Coupled Sequence Labeling on Heterogeneous Annotations: POS Tagging as a Case Study

Zhenghua Li, Jiayuan Chao, Min Zhang*, Wenliang Chen

(1) Soochow University

(2) Collaborative Innovation Center of Novel Software Technology and Industrialization

Jiangsu Province, China

{zhli13,minzhang,wlchen}@suda.edu.cn; china_cjy@163.com

Abstract

In order to effectively utilize multiple datasets with heterogeneous annotations, this paper proposes a coupled sequence labeling model that can directly learn and infer two heterogeneous annotations simultaneously, and to facilitate discussion we use Chinese part-of-speech (POS) tagging as our case study. The key idea is to bundle two sets of POS tags together (e.g. "$[NN, n]$"), and build a conditional random field (CRF) based tagging model in the enlarged space of bundled tags with the help of *ambiguous labelings*. To train our model on two non-overlapping datasets that each has only one-side tags, we transform a one-side tag into a set of bundled tags by considering all possible mappings at the missing side and derive an objective function based on ambiguous labelings. The key advantage of our coupled model is to provide us with the flexibility of 1) incorporating joint features on the bundled tags to implicitly learn the loose mapping between heterogeneous annotations, and 2) exploring separate features on one-side tags to overcome the data sparseness problem of using only bundled tags. Experiments on benchmark datasets show that our coupled model significantly outperforms the state-of-the-art baselines on both one-side POS tagging and annotation conversion tasks. The codes and newly annotated data are released for non-commercial usage.[1]

1 Introduction

The scale of available labeled data significantly affects the performance of statistical data-driven models. As a widely-used structural classification problem, sequence labeling is prone to suffer from the data sparseness issue. However, the heavy cost of manual annotation typically limits one labeled resource in both scale and genre. As a promising research line, semi-supervised learning for sequence labeling has been extensively studied. Huang et al. (2009) show that standard self-training can boost the performance of a simple hidden Markov model (HMM) based part-of-speech (POS) tagger. Søgaard (2011) apply tri-training to English POS tagging, boosting accuracy from 97.27% to 97.50%. Sun and Uszkoreit (2012) derive word clusters from large-scale unlabeled data as extra features for Chinese POS tagging. Recently, the use of natural annotation has becomes a hot topic in Chinese word segmentation (Jiang et al., 2013; Liu et al., 2014; Yang and Vozila, 2014). The idea is to derive segmentation boundaries from implicit information encoded in web texts, such as anchor texts and punctuation marks, and use them as partially labeled training data in sequence labeling models.

The existence of multiple annotated resources opens another door for alleviating data sparseness. For example, Penn Chinese Treebank (*CTB*) contains about 20 thousand sentences annotated with word boundaries, POS tags, and syntactic structures (Xue et al., 2005), which is widely used for research on Chinese word segmentation and POS tagging. People's Daily corpus (*PD*)[2] is a large-scale corpus annotated with word segments and POS tags, containing about 300 thousand sentences from the first half of 1998 of People's

*Correspondence author.

[1] http://hlt.suda.edu.cn/~zhli

[2] http://icl.pku.edu.cn/icl_groups/corpustagging.asp

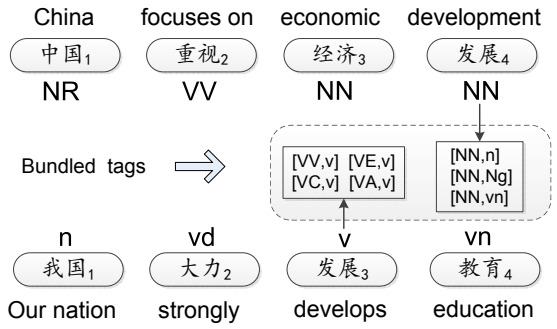

Figure 1: An example to illustrate the annotation differences between *CTB* (above) and *PD* (below), and how to transform a one-side tag into a set of bundled tags. "*NN*" and "*n*" represent nouns; "*VV*"and "*v*" represent verbs.

Daily newspaper (see Table 2). The two resources were independently built for different purposes. *CTB* was designed to serve syntactic analysis, whereas *PD* was developed to support information extraction systems. However, the key challenge of exploiting the two resources is that they adopt different sets of POS tags which are impossible to be precisely converted from one to another based on heuristic rules. Figure 1 shows two example sentences from *CTB* and *PD*. Please refer to Table B.3 in Xia (2000) for detailed comparison of the two guidelines.

Previous work on exploiting heterogeneous data (*CTB* and *PD*) mainly focuses on indirect guide-feature based methods. The basic idea is to use one resource to generate extra guide features on another resource (Jiang et al., 2009; Sun and Wan, 2012), which is similar to stacked learning (Nivre and McDonald, 2008). First, *PD* is used as source data to train a source model $\textit{Tagger}_{PD}$. Then, $\textit{Tagger}_{PD}$ generates automatic POS tags on the target data *CTB*, called *source annotations*. Finally, a target model $\textit{Tagger}_{\textit{CTB-guided}}$ is trained on *CTB*, using source annotations as extra guide features. Although the guide-feature based method is effective in boosting performance of the target model, we argue that it may have two potential drawbacks. First, the target model $\textit{Tagger}_{\textit{CTB-guided}}$ does not directly use *PD* as training data, and therefore fails to make full use of rich language phenomena in *PD*. Second, the method is more complicated in real applications since it needs to parse a test sentence twice to get the final results.

