Improving Chinese Word Segmentation Using Partially Annotated Sentences

Kaixu Zhang[1], Jinsong Su[1], and Changle Zhou[2]

[1] Xiamen University, Xiamen, Fujian, 361005, China
kareyzhang@gmail.com, jssu@xmu.edu.cn
[2] Institute of Artificial Intelligence
Xiamen University, Xiamen, Fujian, 361005, China
dozero@xmu.edu.cn

Abstract. Manually annotating is important for statistical NLP models but time-consuming and labor-intensive. We describe a learning task that can use partially annotated data as the training data. Traditional supervised learning task is a special case of such task. Particularly, we adapt the perceptron algorithm to train Chinese word segmentation models. We mix conventional fully segmented Chinese sentences with partially annotated sentences as the training data. Partially annotated sentences can be automatically generated from the heterogeneous segmented corpora as well as naturally annotated data such as markup language sentences like wikitexts without any additional manual annotating. The experiments show that our method improves the performances of both supervised model and semi-supervised models.

Keywords: naturally annotation, Chinese word segmentation

1 Introduction

Chinese words in sentences are not explicitly separated by spaces. Chinese word segmentation is the preliminary task to segment Chinese sentences into words in order to do deeper processing such as part-of-speech tagging and parsing.

Like other natural language processing tasks based on statistical machine learning, annotated data is crucial for the performance of word segmentation. But annotated corpora are limited in size and scope due to the time-consuming and labor-intensive annotating.

Besides semi-supervised methods, using heterogeneous segmented corpus to improve word segmentation is therefore researched in order to make use of more annotated data [1,2]. For example, annotated sentence (b) in Figure 1 can be used together with (a) by using annotation adaptation methods, though the two annotations are not totally consistent. But still, fully annotated data is limited.

On the other hand, there are abundant naturally annotated sentences that also contain clues for word segmentation. For example, the sentence (c) in Figure 1 with brackets is from the Chinese Wikipedia. The bracketed phrase "中美关系" used to make a hyperlink to another page is also a valid phrase in sentence (a) and

(b). Comparing to the sentences intentionally segmented by skilled annotators to make training data, partially annotated sentences by general netizens can be obtained easily. Another example is the anchor texts in the HTML files on the Web, which is indeed web-scale.

(a) CTB: ··· 保证␣中␣美␣关系␣沿着 ···
(b) MSR: ··· 发展␣中美␣关系␣的 ···
(c) Wikipedia: ··· 发展[[中美关系]]的 ···

Fig. 1. Different annotated sentences containing the same phrase 中美关系 (Sino-US relations) will be treated in a unified way as the training data in this paper.

There are mainly two difficulties to use such partially annotated sentences to improve Chinese word segmentation: the learning algorithm needs to be adapted to learn from partial annotations; a relation is needed to bridge the gap between the arbitrary annotation by netizens and the annotation of words.

Motivated by the related work [3], we formally define the learning task using partially annotated data and propose an adapted perceptron algorithm that can learn from partially annotated data for both supervised and semi-supervised learning (Section 2). Fully and partially annotated sentences are mixed and not distinguished in this algorithm.

A span-based representation is used to represent the information of the partially annotated sentences for word segmentation (Section 3). In such representation, sentences annotated with brackets (Figure 1 (c)) and sentences from heterogeneous corpus (Figure 1 (b)) can be treated in the same way.

With a word-based word segmentation model, experiments are conducted on the Chinese Treebank 5 (Section 5). Sentences from the MSR corpus, People's Daily corpus as well as Baidu Baike (a Chinese wikipedia-like website) are used as partially annotated sentences to improve the performance of the baseline model.

Our contribution is twofold: 1) we proposed an algorithm for word segmentation with partially annotated data which can treat various resources as partially annotated data; 2) we use the naturally annotated sentences provided by common netizens as a resource to improve the performance of word segmentation.

2 Learning with Partially Annotated Data

2.1 Partially Annotated Data as Training Data

The training examples of supervised classification are $\{(x_i, y_i)\}$, where $y_i \in \text{GEN}(x_i)$ and $\text{GEN}(x_i)$ is the set of all possible classes for x_i. For any input x_i, a unique y_i is given as the gold standard output.

For a partially annotated example x_i, the unique gold standard output can not be determined by using only the partial annotation. Instead, a nonempty

Inputs: training example $\{(x_i, Y_i)\}$
Initialization: set $\Lambda = \mathbf{0}$
Output: Averaged parameters $\frac{\sum \Lambda_i^t}{TN}$
1: **for** $t = 1 \ldots T$, $i = 1 \ldots N$
2: calculate $z_i = \arg\max_{z \in \mathbf{GEN}(x_i)} \Phi(x_i, z) \cdot \Lambda$
3: **if** $z_i \notin Y_i$ **then**
4: calculate $y_i = \arg\max_{y \in Y_i} \Phi(x_i, y) \cdot \Lambda$
5: $\Lambda = \Lambda - \Phi(x_i, z_i) + \Phi(x_i, y_i)$
6: **set** $\Lambda_i^t = \Lambda$

Fig. 2. Averaged perceptron algorithm used for learning from partially annotated data.

subset Y_i of $\mathrm{GEN}(x_i)$ which contains the unknown gold standard output is given. The training examples are thus represented as $\{(x_i, Y_i)\}$, where $\emptyset \subset Y_i \subset \mathrm{GEN}(x_i)$ and $y_i \in Y_i$.

