Capturing Long-distance Dependencies in Sequence Models: A Case Study of Chinese Part-of-speech Tagging

Weiwei Sun and **Xiaochang Peng** and **Xiaojun Wan**
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics
{ws,wanxiaojun}@pku.edu.cn; pxc.pku@gmail.com

Abstract

This paper is concerned with capturing long-distance dependencies in sequence models. We propose a two-step strategy. First, the stacked learning technique is applied to integrate sequence models that are good at exploring local information and other high complexity models that are good at capturing long-distance dependencies. Second, the structure compilation technique is employed to transfer the predictive power of hybrid models to sequence models via large-scale unlabeled data. To investigate the feasibility of our idea, we study Chinese POS tagging. Experiments on the Chinese Treebank data demonstrate the effectiveness of our methods. 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

Sequential classification models provide very important solutions to pattern recognition tasks that involve the automatic assignment of a categorical label to each token of a sequence of observed values. A common example is part-of-speech (POS) tagging, which seeks to assign a grammatical category to each word in an input sentence. *Standard* machine learning algorithms to sequential tagging, e.g. linear-chain conditional random fields and max-margin Markov network, directly exploit local dependencies and perform quite well for a large number of sequence labeling tasks. In these models, usually, the relationships between two (or three) successive labels are parameterized and encoded as a single feature, and Viterbi style dynamic programming algorithms are applied to inference over a lattice. Although sequence models perform well for many applications, they are inadequate for tasks where many long-distance dependencies are involved.

Sequential classification models play an important role in natural language processing (NLP). Several fundamental NLP tasks, including named entity recognition, POS tagging, text chunking, supertagging, etc., employ sequential classifiers for lexical and syntactic disambiguation. In addition to learning linear chain structures, sequence models can even be applied to acquire hierarchical syntactic structures (Tsuruoka et al., 2009). However, long-distance dependencies widely exist in linguistic structures, and many NLP systems suffer from the incapability of capturing these dependencies. For example, previous work has shown that sequence models alone cannot deal with syntactic ambiguities well (Clark and Curran, 2004; Tsuruoka et al., 2009). On the contrary, state-of-the-art systems usually utilize high complexity models, such as lexicalized PCFG models for syntactic parsing, to achieve high accuracy. Unfortunately, they are not suitable for many real world applications due to the sacrifice of efficiency.

In this paper, we are concerned with capturing long-distance dependencies in sequence models. Our goal is to develop efficient models with linear time complexity that are also capable to capture non-local dependencies. Two techniques are studied to achieve this goal. First, stacked learning (Breiman, 1996) is employed to integrate sequence models that are good at exploring local information and other high complexity models that are good at capturing non-local dependencies. By combining complementary strengths of heterogeneous models, hybrid systems can obtain more accurate results. Second, structure compilation (Liang et al., 2008) is employed to transfer the predictive power of hybrid models to sequence models via large-scale unlabeled data. In particular, hybrid systems are utilized to create large-scale

International Joint Conference on Natural Language Processing, pages 180–188,
Nagoya, Japan, 14-18 October 2013.

pseudo training data for cheap sequence models. A discriminative model can be improved by incorporating more features, while a generative latent variable model can be improved by increasing the number of latent variables. By using stacking and structure compilation techniques, a sequence model can be enhanced to better capture long-distance dependencies and to achieve more accurate results.

To demonstrate the feasibility to capture long-distance dependencies in a sequence model, we present our work on Chinese POS tagging. The Chinese language has a number of characteristics that make Chinese POS tagging particularly challenging. While simple sequential classifiers can easily achieve tagging accuracies of above 97% on English, Chinese POS tagging has proven to be more challenging and has obtained accuracies of about 93-94% (Huang et al., 2009; Sun and Uszkoreit, 2012) when applying sequence models. Recent work shows that higher accuracy (c.a. 95%) can be achieved by applying advanced learning techniques to capture deep lexical relations (Sun and Uszkoreit, 2012). Especially, syntagmatic lexical relations have been shown playing an essential role in Chinese POS tagging. To capture such relations, an accurate POS tagging model should know more information about long range dependencies. Previous work has used syntactic parsers in either constituency or dependency formalisms to exploit such useful information (Sun and Uszkoreit, 2012; Hatori et al., 2011). However, it is inapproporiate to employ computationally expensive parsers to improve POS tagging for many realistic NLP applications, mainly due to efficiency considerations.