This paper proposes a coupled sequence labeling model that can directly learn and infer two heterogeneous annotations simultaneously. We use Chinese part-of-speech (POS) tagging as our case study.[3] The key idea is to bundle two sets of POS tags together (e.g. "[*NN*, *n*]"), and build a conditional random field (CRF) based tagging model in the enlarged space of bundled tags. To make use of two non-overlapping datasets that each has only one-side tags, we transform a one-side tag into a set of bundled tags by considering all possible mappings at the missing side and derive an objective function based on *ambiguous labelings*. During training, the CRF-based coupled model is supervised by such ambiguous labelings. The advantages of our coupled model are to provide us the flexibility of 1) incorporating joint features on the bundled tags to implicitly learn the loose mapping between two sets of annotations, and 2) exploring separate features on one-side tags to overcome the data sparseness problem of using bundled tags. In summary, this work makes two major contributions:

1. We propose a coupled model which can more effectively make use of multiple resources with heterogeneous annotations, compared with both the baseline and guide-feature based method. Experiments show our approach can significantly improve POS tagging accuracy from 94.10% to 95.00% on *CTB*.

2. We have manually annotated *CTB* tags for $1,000$ *PD* sentences, which is the first dataset with two-side annotations and can be used for annotation-conversion evaluation. Experiments on the newly annotated data show that our coupled model also works effectively on the annotation conversion task, improving conversion accuracy from 90.59% to 93.90% ($+3.31\%$).

2 Traditional POS Tagging ($\textit{Tagger}_{\textit{CTB}}$)

Given an input sentence of n words, denoted by $\mathbf{x} = w_1...w_n$, POS tagging aims to find an optimal tag sequence $\mathbf{t} = t_1...t_n$, where $t_i \in \mathcal{T}$ ($1 \leq i \leq n$) and $\mathcal{T}$ is a predefined tag set. As a log-linear probabilistic model (Lafferty et al., 2001), CRF

[3] There are some slight differences in the word segmentation guidelines between *CTB* and *PD*, which are ignored in this work for simplicity.

01: $t_i \circ t_{i-1}$	02: $t_i \circ w_i$
03: $t_i \circ w_{i-1}$	04: $t_i \circ w_{i+1}$
05: $t_i \circ w_i \circ c_{i-1,-1}$	06: $t_i \circ w_i \circ c_{i+1,0}$
07: $t_i \circ c_{i,0}$	08: $t_i \circ c_{i,-1}$
09: $t_i \circ c_{i,k}, 0 < k < \#c_i - 1$	
10: $t_i \circ c_{i,0} \circ c_{i,k}, 0 < k < \#c_i - 1$	
11: $t_i \circ c_{i,-1} \circ c_{i,k}, 0 < k < \#c_i - 1$	
12: **if** $\#c_i = 1$ **then** $t_i \circ w_i \circ c_{i-1,-1} \circ c_{i+1,0}$	
13: **if** $c_{i,k} = c_{i,k+1}$ **then** $t_i \circ c_{i,k} \circ$ *"consecutive"*	
14: $t_i \circ \mathsf{prefix}(w_i, k), 1 \leq k \leq 4, k \leq \#c_i$	
15: $t_i \circ \mathsf{suffix}(w_i, k), 1 \leq k \leq 4, k \leq \#c_i$	

Table 1: POS tagging features $\mathbf{f}(\mathbf{x}, i, t_{i-1}, t_i)$. $\circ$ means string concatenation; $c_{i,k}$ denotes the k^{th} Chinese character of w_i; $c_{i,0}$ is the first Chinese character; $c_{i,-1}$ is the last Chinese character; $\#c_i$ is the total number of Chinese characters contained in w_i; $\mathsf{prefix}/\mathsf{suffix}(w_i, k)$ denote the k-Character prefix/suffix of w_i.

defines the probability of a tag sequence as:

$$P(\mathbf{t}|\mathbf{x};\theta) = \frac{\exp(Score(\mathbf{x},\mathbf{t};\theta))}{\sum_{\mathbf{t}'}\exp(Score(\mathbf{x},\mathbf{t}';\theta))}$$
$$Score(\mathbf{x},\mathbf{t};\theta) = \sum_{1\leq i\leq n} \theta \cdot \mathbf{f}(\mathbf{x}, i, t_{i-1}, t_i) \quad (1)$$

where $\mathbf{f}(\mathbf{x}, i, t_{i-1}, t_i)$ is the feature vector at the i^{th} word and θ is the weight vector. We adopt the state-of-the-art tagging features in Table 1 (Zhang and Clark, 2008).