A full annotated example can be seen as a special partially annotated example where $Y_i = \{y_i\}$.

2.2 Perpectron Algorithm for Partially Annotated Data

Collins [4] proposed a perceptron algorithm for structured classification tasks such as part-of-speech tagging. Since it is widely used for Chinese word segmentation, we decide to adapt this algorithm for partially annotated data.

The adapted algorithm is shown in Figure 2 which is similar to the related work [3]. The adapted algorithm is a natural extension of the traditional one [4]. When $Y_i = \{y_i\}$ holds for all the training example, the adapted perceptron algorithm degenerates to the traditional one.

This algorithm can not learn from unannotated sentences. For any unannotated example $(x_i, \mathrm{GEN}(x_i))$, since $z_i \in \mathrm{GEN}(x_i)$ is always true, the updating in the **if** statement will never be executed.

Additionally, we find that the convergence to the expected optimum of this adapted algorithm is not theoretically guaranteed. But fortunately, this algorithm works well in practice as we will show.

2.3 Self-Training with Partially Annotated Data

Partially annotated sentences can be also used for semi-supervised algorithms such as self-training.

Figure 3 shows a self-training algorithm which uses partially annotated sentences in the training process. In Step 4, we use the margin to define the confidence:

$$\mathrm{conf}_i = \Phi(x_i, z_i) \cdot \Lambda - \max_{z \neq z_i} \Phi(x_i, z) \cdot \Lambda \tag{1}$$

There are two differences between our algorithm and the conventional self-training algorithm. First, in Step 2, examples that not fully annotated in P are also used to train the model (we call it "p_train" in the experiments). Second,

Inputs:
Fully annotated example set $\mathcal{F}$
Partially annotated example set $\mathcal{P}$
Output:
Model parameter Λ
Algorithm:
1: Loop for k-iterations:
2: use $\mathcal{F}$ and $\mathcal{P}$ to train parameter Λ'
3: use Λ' to segment $\mathcal{P}$
4: move q examples with high confidence from $\mathcal{P}$ to $\mathcal{F}$
5: use $\mathcal{F}$ to train parameter Λ

Fig. 3. Self-training algorithm with partially annotated data.

in Step 3, when segmenting an example (x_i, Y_i) in P, the search space of the decoding is limited in the set Y_i (we call it "p_predict" in the experiments).

2.4 Distributed Learning for Large-Scale Training Data

As we mentioned that available partially annotated sentences are large-scale, there are two reasons to use distributed learning for large-scale training data. First, when partially annotated sentences are much more than fully annotated sentences, the learning is harder to converge. Second, our distributed method is faster and is suitable for incremental learning.

Suppose we have n sets of training examples. The parameters of the trained model using these sets are $\Lambda^{(1)} \dots \Lambda^{(n)}$, respectively. Then we calculate the parameters of the final model as:

$$\lambda_i = \frac{\sum_{k=1\dots n} \lambda_i^{(k)}}{\sum_{k=1\dots n}^{\lambda_i^{(k)} \neq 0} 1} \tag{2}$$

where $\lambda_i^{(k)}$ is the i-th parameter of the k-th model. Note that the denominator in this equation is not n. The reason is that when $\lambda_i^{(k)} = 0$, it is not because this feature is not important, but because this feature is unseen in the training process of the k-th training data.

For incremental learning, used sentences are not needed to be stored. When new training data is acquired, we only need to train a new model $\Lambda^{(n+1)}$ using the new data and then update the parameter Λ without using any old data.

3 Partially Annotated Sentences for Chinese Word Segmentation

Now we narrow down our discussion to the Chinese word segmentation task.

A raw sentence x is a Chinese sentence where no spaces are presented to seperate words, while a segmented sentence is a sentence where words are separated by spaces. For example, "发展中美关系" is a raw sentence, and "␣发展␣中␣美␣关系␣" is one of the possible segmented sentences corresponding to the raw sentence.

We use a span set (a set of spans) as z to represents words in a segmented sentence. The corresponding span set z for the segmented sentence above is $\{\langle 0,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \langle 4,6\rangle\}$, the two numbers in $\langle 0,2\rangle$ are the indexes of the beginning and end of the word "发展".