In this paper, we study several hybrid systems that are built upon various complementary tagging systems. We investigate stacked learning to build more accurate solutions by integrating heterogeneous models. Experiments on the Chinese Treebank (CTB) data show that stacking is very effective to build high-accuracy tagging systems. 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. 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. Experiments show that unlabeled data is effective to re-compile simple models, including latent variable hidden Markov models, local and global linear classifiers. On one hand, the precison in terms of word classification is improved to 95.33%, which reachs the state-of-the-art. 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 as well. Different from the purely supervised sequence models, re-compiled models also serve as a front-end to a parser well.

2 Background

The Chinese language has a number of characteristics that make Chinese POS tagging particularly challenging. For example, Chinese is characterized by the lack of formal devices such as morphological tense and number that often provide important clues for syntactic processing. Chinese POS tagging has proven to be very difficult and has obtained accuracies of about 93-94% (Huang et al., 2009; Li et al., 2011; Hatori et al., 2011; Sun and Uszkoreit, 2012). On the other hand, Chinese POS information is very important for advanced NLP tasks, e.g. supertagging, full parsing and semantic role labeling. Previous work has repeatedly demonstrated the significant performance gap of NLP systems while using gold standard and automatically predicted POS tags (Zhang and Clark, 2009; Li et al., 2011; Tse and Curran, 2012). In this section, we give a brief introduction and a comparative analysis to several models that are recently designed to resolve the Chinese POS tagging problem.

2.1 Various Chinese POS Tagging Models

Local linear model (LLM) 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 studies on many languages have shown that local classification is inadequate to capture structural information of output labels, and thus does not perform as well as structured models.

Linear-chain global linear model (LGLM) Sequence labeling models can capture output structures by exploiting local dependencies among words. A global linear model is flexible to include linguistic knowledge from multiple information sources, and thus suitable to recognize more new words. A majority of state-of-the-art English POS taggers are based on LGLMs, e.g. structured perceptron (Collins, 2002) and conditional random fields (Lafferty et al., 2001). Such models are also very popular for building Chinese POS taggers (Sun and Uszkoreit, 2012).

Hidden Markov model with latent variables (HMMLA) Generative models with latent annotations (LA) obtain state-of-the-art performance for a number of NLP tasks. For example, both PCFG and TSG with refined latent variables achieve excellent results for syntactic parsing (Matsuzaki et al., 2005; Shindo et al., 2012). For Chinese POS tagging, Huang, Eidelman and Harper (2009) described and evaluated a bi-gram 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.

PCFG Parsing with latent variables (PCFGLA) POS tags can be taken as preterminals 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 PCFG learning, with lexicalized (Collins, 2003; Charniak, 2000) or latent annotation (Matsuzaki et al., 2005; Petrov et al., 2006) refinements. Compared to complex lexicalized parsers, the PCFGLA 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 PCFGLA parser achieves the state-of-the-art performance and outperforms many other types of parsers (Zhang and Clark, 2009).

2.1.1 Joint POS Tagging and Dependency Parsing (DEP)

(Hatori et al., 2011) proposes an incremental processing model for the task of joint POS tagging and dependency parsing, which is built upon a shift-reduce parsing framework with dynamic programming. Given a segmented sentence, a joint model simultaneously considers possible POS tags and dependency relations. In this way, the learner can better predict POS tags by using bi-lexical dependency information. Their experiments show that the joint approach achieved substantial improvements over the pipeline systems in both POS tagging and dependency parsing tasks.