3 Coupled POS Tagging (*Tagger*$_{CTB\&PD}$)

In this section, we introduce our coupled model, which is able to learn and predict two heterogeneous annotations simultaneously. The idea is to bundle two sets of POS tags together and let the CRF-based model work in the enlarged tag space. For example, a *CTB* tag "*NN*" and a *PD* tag "*n*" would be bundled into "[*NN,n*]". Figure 2 shows the graphical structure of our model.

Different from the traditional model in Eq. (1), our coupled model defines the score of a bundled tag sequence as follows:

$$Score(\mathbf{x}, [\mathbf{t}^a, \mathbf{t}^b]; \theta) = \sum_{1\leq i\leq n} \theta \cdot \begin{bmatrix} \mathbf{f}(\mathbf{x}, i, [t^a_{i-1}, t^b_{i-1}], [t^a_i, t^b_i]) \\ \mathbf{f}(\mathbf{x}, i, t^a_{i-1}, t^a_i) \\ \mathbf{f}(\mathbf{x}, i, t^b_{i-1}, t^b_i) \end{bmatrix} \quad (2)$$

where the first item of the enlarged feature vector is called *joint features*, which can be obtained by instantiating Table 1 by replacing t_i with bundled tags $[t^a_i, t^b_i]$; the second and third items are called *separate features*, which are based on single-side tags. The advantages of our coupled model over the traditional model are to provide us with the flexibility of using both kinds of features, which significantly contributes to the accuracy improvement as shown in the following experiments.

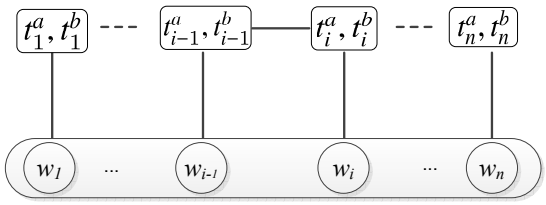

Figure 2: Graphical structure of our coupled CRF model.

3.1 Mapping Functions

The key challenge of our idea is that both *CTB* and *PD* are non-overlapping and each contains only one-side POS tags. Therefore, the problem is how to construct training data for our coupled model. We denote the tag set of *CTB* as $\mathcal{T}^a$, and that of *PD* as $\mathcal{T}^b$, and the bundled tag set as $\mathcal{T}^{a\&b}$. Since the full Cartetian $\mathcal{T}^a \times \mathcal{T}^b$ would lead to a very large number of bundled tags, making the model very slow, we would like to come up with a much smaller $\mathcal{T}^{a\&b} \subseteq \mathcal{T}^a \times \mathcal{T}^b$, based on linguistic insights of the annotation guidelines of the two datasets.

To obtain a proper $\mathcal{T}^{a\&b}$, we introduce a mapping function between the two sets of tags as $\mathbf{m} : \mathcal{T}^a \times \mathcal{T}^b \rightarrow \{0, 1\}$, which only allow specific tag pairs to be bundled together.

$$\mathbf{m}(t^a, t^b) = \begin{cases} 1 & \textit{if the two tags can be bundled} \\ 0 & \textit{otherwise} \end{cases} \quad (3)$$

where one mapping function $\mathbf{m}$ corresponds to one $\mathcal{T}^{a\&b}$. When the mapping function becomes looser, the tag set size $|\mathcal{T}^{a\&b}|$ becomes larger.

Then, based on the mapping function, we can map a single-side POS tag into a set of bundled tags by considering all possible tags at the missing side, as illustrated in Figure 1. The word "发展$_4$" is tagged as "*NN*" at the *CTB* side. Suppose that the mapping function $\mathbf{m}$ tells that "*NN*" can be mapped into three tags at the *PD* side, i.e., "*n*", "*Ng*", and "*vn*". Then, we create three bundled tags for the word, i.e., "[*NN, n*]", "[*NN, Ng*]",

"[*NN, vn*]" as its gold-standard references during training. It is known as *ambiguous labelings* when a training instance has multiple gold-standard labels. Similarly, we can obtain bundled tags for all other words in sentences of *CTB* and *PD*. After such transformation, the two datasets are now in the same tag space.

At the beginning of this work, our intuition is that the coupled model would achieve the best performance if we build a tight and linguistically motivated mapping function. However, our preliminary experiments show that our intuitive assumption is actually incorrect. Therefore, we experiment with the following four mapping functions to manage to figure out the reasons behind and to better understand our coupled model.

- The **tight** mapping function produces 145 tags, and is constructed by strictly following linguistic principles and our careful study of the two guidelines and datasets.
- The **relaxed** mapping function results in 179 tags, which is an looser version of the tight mapping function by including extra 34 weak mapping relationships.
- The **automatic** mapping function generates 346 tags. We use the baseline *Tagger*$_{CTB}$ to parse *PD*, and collect all automatic mapping relationships.
- The **complete** mapping function obtains $1,254$ tags ($|\mathcal{T}^a| \times |\mathcal{T}^b| = 33 \times 38$).

3.2 Training Objective with Ambiguous Labelings

So far, we have formally defined a coupled model and prepared both *CTB* and *PD* in the same bundled tag space. The next problem is how to learn the model parameters θ. Note that after our transformation, a sentence in *CTB* or *PD* have many tag sequences as gold-standard references due to the loose mapping function, known as *ambiguous labelings*. Here, we derive a training objective based on ambiguous labelings. For simplicity, we illustrate the idea based on the notations of the baseline CRF model in Eq. (1).

