiments shows that the "beam search with state merging" is quite effective, even with a very small beam size. We vary the beam width and present the results in Table 17.

Compared with the generative model, the re-compiled discriminative model is more effective and more efficient. Although the time complexity for the SR-HMM is linear with respect to the number of words contained in a sentence, the practical running time is influenced by the number of latent variables. Even if we expect further accuracy improvements via adding more data and using more split-merge-smooth iterations to get more effective latent variables, such a setting will significantly affect the tagging efficiency. On the other hand, the beam decoder for the discriminative model achieves equivalent tagging accuracies to the Viterbi decoder. As a result, the efficiency of both training and testing can be guaranteed. Another advantage of the discriminative tagger is its relatively good prediction power of the longer-distance dependencies. The best re-compiled LGLM$_2$ obtains better DEC/DEG prediction than the Berkeley parser.

5.5.3 Voting. Table 18 is the final voting results of the SR-HMM and LGLM. We use three base models for combination, which is the minimum for performing voting. In other words, the final tagging is the voting result of these three labels. Obviously, the re-trained models are still diverse and complementary, so the voting can further improve the sequence models. The result of the best hybrid sequence model is equivalent to the best stacking models.

Table 17
Tagging accuracies (%) relative to beam width on the development data. The LGLM$_2$ model is applied.

#Sent	Beam	win=2	win=3	win=4
500K	8	95.19	95.31	95.26
500K	16	95.18	95.31	95.31
500K	32	95.22	95.25	95.34
500K	64	95.20	95.30	95.27

Table 18
Tagging accuracies (%) of the voting models on the development data. **Bold** identifies best performance results.

Voter 1	Voter 2	Voter 3	Acc.
SR-HMM, I-8, 1000K	SR-HMM, I-7, 1000K	LGLM, win=4, 1000K	95.17
SR-HMM, I-8, 1000K	LGLM, win=4, 1000K	LGLM, win=3, 1000K	95.45
SR-HMM, I-8, 1000K	LGLM, win=4, 1000K	LGLM, win=4, 500K	**95.54**

Table 19
Accuracies (%) of parsing based on re-compiled tagging. Column "SC" denotes whether structure compilation is applied.

	SC	LP	LR	LF
Berkeley	- -	82.44	80.31	81.36
SR-HMM	NO	80.59	79.35	79.96 (↓)
LGLM$_2$	NO	79.59	80.58	80.08 (↓)
SR-HMM	YES	82.86	80.60	81.22 (↓)
LGLM$_2$	YES	82.50	81.39	81.94 (↑)
Voting	YES	82.57	81.47	82.01 (↑)

5.5.4 Improved Parsing. There are two ways for the sequence models to encode long-range information. On one hand, the models can be built upon high-order linear structures (e.g., Ye et al. 2009). One of main challenges of this solution is the high computational complexity. On the other hand, sequence models can incorporate features extracted from a larger context (e.g., by extending window size). This solution cannot work well if only a limited amount of annotated data is available. The key idea underlying structure compilation is to appropriately utilize automatically annotated data to estimate weights for more contextual features. Because features extracted from a larger context provide important clues to detect longer-distance relationships, a re-compiled sequence model can approximate the behavior of a parser to some extent.

Purely supervised sequence models are not good at predicting function words, and accordingly are not good enough to be used as front modules to parsers. The re-compiled models can mimic some behaviors of parsers, and therefore are suitable for parsing. Especially, we have seen that the predictive power for the function word disambiguation is enhanced significantly. Our evaluation shows that the significant improvement of the POS tagging stop harming syntactic parsing. Results in Table 19[6] indicate that the parsing accuracy of the Berkeley parser can be simply improved by inputting the Berkeley parser with the re-trained sequential tagging results. Additionally, the success to separate tagging and parsing can improve the efficiency of the syntactic processing.

5.5.5 Final Results. Table 20 shows the performance of different systems evaluated on the test data. Our final sequence model achieves the state-of-the-art performance, which is obtained by combining a state-of-the-art parser as well as sequence models.