2.2 Comparison

We can distinguish the five representative tagging models from two views (see Table 2). From a linguistic view, we can distinguish syntax-free and syntax-based models. In a syntex-based model, POS tagging is integrated into parsing, and thus (to some extent) is capable of capturing long range syntactic information. From a machine learning view, we can distinguish generative and discriminative models. Compared to generative models, discriminative models define expressive features to classify words. Note that the two generative models employ latent variables to refine the output spaces, which significantly boost the accuracy and increase the robustness of simple generative models.

	Generative	Discriminative
Syntax-free	HMMLA	LLM, LGLM
Syntax-based	PCFGLA	DEP

Table 2: Two views of different tagging models.

2.3 Evaluation

2.3.1 Experimental 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, POS tagging, syntactic parsing in both constituency and dependency formalisms. In this paper, we use CTB 6.0 as the labeled training data for the study. In order to obtain a representative split of data sets, we conduct experiments following the setting of the CoNLL 2009 shared task (Hajič et al., 2009), which is also used by (Sun and Uszkoreit, 2012). The setting is provided by the principal organizer of the CTB project, and has considered many annotation details. This setting is very robust for evaluating Chinese language processing algorithms.

We present an empirical study of the five typical approaches introduced above. In our experiments, to build local and global word classifiers (i.e. LLMs and LGLMs), we implement the feature set used in (Sun and Uszkoreit, 2012). Denote a word w in focus with a fixed window $w_{-2}w_{-1}ww_{+1}w_{+2}$. The features include:

- Word unigrams: w_{-2}, w_{-1}, w, w_{+1}, w_{+2};

Devel.	LLM	LGLM(SP)	LGLM(PA)	HMMLA	PCFGLA	DEP
Overall	93.96%	94.30%/94.49%	94.24%/94.33%	94.16%	93.69%	94.58%
NR	95.07	94.47/94.85	94.41/94.56	94.22	89.84	93.55
NT	97.61	97.22/97.75	97.66/97.59	97.18	96.70	96.84
NN	94.89	94.67/94.79	94.72/94.71	94.30	93.56	94.55
DEC	78.61	81.98/82.36	80.68/81.76	80.60	85.78	86.73
DEG	82.44	85.58/86.72	85.37/85.00	85.19	88.94	89.45
UNK	- -	80.0%/81.1%	- -	78.2%	- -	- -

Table 1: Tagging accuracies of different supervised models on the development data.

- Word bigrams: $w_{-2}\_w_{-1}$, w_{-1}_w, w_w_{+1}, $w_{+1}\_w_{+2}$;
- Character n-gram prefixes and suffixes for n up to 3.

To train LLMs, we use the open source linear classifier – LIBLINEAR[1]. To train LGLMs, we choose structured perceptron (SP) (Collins, 2002) and passive aggressive (PA) (Crammer et al., 2006) learning algorithms. For the LAHMM and DEP models, we use the systems discribed in (Huang et al., 2009; Hatori et al., 2011); for the PCFGLA models, we use the Berkeley parser[2].

2.3.2 Results

Table 1 summarizes the performance in terms of per word classification of different supervised models on the development data. We present the results of both first order (on the left) and second order (on the right) LGLMs. We can see that the perceptron algorithm performs a little better than the PA algorithm for Chinese POS tagging. There is only a slight gap between the local classification model and various structured models. This is very different from English POS tagging. Although the local classifier achieves comparable results when respectively applied to English and Chinese, there is much more significant gap between the corresponding structured models. Similarly, the gap between the first and second order LGLMs is very modest too.

From the linguistic view, we mainly consider the disambiguiation ability of local and non-local dependencies. Table 1 presents accuracy results 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 which only explore local dependencies are good enough to deal with nouns. Surprisingly, 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, while the tag *DEG* denotes a genitive marker and an associative marker. These two types only include two words: "的" and "之." The latter one is mainly used in ancient Chinese. 5.19% of words appearing in the training data set is *DEC/DEG*. 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 of how well a tagging model resolves long distance dependencies. We can see that the two syntactic parsers significantly outperform local models on the prediction of these types of words.