Given a sentence $\mathbf{x}$, we denote a set of ambiguous tag sequences as $\mathcal{V}$. Then, the probability of $\mathcal{V}$ is the sum of probabilities of all tag sequences contained in $\mathcal{V}$:

$$p(\mathcal{V}|\mathbf{x};\theta) = \sum_{\mathbf{t}\in\mathcal{V}} p(\mathbf{t}|\mathbf{x};\theta) \tag{4}$$

Algorithm 1 SGD training with two labeled datasets.

1: **Input:** Two labeled datasets: $\mathcal{D}^{(1)} = \{(\mathbf{x}_i^{(1)}, \mathcal{V}_i^{(1)}\}_{i=1}^{N}$, $\mathcal{D}^{(2)} = \{(\mathbf{x}_i^{(2)}, \mathcal{V}_i^{(2)})\}_{i=1}^{M}$; Parameters: I, N', M', b
2: **Output:** θ
3: **Initialization:** $\theta_0 = \mathbf{0}$, $k = 0$;
4: **for** $i = 1$ **to** I **do** {*iterations*}
5: Randomly select N' instances from $\mathcal{D}^{(1)}$ and M' instances from $\mathcal{D}^{(2)}$ to compose a new dataset $\mathcal{D}_i$, and shuffle it.
6: Traverse $\mathcal{D}_i$, and use a small batch $\mathcal{D}_k^b \subseteq \mathcal{D}_i$ at one step.
7: $\theta_{k+1} = \theta_k + \eta_k \frac{1}{b} \nabla \mathcal{L}(\mathcal{D}_k^b; \theta_k)$
8: $k = k + 1$
9: **end for**

Suppose the training data is $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{V}_i)\}_{i=1}^{N}$. Then the log likelihood is:

$$\mathcal{L}(\mathcal{D};\theta) = \sum_{i=1}^{N} \log p(\mathcal{V}_i|\mathbf{x}_i;\theta) \tag{5}$$

After derivation, the gradient is:

$$\frac{\partial \mathcal{L}(\mathcal{D};\theta)}{\partial \theta} = \sum_{i=1}^{N} (E_{\mathbf{t}\in\mathcal{V}_i}[\mathbf{f}(\mathbf{x}_i,\mathbf{t})] - E_{\mathbf{t}}[\mathbf{f}(\mathbf{x}_i,\mathbf{t})]) \tag{6}$$

where $\mathbf{f}(\mathbf{x}_i, \mathbf{t})$ is an aggregated feature vector for tagging $\mathbf{x}_i$ as $\mathbf{t}$; $E_{\mathbf{t}\in\mathcal{V}_i}[.]$ means model expectation of the features in the constrained space of $\mathcal{V}_i$; $E_{\mathbf{t}}[.]$ is model expectation with no constraint. This function can be efficiently solved by the forward-backward algorithm. Please note that the training objective of a traditional CRF model can be understood as a special case where $\mathcal{V}_i$ contains one sequence.

3.3 SGD Training with Two Datasets

We adopt stochastic gradient descent (SGD) to iteratively learn θ for our baseline and coupled models. However, we have two separate training data, and *CTB* may be overwhelmed by *PD* if directly merging the two datasets into one, since *PD* is 15 times larger than *CTB* (see Table 2), Therefore, we propose a simple corpus-weighting strategy, as shown in Algorithm 1, where $\mathcal{D}_k^b$ is a subset of training data used in k^{th} step update; b is the batch size; η_k is a update step. The idea is to randomly sample instances from each training data in a certain proportion before each iteration.

The sampled data is then used for one-iteration training. Later experiments will investigate the effect of the weighting proportion. In this work, we use $b = 30$, and follow the implementation in CRFsuite[4] to decide η_k.

4 Manually Annotating *PD* Sentences with *CTB* Tags

To evaluate different methods on annotation conversion, we build the first dataset that contains $1,000$ sentences with POS tags on both sides of *CTB* and *PD*. The sentences are randomly sampled from *PD*. To save annotation effort, we only select 20% most difficult tokens to manually annotate. The difficulty of a word w_i is measured based on marginal probabilities produced by the baseline *Tagger*$_{CTB}$. $p(t_i|\mathbf{x}, w_i; \theta)$ denotes the marginal probability of tagging w_i as t_i. The basic assumption is that w_i is more difficult to annotate if its most likely tag candidate ($\arg\max_t p(t|\mathbf{x}, w_i; \theta)$) gets lower marginal probability.