5.5.6 Comparison with Other Taggers. We compare our final sequence labeling based tagger to other representative taggers. Though most research papers report experiments on CTB, they usually define different training/developement/test sets. Nevertheless, numeric performance still reflects accuracy level of existing systems and our tagger. The first three taggers for comparison are based on the joint POS tagging and dependency parsing architecture, which is able to leverage on rich syntactic information to capture syntagmatic relations. They also use global linear models for disambiguation, given

[6] Some relevant information is copied from Table 12.

Table 20
Tagging accuracies (%) on the test data.

System		Acc.
Baseline	$LGLM_2$	94.29
Hybrid	$LGLM_1$(SR-PCFG)+c500	95.32
Re-compiled	Voting	95.34

that such discriminative learning method achieves state-of-the-art for both tagging and parsing. The major difference between these three taggers is the corresponding parsing approach: They apply transition-based, graph-based and easy-first methods, respectively. Table 21 presents the results. We can see our re-compiled tagger achieves significantly better results, though it utilizes a simpler technique (i.e., sequence labeling) and does not explicitly use syntactic information.

Very recently, neural networks have been widely applied various NLP tasks, including word segmentation (Chen et al. 2015; Ma and Hinrichs 2015), syntactic parsing (Chen and Manning 2014; Weiss et al. 2015), and machine translation (Devlin et al. 2014). We also compare our tagger with a neural network–based tagger. Alberti et al. (2015) introduced a neural network–based joint tagging and parsing model that obtains state-of-the-art results on multiple languages. Table 22 shows the results. Because their experiments used the data from CoNLL 2009 shared task, their results are directly comparable to ours. We can see that our final tagger is significantly better than this currently developed neural network–based system.

Table 21
Comparison with other taggers. Tagging accuracies are all evaluated on CTB, but different training and test data sets are used.

System	Architecture	Learning	Acc. (%)
Ours	Sequential Tagging	Linear	95.34
Hatori et al. 2011	Transition-based Joint Tagging & Parsing	Linear	94.01
Li et al. 2011	Graph-based Joint Tagging & Parsing	Linear	93.08
Ma et al. 2012	Easy-first Joint Tagging & Parsing	Linear	94.27

Table 22
Comparison with other taggers. Tagging accuracies are obtained on the test data of CoNLL 2009 shared task.

System	Architecture	Learning	Acc. (%)
Ours	Sequential Tagging	Linear	95.34
Alberti et al. 2015	Transition-based Joint Tagging & Parsing	Neural	94.62

6. Related Work

Many successful tagging algorithms designed for English have been applied to many other languages as well. In some cases, the methods work well without large modifications, such as for German POS tagging. But a number of augmentations and changes became necessary when dealing with highly inflected or agglutinative languages, as well as analytic languages, of which Chinese is the focus of this article.

Both discriminative and generative models are explored for accurate Chinese POS tagging (Ng and Low 2004; Tseng, Jurafsky, and Manning 2005; Huang, Harper, and Wang 2007; Huang, Eidelman, and Harper 2009). Ng and Low (2004) and Tseng et al. (2005) introduced a maximum entropy–based model, which includes morphological features for unknown word recognition. Huang, Harper, and Wang (2007) and Huang, Eidelman, and Harper (2009) mainly focused on the generative HMM models. To enhance a trigram HMM model, Huang, Harper, and Wang (2007) proposed a re-ranking procedure to include both morphology and syntactic structure features, which is difficult to capture for a generative model. Different from the discriminative re-ranking strategy, Huang, Eidelman, and Harper (2009) proposed a latent variable incorporated model to improve a bigram HMM model.

Recently, researchers developed several models that integrate tagging into parsing (Hatori et al. 2011; Li et al. 2011; Bohnet and Nivre 2012; Ma et al. 2012; Alberti et al. 2015). The joint decoding architecture on one hand allows tagging to use rich syntactic features to improve accuracy, but on the other hand decreases the decoding efficiency. Different from the joint tagging and parsing approach, our method does not explicitly use syntactic features in the tagging phase. Only a simple sequence labeler with beam search is applied and therefore our tagger is much more efficient.