The weak ability for non-local disambiguation also imposes restrictions on using a sequence POS tagging model as front module for parsing. To evaluate the impact, we employ the PCFGLA parser to parse a sentence based on the POS tags provided by sequence models. Table 4 shows the parsing performance. Note that the overall tagging performance of the Berkeley parser is significantly worse than sequence models. However, better POS tagging does not lead to better parsing. The experiments suggest that sequence models propagate too many errors to the parser. Our linguistic analysis can also well 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 supertag-

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

[2] code.google.com/p/berkeleyparser/

	LLM	First order LGLM		Second order LGLM	
		SP	PA	SP	PA
Baseline	93.96%	94.30%	94.24%	94.49%	94.33%
+Word clustering	94.75%	94.90%	94.80%	95.05%	94.96%
+Word clustering+HMMLA	**95.12%**	**95.19%**	**95.18%**	**95.14%**	**95.22%**
+Word clustering+PCFGLA	95.42%	95.50%	95.40%	95.56%	95.44%
+Word clustering+DEP	95.28%	95.22%	95.26%	95.29%	95.25%
+ALL	95.56%	95.61%	95.60%	95.53%	95.53%

Table 3: Tagging accuracies of different stacking models on the development data.

ging.

Devel.	LP	LR	F1
Berkeley	80.44	80.31	81.36
1or LGLM	80.38	79.48	79.93↓
2or LGLM	80.98	79.93	80.45↓
HMMLA	80.65	79.62	80.13↓
1or LGLM(HMMLA)	81.55	80.80	81.17↓
1or LGLM(PCFGLA)	82.84	81.75	82.29↑
1or LGLM(DEP)	82.69	81.68	82.18↑

Table 4: Parsing accuracies on the development data. *1or* and *2or* respectively denote first order and second order. *LGLM(X)* denotes a stacking model with *X* as the level-0 processing. All stacking models incorporate word clusters to improve the tagging accuracy.

To distinguish the predictive abilities of generative and discriminative models, we report the precison of the prediction of unknown words (UNK). Discriminative learning can define arbitrary (even overlapping) features which play a central role in tagging English unknown words. The difference between generative and discriminative learning in Chinese POS tagging is not that much, mainly because most Chinese words are compactly composed by a very few Chinese characters that are usually morphemes. This language-specific property makes it relatively easy to smooth parameters of a generative model.

3 Improving Tagging Accuracy via Stacking

In this section, we study a simple way of integrating multiple heterogeneous models in order to exploit their complementary strength 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 the HMMLA, PCFGLA and DEP to define features for the LLM/LGLM.

3.1 Stacked Learning

Stacked generalization is a meta-learning algorithm that has been 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).

3.2 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 the outputs of other models as new features. 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/bi-gram* features are added: w_{-1}, w, w_{+1}, w_{-1}_w, w_w_{+1}. This choice is tuned on the development data.

Word clusters that are automatically acquired from large-scale unlabeled data have been shown to be very effective to bridge the gap between high and low frequency words, and therefore significantly improve tagging, as well as other syntactic processing tasks. Our stacking models are all built on word clustering enhanced discriminative linear models. Five *word cluster uni/bi-gram* features are added: w_{-1}, w, w_{+1}, w_{-1}_w, w_w_{+1}. The clusters are acquired based on the Chinese giga-word data with the MKCLS tool. The number of total clusters is set to 500, which is tuned by (Sun and Uszkoreit, 2012).

3.3 Evaluation

Table 3 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 PCFGLA model are most effective in improving the LLM/LGLM baseline systems. Among different stacking models, the syntax-free hybrid one (i.e., stacking LLM/LGLM with HMMLA) does not need any treebank to train their systems. For the situations that 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 5 is the F1 scores of the DEC/DEG prediction which are obtained by different stacking models. Compared to Table 1, we can see that the hybrid sequence model is still not good at handling long-distance ambiguities. As a result, it harms the parsing performance (see Table 4), though it achieves higher overall precison.