We build a visualized online annotation system to facilitate manual labeling. The annotation task is designed in such way that at a time an annotator is provided with a sentence and one focus word, and is required to decide the *CTB* POS tag of the word. To further simplify annotation, we provide two or three most likely tag candidates as well, so that annotators can choose one either among the candidates or from a full list. We employ 8 undergraduate students as our annotators. Annotators are trained on simulated tasks from *CTB* data for several hours, and and start real annotation once reaching certain accuracy. To guarantee annotation quality, we adopt *multiple annotation*. Initially, one task is randomly assigned to two annotators. Later, if the two annotators submit different results, the system will assign the task to two more annotators. To aggregate annotation results, we only retain annotation tasks that the first two annotators agree (91.0%) or three annotators among four agree (5.6%), and discard other tasks (3.4%). Finally, we obtain $5,769$ words with both *CTB* and *PD* tags, with each annotator's detailed submissions, and could be used as a non-synthesized dataset for studying aggregating submissions from non-expert annotators in crowdsourcing platforms (Qing et al., 2014). The data is also fully released for non-commercial usage.

[4] http://www.chokkan.org/software/crfsuite/

5 Experiments

In this section, we conduct experiments to verify the effectiveness of our approach. We adopt *CTB* (version 5.1) with the standard data split, and randomly split *PD* into four sets, among which one set is 20% *partially* annotated with *CTB* tags. The data statistics is shown in Table 2. The main concern of this work is to improve accuracy on *CTB* by exploring large-scale *PD*, since *CTB* is relatively small, but is widely-used benchmark data in the research community.

We use the standard token-wise tagging accuracy as the evaluation metric. For significance test, we adopt Dan Bikel's randomized parsing evaluation comparator (Noreen, 1989).[5].

The baseline CRF is trained on either *CTB* training data with 33 tags, or *PD* training data with 38 tags. The coupled CRF is trained on both two separate training datasets with bundled tags (179 tags for the relaxed mapping function). During evaluation, the coupled CRF is not directly evaluated on bundled tags, since bundled tags are unavailable in either *CTB* or *PD* test data. Instead, the coupled and baseline CRFs are both evaluated on one-side tags.

5.1 Model Development

Our coupled model has two major parameters to be decided. The first parameter is to determine the mapping function between *CTB* and *PD* annotations, and the second parameter is the relative weights of the two datasets during training (N' vs. M': number of sentences in each dataset used for training at one iteration).

Effect of mapping functions (described in Subsection 3.1) is illustrated in Figure 3. Empirically, we adopt $N' = 5K$ vs. $M' = 20K$ to merge the two training datasets at each iteration. Our intuition is that using this proportion, *CTB* should not be overwhelmed by *PD*, and both training data can be used up in relatively similar speed. Specifically, all training data of *CTB* can be consumed in about 3 iterations, whereas *PD* can be consumed in about 14 iterations. We also present the results of the baseline model trained using $5K$ sentences in one iteration for better comparison.

Contrary to our intuitive assumption, it actually leads to very bad performance when using the

[5] http://www.cis.upenn.edu/~dbikel/software.html

		#sentences	#tokens with *CTB* tags	#tokens with *PD* tags
CTB	train	16,091	437,991	–
	dev	803	20,454	–
	test	1,910	50,319	–
PD	train	273,883	–	6,488,208
	dev	1,000	–	23,427
	test	2,500	–	58,301
	newly labeled	1,000	5,769	27,942

Table 2: Data statistics. Please kindly note that the $1,000$ sentences originally from *PD* are only partially annotated with *CTB* tags (about 20% most ambiguous tokens).

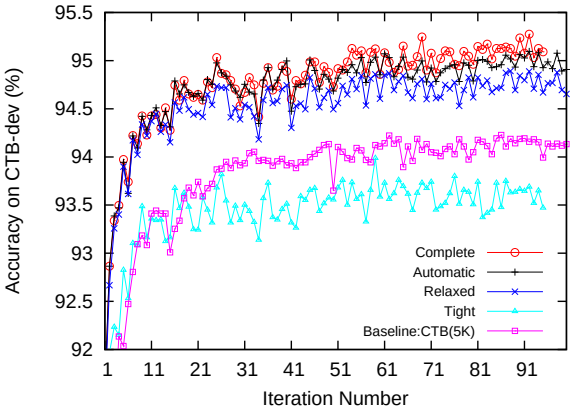

Figure 3: Accuracy on *CTB*-dev regarding to mapping functions. Please kindly note that some experiments run very slowly, and will be updated when available.

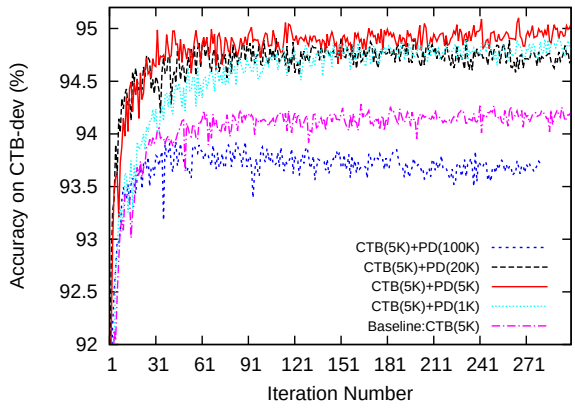

Figure 4: Accuracy on *CTB*-dev with different weighting settings.

tight mapping function that is carefully created based on linguistic insights, which is even inferior to the baseline model. The relaxed mapping function outperforms the tight function by large margin. The automatic function works slightly better than the relaxed one. The complete function achieves similar accuracy with the automatic one. In summary, we can conclude that our coupled model achieves much better performance when the mapping function becomes looser. In other words, this suggests that *our coupled model can effectively learn the implicit mapping between heterogeneous annotations, and does not rely on a carefully designed mapping function.*

Since a looser mapping function leads to a larger number of bundled tags and makes the model slower, we implement a paralleled training procedure based on Algorithm 1, and run each experiment with five threads. However, it still takes about 20 hours for one iteration when using the complete mapping function; whereas the other three mapping functions need about 6, 2, and 1 hours respectively. Therefore, as a compromise, *we adopt the relaxed mapping function in the following experiments*, which achieves slightly lower accuracy than the complete mapping function, but is much faster.