Our work also borrows some ideas from investivations in Chinese word segmentation. Notably, the idea to harvest string knowledges from large-scale raw texts to define new features for disambiguation is also successfully applied in our early work on semi-supervised segmentation (Sun and Xu 2011). Recently, neural network models have been widely applied to induce various linguistic knowledges in an unsupervised learning fashion. Such models have also been applied to word segmentation (Zheng, Chen, and Xu 2013; Chen et al. 2015; Ma and Hinrichs 2015). As an alternative way to exploit unlabeled data, neural network models can be also applied in our solution.

7. Conclusion

Chinese POS tagging has been proven much more challenging because of language-specific properties. We hold a view of structuralist linguistics and study the impact of paradigmatic and syntagmatic lexical relations on Chinese POS tagging. First, we harvest word partition information from large-scale raw texts to capture paradigmatic relations and use such knowledge to enhance a supervised tagger via feature engineering. Second, we comparatively analyze syntax-free and syntax-based models and use a stacking model to integrate a sequential tagger and a chart parser to capture syntagmatic relations that have a great impact on non-local disambiguation. Both enhancements significantly improve the state-of-the-art of Chinese POS tagging. The final model results in an error reduction of 18% over a state-of-the-art baseline. To improve tagging efficiency at test time, we explore unlabeled data to transfer the predictive power of hybrid models to simple sequence or even local classification models. Hybrid systems are utilized to create large-scale pseudo training data for cheap models. By applying complex machine learning techniques, we are able to build good sequential

Table A.1
Illustration of some POS tags mentioned in Tables 9 and 10.

AD	AD represents various adverbs that modify predicate or sentential phrases.
CC	CC represents coordinating conjunctions.
CD	CD represents cardinal numbers.
DEC	DEC represents complementizers or nominalizers.
DEG	DEG represents genitive markers or associative markers.
DER	DER denotes the specific word “得” when it locates before a resultative phrase.
DEV	DEV denotes the specific word “地” when it locates in between a predicate and its modifier.
JJ	JJ represents prepositions.
NN	NN represents common nouns.
NR	NR represents proper nouns that are a subclass of nouns.
NT	NT represents temporal nouns.
P	P represents prepositions.
VA	VA represents predicative adjectives which roughly correspond to adjective in English.
VC	VA represents copula words.
VV	VV represents common verbs.

POS taggers. Another advantage of our system is that it serves as a front-end to a parser very well, and more accurate POS tagging yields more accurate phrase-structure parsing.

Appendix A. POS Tags Used in This Article

The CTB utilizes syntactic distribution as the main criterion for distinguishing lexical categories. In Table A.1, we present a brief introduction to the POS tags mentioned in Table 9 and 10. For more details, refer to the original annotation guidelines.

Acknowledgments
This work was supported by the National Natural Science Foundation of China under grants no. 61300064 and 61331011, and the National High-Tech R&D Program under grant no. 2015AA015403. We are very grateful to the anonymous reviewers for their insightful and constructive comments and suggestions.

References

Alberti, Chris, David Weiss, Greg Coppola, and Slav Petrov. 2015. Improved transition-based parsing and tagging with neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1354–1359, Lisbon.

Bohnet, Bernd and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island.

Breiman, Leo. 1996. Stacked regressions. *Machine Learning*, 24:49–64.

Brown, Peter F., Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18:467–479.

Bucila, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, Philadelphia, PA.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the First Conference of the North American Chapter of the Association for*

Computational Linguistics, pages 132–139, Seattle, WA.

Chen, Danqi and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha.

Chen, Xinchi, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015. Gated recursive neural network for Chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1744–1753, Beijing.

Cohen, William W. and Vitor R. Carvalho. 2005. Stacked sequential learning. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 671–676, San Francisco, CA.

Collins, Michael. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Philadelphia, PA.

Collins, Michael. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Craven, Mark. 1996. *Extracting Comprehensible Models from Trained Neural Networks*. Ph.D. thesis, University of Wisconsin–Madison, Department of Computer Sciences. Also appears as UW Technical Report CS-TR-96-1326.

Devlin, Jacob, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, MD.

Giménez, Jesús and Lluís Màrquez. 2004. Svmtool: A general POS tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 43–46, Lisbon.

Hatori, Jun, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint POS tagging and dependency parsing in Chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1216–1224, Chiang Mai.

Huang, Liang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–151, Montreal.