Devel.	DEC	DEG
1or LGLM(HMMLA)	82.93	86.64
1or LGLM(PCFGLA)	88.11	91.12
1or LGLM(DEP)	87.46	89.86

Table 5: F1 score of the *DEC/DEG* prediction of different stacking models on the development data.

3.4 Related Work

(Sun and Uszkoreit, 2012) introduced a Bagging model to effectively combine the outputs of individual systems. In the training phase, given a training set D of size n, the Bagging model generates m new training sets D_i's by sampling examples from D. Each D_i is separately used to train k individual models. In the tagging phase, the km models outputs km tagging results, each word is assigned one POS label. The final tagging is the voting result of these km labels. Although this model is effective, it is too expensive in the sense that it uses parser multiple times. We also implement their method and compare the results with our stacking model. We find the accuracy performance produced by the two different methods are comparable.

(Rush et al., 2010) introduced dual decomposition as a framework for deriving inference algorithms for serious combinatorial problems in NLP. They successfully applied dual decomposition to the combination of a lexicalized parsing model and a trigram POS tagger. Despite the effectiveness, their method iteratively parses a sentence many times to achieve convergence, and thus is not as efficient as stacking.

4 Improving Tagging Efficiency through Unlabeled Data

4.1 The Idea

Hybrid structured models often achieve excellent performance but can be slow at test time. In our problem, it is obviously too inefficient to improve POS tagging by parsing a sentence first. In this section, 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 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 et al., 2006), or from structured prediction models to local classification models (Liang et al., 2008),

4.2 Reducing Hybrid Models to Sequence Models

For English POS tagging, Liang, Daumé and Klein (2008) have done some experiments to

Size of data	HMMLA	LLM	LGLM	LLM	LGLM	Voting	
		win size=3		win size=4			DEC/DEG
+100k	94.72%	95.05%	95.07%	95.04%	95.10%	95.36%	- -
+200k	94.77%	95.06%	95.18%	95.20%	95.23%	95.43%	- -
+500k	94.97%	95.11%	95.21%	95.15%	95.23%	95.43%	- -
+1000k	95.09%	95.19%	95.23%	95.22%	95.31%	95.49%	85.75/89.01

Table 6: Tagging accuracies of different re-compiled models on the development data.

transfer the power of a chain conditional random field to a logistic regression model. Similarly, we do some experiments to explore the feasibility of reducing hybrid tagging models to a HMMLA, LLM or LGLM, for Chinese POS tagging. The large-scale unlabeled data we use in our experiments comes from the Chinese Gigaword (LDC2005T14), which is a comprehensive archive of newswire text data that has been acquired over several years by the Linguistic Data Consortium (LDC). We choose the Mandarin news text, i.e. Xinhua newswire. We tag giga-word sentences by applying the stacked first order LGLMs with all other models. In other words, the HMMLA, PCFGLA and DEP systems are applied to tag unlabeled data features and their outputs are utilized to define features for first-order and second-order LGLMs which produce pseudo training data. Both original gold standard training data and pseudo training data are used to re-train a HMMLA, a LLM/LGLM with extended features.

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 lots of labeled training examples, unlabeled instances are still helpful for hybrid models since they can help to increase the diversity among the base learners. The author also briefly introduced a preliminary theoretical study. In this paper, we also combine the re-trained models to see if we can benefit more. We utilize voting as the strategy for final combination. In the tagging phase, the re-trained LLM, LGLM and HMMLA systems outputs 3 tagging results, each word is assigned one POS label. The final tagging is the voting result of these 3 labels.

4.3 Experiments

4.3.1 Reducing Hybrid Models to HMMLA

With the increase of (pseudo) training data, a HMMLA may learn better latent variables to subcategorize POS tags, which could significantly improve a purely supervised HMMLA. In our experiments, all HMMLA models are trained with 8 iterations of split, merge, smooth. The second column of Table 6 shows the performance of the re-trained HMMLAs. The first column is the number of sentences of pseudo sentences. The pseudo sentences are selected from the begining of the Chinese gigaword. We can clearly see that the idea to leverage unlabeled data to transfer the predictive ability of the hybrid model works. Self-training can also slightly improve a HMMLA (Huang et al., 2009). Our auxiliary experiments show that self-training is not as effective as our methods.