Effect of weighting *CTB* and *PD* is investigated in Figure 4 and 5. Since the scale of *PD* is much larger than *CTB*, we adopt Algorithm 1 to merge the training data in a certain proportion (N' *CTB* sentences and M' *PD* sentences) at each iteration. We use $N' = 5K$, and vary $M' = 1K/5K/20K/100K$. Figure 4 shows the accuracy curves on *CTB* development data. We find that when $M' = 100K$, our coupled model achieve very low accuracy, which is even worse than the baseline model. The reason should be that the training instances in *CTB* are overwhelmed by those in *PD* when M' is large. In contrast, when $M' = 1K$, the accuracy is also inferior to the case of $M' = 5K$, which indicates that *PD* is not effectively utilized in this setting. Our model works best when $M' = 5K$, which is slightly better than the case of $M' = 1K/20K$.

Figure 5 shows the accuracy curves on *PD*

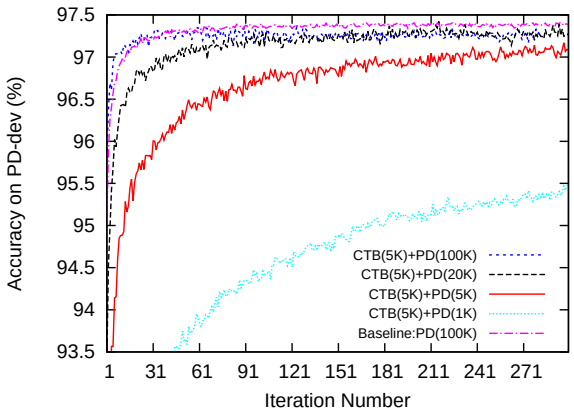

Figure 5: Accuracy on *PD*-dev with different weighting settings.

development data. The baseline model is trained using $100K$ sentences in one iteration. We find that when $M' = 100K$, our coupled model achieves similar accuracy with the baseline model. When M' becomes smaller, our coupled model becomes inferior to the baseline model. Particularly, when $M' = 1K$, the model converges very slowly. However, from the trend of the curves, we expect that the accuracy gap between our coupled model with $M' = 5K/20K$ and the baseline model should be much smaller when reaching convergence. Based on the above observation, *we adopt $N' = 5K$ and $M' = 5K$ in the following experiments.* Moreover, we select the best iteration on the development data, and use the corresponding model to parse the test data.

5.2 Final Results

Table 3 shows the final results on the *CTB* test data. We re-implement the guide-feature based method of Jiang et al. (2009), referred to as two-stage CRF. Li et al. (2012) jointly models Chinese POS tagging and dependency parsing, and report the best tagging accuracy on *CTB*. The results show that *our coupled model outperforms the baseline model by large margin, and also achieves slightly higher accuracy than the guide-feature based method*.

5.3 Feature Study

We conduct more experiments to measure individual contribution of each feature set, namely the joint features based on bundled tags and separate features based on single-side tags, as defined in Eq. (2). Table 4 shows the results. We can see that when only using separate features, our coupled model achieves only slightly better accuracy than the baseline model. This is because there is little connection and help between the two sets annotations. When only using joint features, our coupled model becomes largely inferior to the baseline, which is due to the data sparseness problem for the joint features. However, when the two sets of features are combined, the coupled model largely outperforms the baseline model. These results indicate that *both joint features and separate features are indispensable components and complementary to each other for the success of our coupled model.*

	Accuracy
Baseline CRF	94.10
Two-stage CRF (guide-feature)	94.81 (+0.71) †
Coupled CRF	**95.00** (+0.90) †‡
Best result (Li et al., 2012)	94.60

Table 3: Final results on *CTB* test data. † means the corresponding approach significantly outperforms the baseline at confidence level of $p < 10^{-5}$; whereas ‡ means the accuracy difference between the two-stage CRF and the coupled CRF is significant at confidence level of $p < 10^{-2}$.

	dev	test
Baseline CRF	94.28	94.10
Coupled CRF (w/ separate feat)	94.36	94.43 (+0.33)
Coupled CRF (w/ joint feat)	92.92	92.90 (-1.20)
Coupled CRF (full)	**95.10**	**95.00** (+0.90)

Table 4: Accuracy on *CTB*: feature study.