Huang, Liang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086, Uppsala.

Huang, Zhongqiang, 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, Co.

Huang, Zhongqiang, Mary Harper, and Wen Wang. 2007. Mandarin part-of-speech tagging and discriminative reranking. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1093–1102, Prague.

Kneser, Reinhard and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modeling. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, pages 973–976, Berlin.

Koo, Terry, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, OH.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA.

Li, Zhenghua, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for Chinese POS tagging and dependency parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1180–1191, Edinburgh.

Liang, Percy, Michael Collins, and Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, MIT. Cambridge, MA.

Liang, Percy, Hal Daumé, III, and Dan Klein. 2008. Structure compilation: Trading structure for features. In *Proceedings of the 25th International Conference on Machine Learning*, pages 592–599, New York, NY.

Ma, Ji, Tong Xiao, Jingbo Zhu, and Feiliang Ren. 2012. Easy-first Chinese POS tagging and dependency parsing. In *Proceedings of COLING 2012*, pages 1731–1746, Mumbai.

Ma, Jianqiang and Erhard Hinrichs. 2015. Accurate linear-time Chinese word segmentation via embedding matching. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1733–1743, Beijing.

Matsuzaki, Takuya, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 75–82, Stroudsburg, PA.

Miller, Scott, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *HLT-NAACL 2004: Main Proceedings Association for Computational Linguistics*, pages 337–342, Boston, MA.

Ng, Hwee Tou and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? In *Proceedings of EMNLP 2004*, pages 277–284, Barcelona.

Nivre, Joakim and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, OH.

Och, Franz Josef. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 71–76, Stroudsburg, PA.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney.

Petrov, Slav, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 705–713, Cambridge, MA.

Petrov, Slav and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 404–411, Rochester, NY.

Rush, Alexander M., David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Cambridge, MA.

Shen, Libin, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 760–767, Prague.

Shindo, Hiroyuki, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. 2012. Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–448, Jeju Island.

Sun, Weiwei. 2010. Word-based and character-based word segmentation models: Comparison and combination. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1211–1219, Beijing.

Sun, Weiwei. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1385–1394, Portland, OR.

Sun, Weiwei, Xiaochang Peng, and Xiaojun Wan. 2013. Capturing long-distance dependencies in sequence models: A case study of chinese part-of-speech tagging. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 180–188, Nagoya.

Sun, Weiwei, Zhifang Sui, and Haifeng Wang. 2008. Prediction of maximal projection for semantic role labeling. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 833–840, Manchester.

Sun, Weiwei and Hans Uszkoreit. 2012. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate Chinese part-of-speech tagging. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 232–241, Jeju Island.

Sun, Weiwei and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 970–979, Edinburgh.

Sun, Xu. 2014. Structure regularization for structured prediction. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pages 2402–2410.

Torres Martins, André Filipe, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 157–166, Honolulu, HI.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 173–180, Stroudsburg, PA.

Tse, Daniel and James R. Curran. 2012. The challenges of parsing Chinese with combinatory categorial grammar. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 295–304, Montréal.

Tseng, Huihsin, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168–171, Jeju Island.

Tseng, Huihsin, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help POS tagging of unknown words across language varieties. In *The Fourth SIGHAN Workshop on Chinese Language Processing*, pages 32–39, Jeju Island.

Wang, Mengqiu, Kenji Sagae, and Teruko Mitamura. 2006. A fast, accurate deterministic parser for Chinese. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 425–432, Sydney.

Weiss, David, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing.

Wolpert, David H. 1992. Original contribution: Stacked generalization. *Neural Networks*, 5:241–259.

Xue, Naiwen, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11:207–238.

Ye, Nan, Wee S. Lee, Hai L. Chieu, and Dan Wu. 2009. Conditional random fields with high-order features for sequence labeling. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., pages 2196–2204.

Zhang, Yue and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu, HI.

Zhang, Yue and Stephen Clark. 2009. Transition-based parsing of the Chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 162–171, Paris.

Zhang, Yue and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.

Zheng, Xiaoqing, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, WA.

Zhou, Zhi-Hua. 2009. When semi-supervised learning meets ensemble learning. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 529–538, Berlin.