4.3.2 Reducing Hybrid Models to LLM/LGLM

To increase the expressive power of a discriminative classification model, we extend the feature templates. This strategy is proposed by (Liang et al., 2008). In our experiments, we increase the window size of word uni/bi-gram features to approximate long 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; Column 3 to 6 of Table 6 show the performance of the re-compiled LLMs/LGLMs. Similar to the generative model, the discriminative LLM/LGLM can be improved too.

4.3.3 Voting

The last two columns of Table 6 are the final voting results of the HMMLA, LLM and LGLM. The window size of word uni/bi-gram features for the LLM and LGLM is set to 4. 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 very close to the best stacking models. Furthermore, the F1 scores of the DEC/DEG prediction are 85.75 and 89.01, which are very close to parsers too.

4.3.4 Improving Parsing

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. Our evaluation shows that the significant improvement of the POS tagging stop harming syntactic parsing. Results in Table 7 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 whole syntactic processing efficiency.

Devel.	LP	LR	F1
HMMLA	82.18	81.16	81.66↑
LLM	81.86	80.93	81.40↑
LGLM	82.07	81.21	81.64↑
Voting	82.34	81.42	81.88↑

Table 7: Accuracies of parsing based on re-compiled tagging.

4.3.5 Final results

Table 8 shows the performance of different systems evaluated on the test data. Our final sequence model achieve the state-of-the-art performance, which is once obtained by combining multiple parsers as well as sequence models.

Systems	Acc.
(Sun and Uszkoreit, 2012)	95.34%
Our system	95.33%

Table 8: Tagging accuracies on the test data.

5 Conclusion

In this paper, we study two techniques to build accurate and fast sequence models for Chinese POS tagging. In particular, our goal is to capture long-distance dependencies in sequence models. To improve tagging accuracy, we study stacking to integrate multiple models with heterogeneous views. 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 POS taggers. Another advantage of our system is that it serves as a front-end to a parser very well. Our study suggests that complicated structured models can be well simulated by simple sequence models through unlabeled data.

Acknowledgement

The work was supported by NSFC (61170166), Beijing Nova Program (2008B03) and National High-Tech R&D Program (2012AA011101).

References

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

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*, pages 535–541.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL*.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage ccg parsing. In *Proceedings of Coling 2004*, pages 282–288, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8. Association for Computational Linguistics, July.

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

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *JOURNAL OF MACHINE LEARNING RESEARCH*, 7:551–585.

Mark Craven. 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.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*, Boulder, Colorado, USA.

Jun Hatori, 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, Thailand, November. Asian Federation of Natural Language Processing.

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.

John D. Lafferty, 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*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Zhenghua Li, 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, Scotland, UK., July. Association for Computational Linguistics.

Percy Liang, Hal Daumé, III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 592–599, New York, NY, USA. ACM.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of ACL*, ACL '05, pages 75–82, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Slav Petrov, 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, Australia, July. Association for Computational Linguistics.

Alexander M Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of EMNLP*, pages 1–11, Cambridge, MA, October. Association for Computational Linguistics.

Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. 2012. Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of ACL*, pages 440–448, Jeju Island, Korea, July. Association for Computational Linguistics.

Weiwei Sun 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*. Association for Computational Linguistics, July.

Weiwei Sun. 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, Oregon, USA, June. Association for Computational Linguistics.

Daniel Tse 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, Canada, June. Association for Computational Linguistics.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Fast full parsing by linear-chain conditional random fields. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 790–798, Athens, Greece, March. Association for Computational Linguistics.

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

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.

Yue Zhang 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, France, October. Association for Computational Linguistics.

Zhi-Hua Zhou. 2009. When semi-supervised learning meets ensemble learning. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, MCS '09, pages 529–538, Berlin, Heidelberg. Springer-Verlag.