	PD-to-*CTB* conversion
Baseline CRF	90.59
Two-stage CRF (guide-feature)	93.22 (+2.63) †
Coupled CRF	**93.90** (+3.31) †‡

Table 5: Conversion accuracy on our annotated data. † means the corresponding approach significantly outperforms the baseline at confidence level of $p < 10^{-5}$; whereas ‡ means the accuracy difference between the two-stage CRF and the coupled CRF is significant at confidence level of $p < 10^{-2}$.

	dev	test
Baseline CRF	94.28	94.10
Coupled CRF	**95.10**	**95.00** (+0.90) †
Baseline CRF + converted *PD*	95.01	94.81 (+0.71) †‡

Table 6: Accuracy on *CTB*: using converted *PD*. † means the corresponding approach significantly outperforms the baseline at confidence level of $p < 10^{-5}$; whereas ‡ means the accuracy difference between the coupled CRF and the baseline CRF with converted *PD* is significant at confidence level of $p < 10^{-2}$.

5.4 Results on Annotation Conversion

In this subsection, we evaluate different methods on the annotation conversion task using our newly annotated $1,000$ sentences. The gold-standard *PD*-side tags are provided, and the goal is to obtain the *CTB*-side tags via annotation conversion. We evaluate accuracy on the $5,769$ words having manually annotated *CTB*-side tags.

Our coupled model can be naturally used for annotation conversion. The idea is to perform constrained decoding on the test data, using the *PD*-side tags as hard constraints. The guide-feature based method can also perform annotation conversion by using the gold-standard *PD*-side tags to compose guide features. Table 5 shows the results. The accuracy is much lower than those in Table 3, because the $5,769$ words used for evaluation are 20% most ambiguous tokens in the $1,000$ test sentence (partial annotation to save annotation effort). From Table 5, we can see that *our coupled model outperforms both the baseline and guide-feature based methods by large margin*.

5.5 Results of Training with Converted Data

One weakness of our coupled model is the inefficiency problem due to the large bundled tag set. In practice, we usually only need results following one annotation style. Therefore, we employ our coupled model to convert *PD* into the style of *CTB*, and train our baseline model with two training data with homogeneous annotations. Again, Algorithm 1 is used to merge the two data with $N' = 5K$ and $M' = 5K$. The results are shown in the bottom row in Table 6. We can see that *with the extra converted data, the baseline model can achieve slightly lower accuracy with the coupled model and avoid the inefficiency problem at the meantime.*

6 Related Work

This work is partially inspired by Qiu et al. (2013), who propose a model that performs heterogeneous Chinese word segmentation and POS tagging and produces two sets of results following *CTB* and *PD* styles respectively. Different from our CRF-based coupled model, their approach adopts a linear model, which directly combines two separate sets of features based on single-side tags, without considering the interacting joint features between the two annotations. They adopt an approximate decoding algorithm which tries to find the best single-side tag sequence with reference to tags at the other side. In contrast, our approach is a direct extension of traditional CRF, and is more theoretically simple from the perspective of modelling. The use of both joint and separate features is proven to be crucial for the success of our coupled model. In addition, their work indicates that their model relies on a hand-crafted loose mapping between annotations, which is opposite to our findings. The naming of the "coupled" CRF is borrowed from the work of Qiu et al. (2012), which treats the joint task of Chinese word segmentation and POS tagging as two coupled sequence labeling problems.

Zhang et al. (2014) propose a shift-reduce dependency parsing model which can simultaneously learn and produce two heterogeneous parse trees. However, their approach assumes the existence of data with annotations at both sides, which is obtained by converting phrase-structure trees into dependency trees with different heuristic rules.

This work is also closely related with multi-task learning, which aims to jointly learn multiple related tasks with the benefit of using interactive features under a share representation (Ben-David and Schuller, 2003; Ando and Zhang, 2005; Parameswaran and Weinberger, 2010). However, according to our knowledge, multi-task learning typically assumes the existence of data with labels for multiple tasks at the same time, which is unavailable in our situation.

As one reviewer kindly pointed out that our model is a factorial CRF (Sutton et al., 2004), in the sense that the bundled tags can be factorized two connected latent variables. Initially, factorial CRFs are designed to jointly model two related (and typically hierarchical) sequential labeling tasks, such as POS tagging and chunking. In this

work, our coupled CRF jointly models two same tasks which have different annotation schemes. Moreover, this work provides a natural way to learn from incomplete annotations where one sentence only contains one-side labels. The reviewer also suggests that our objective can be optimized with the latent variable structured perceptron of Sun et al. (2009), which we leave as future work.

Learning with ambiguous labelings are previously explored for classification (Jin and Ghahramani, 2002), sequence labeling (Dredze et al., 2009), parsing (Riezler et al., 2002; Täckström et al., 2013; Li et al., 2014a; Li et al., 2014b). Recently, researchers derive natural annotations from web data, transform them into ambiguous labelings to supervise Chinese word segmentation models (Jiang et al., 2013; Liu et al., 2014; Yang and Vozila, 2014).

7 Conclusions

This paper proposes an effective coupled sequence labeling model for exploiting multiple non-overlapping datasets with heterogeneous annotations. Please note that our model can also be naturally trained on datasets with both-side annotations if such data exists. Experimental results demonstrate that our model work better than the baseline and guide-feature based methods on both one-side POS tagging and annotation conversion. Specifically, detailed analysis shows several interesting findings. First, both the separate features and joint features are indispensable components for the success of our coupled model. Second, our coupled model does not rely on a carefully hand-crafted mapping function. Our linguistically motivated mapping function is only used to reduce the size of the bundled tag set for the sake of efficiency. Finally, using the extra training data converted with our coupled model, the baseline tagging model achieves similar accuracy improvement. In this way, we can avoid the inefficiency problem of our coupled model in real application.

For future, our immediate plan is to annotate more data with both *CTB* and *PD* tags (a few t-housand sentences), and to investigate our coupled model with small amount of such annotation as extra training data. Meanwhile, Algorithm 1 is empirically effective in merging two training data, but still needs manual tuning of the weighting factor on held-out data. Thus, we would like to find a more principled and theoretically sound method to merge multiple training data.

Acknowledgments

The authors would like to thank the undergraduate students Fangli Lu and Xiaojing Wang for building our annotation system, and Le Lu, Die Hu, Yue Zhang, Jian Zhang, Qiuyi Yan, Xinzhou Jiang for data annotation. We are also grateful that Yu Ding kindly shared her earlier codes on which our annotation system was built. We also thank the helpful comments from our anonymous reviewers. This work was supported by National Natural Science Foundation of China (Grant No. 61432013, 61203314) and Jiangsu Planned Projects for Postdoctoral Research Funds (No. 1401075B), and was also partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization of Jiangsu Province.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learn Research*, 6:1817–1853.

Shai Ben-David and Reba Schuller. 2003. Exploiting task relatedness for multiple task learning. In *COLT*.

Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. 2009. Sequence learning from data with multiple labels. In *ECML/PKDD Workshop on Learning from Multi-Label Data*.

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 NAACL*, pages 213–216.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging – a case study. In *Proceedings of ACL*, pages 522–530.

Wenbin Jiang, Meng Sun, Yajuan Lü, Yating Yang, and Qun Liu. 2013. Discriminative learning with natural annotations: Word segmentation as a case study. In *Proceedings of ACL*, pages 761–769.

Rong Jin and Zoubin Ghahramani. 2002. Learning with multiple labels. In *Proceedings of NIPS*.

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

Zhenghua Li, Min Zhang, Wanxiang Che, and Ting Liu. 2012. A separately passive-aggressive training algorithm for joint POS tagging and dependency parsing. In *COLING*, pages 1681–1698.

Zhenghua Li, Min Zhang, and Wenliang Chen. 2014a. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In *ACL*, pages 457–467.

Zhenghua Li, Min Zhang, and Wenliang Chen. 2014b. Soft cross-lingual syntax projection for dependency parsing. In *COLING*, pages 783–793.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *Proceedings of EMNLP*, pages 864–874.

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

Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses: An introduction*. John Wiley & Sons, Inc., New York.

S. Parameswaran and K.Q. Weinberger. 2010. Large margin multi-task metric learning. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1867–1875.

Ciyang Qing, Ulle Endriss, Raquel Fernandez, and Justin Kruger. 2014. Empirical analysis of aggregation methods for collective annotation. In *COLING*, pages 1533–1542.

Xipeng Qiu, Feng Ji, Jiayi Zhao, and Xuanjing Huang. 2012. Joint segmentation and tagging with coupled sequences labeling. In *Proceedings of COLING 2012: Posters*, pages 951–964, Mumbai, India.

Xipeng Qiu, Jiayi Zhao, and Xuanjing Huang. 2013. Joint Chinese word segmentation and POS tagging on heterogeneous annotated corpora with multiple task learning. In *Proceedings of EMNLP*, pages 658–668.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. III Maxwell, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of ACL*, pages 271–278.

Anders Søgaard. 2011. Semi-supervised condensed nearest neighbor for part-of-speech tagging. In *Proceedings of ACL*, pages 48–52.

Weiwei Sun and Hans Uszkoreit. 2012. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate Chinese part-of-speech tagging. In *Proceedings of ACL*, pages 242–252.

Weiwei Sun and Xiaojun Wan. 2012. Reducing approximation and estimation errors for Chinese lexical processing with heterogeneous annotations. In *Proceedings of ACL*, pages 232–241.

Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun'ichi Tsujii. 2009. Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1236–1242.

Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *International Conference on Machine Learning (ICML)*.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of NAACL*, pages 1061–1071.

Fei Xia. 2000. The part-of-speech tagging guidelines for the penn Chinese treebank 3.0. In *Technical Report, Linguistic Data Consortium*.

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

Fan Yang and Paul Vozila. 2014. Semi-supervised Chinese word segmentation using partial-label learning with conditional random fields. In *Proceedings of EMNLP*, pages 90–98.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL-08: HLT*, pages 888–896.

Meishan Zhang, Wanxiang Che, Yanqiu Shao, and Ting Liu. 2014. Jointly or separately: Which is better for parsing heterogeneous dependencies? In *Proceedings of COLING*, pages 530–540.