For a training sentence x with partial annotation such as heterogeneous segmented corpus or wikitexts, y can not be determined. Instead, we need to define the set $Y = \{z_j\}$ which include the gold standard output ($y \in Y$) and exclude some impossible outputs. Before giving the definition of Y, we introduce some basis concepts in the next subsection.

3.1 Agreement between Span Sets

We used spans in the span set z to indicate words. Note that spans can also be used to indicate other linguistic components such as morphemes and phrases. Roughly speaking, all these components of a sentence are organized as a hierarchical tree. In other words, following the definition by Klein and Manning [5], the spans of any two components will never cross:

Definition 1 Two spans $\langle b, e\rangle$ and $\langle b', e'\rangle$ (without loss of generality, $b \le b'$) *disagree* (or cross) if and only if $b < b' < e$ and $b' < e < e'$. Otherwise, they *agree* (or are non-cross).

For example, span $\langle 2,5\rangle$ agree with $\langle 0,2\rangle$ and $\langle 2,4\rangle$ but disagree with $\langle 0,3\rangle$.

Furthermore, we can define the "agree" relation between span sets:

Definition 2 Two span sets S and S' *agree* (denoted as $S \sim S'$) if and only if any span in S agrees with any span in S'.

Then we can give an assumption about the nature of Chinese:

Assumption 0 Span set of words agree with span set of morphemes or phrases in the same sentence.

This assumption may be widely accepted by linguists and is the basis assumption to define partially annotated examples. Although we do not directly use this assumption in this paper, it is the motivation that we choice spans and the "agree" relation to define Y. In the following subsections we will give two more assumptions based on our observation. They may not always hold as the previous one but can be used to define Y effectively.

3.2 Partially Annotated Sentences from Heterogeneous Segmented Corpus

For a sentence x from a heterogeneous segmented corpus, the given gold standard output y' may be different with the gold standard output y we expected. Using (x, y') as the training data will result in bad performance.

We establish a relation between y' and y by given the following assumption:

Assumption 1: Span sets of words in different annotation specifications agree with each other ($y \sim y'$).

This assumption is based on that most of the inconsistency of word definition between different annotation specifications is about the granularity of words. This means that a word under one annotation specification is generally still a word, phrase or morpheme in another annotation specification.

Thus we can define partially annotated examples as

$$(x, Y) := (x, \{z | z \sim y'\}) \tag{3}$$

The set Y is defined by using heterogeneous annotation y'. And with Assumption 1, we have $y \in Y$.

3.3 Partially Annotated Sentences from Wikitexts

Based on our observation, although the annotation in wikitexts is more arbitrary, bracketed texts are still usually words, phrases or morphemes. So similar to the method we used for the heterogeneous corpora, we give an assumption:

Assumption 2: Span set of words and span set of bracketed texts of the same sentence agree with each other.

The partially annotated sentences can thus be defined as

$$(x, Y) := (x, \{z | z \sim b\}) \tag{4}$$

where b is the span set of bracketed texts.

3.4 Mixed Training Data

It is not sufficient that we only use partially annotated sentences defined above as the training data. If so, the algorithm may result in an unexpected optimum that the model segments every single character as a word. Those results will agree with any span sets.

In practice, we mix the fully segmented sentences and the partially annotated sentences and randomly shuffle them as the training data. And our learning algorithm can treat them in the same way.

4 Related Work

In recent years, learning with partially annotated data is concerned by researchers of machine learning [6] as well as natural language processing. Partially annotated data can be used for corpus construction [7], sequence labeling [8], syntactic parsing [9,10] and other NLP tasks [11]. Our algorithm can be seen as a version of the latent structure perceptron [12] which can learn from examples with hidden variables [3]. Zettlemoyer and Collins [13] used similar algorithm for semantic parsing.

We use self-training [14] for our task. Other semi-supervised learning methods such as co-training [15] may also benefit from partially annotated sentences with our method.

Jiang et al. [16] used model trained using heterogeneous segmented corpus to generate new features to improve the performance of joint word segmentation and part-of-speech tagging model. Sun and Wan [2] further used the re-training method to transform the heterogeneous corpus in order to use it directly as the training data. Jiang et al. [1] further used iterative annotation transformation with predict-self reestimation to improve the performance.

New features for word segmentation can also be generated based on the statistical information of the unannotated corpus [17,18,19,20]. Punctuation marks can be seen as artificial annotations for natural language. Li and Sun [21] used the punctuation marks in the unsegmented corpus as clues for word boundaries. Spitkovsky et al. [22] used hyper-text annotations for unsupervised English parsing.

Our model for word segmentation [23] is mainly motived by the word-based word segmentation model proposed by Zhang and Clark [24,25] and the linear-time incremental shift-reduce parser proposed by Huang and Sagae [26].

5 Experiments

We use Penn Chinese Treebank 5 (CTB) as the main corpus of our experiments. The partitions of training set (18,086 sentences) and test set are the same with [25]. We use the training data of the MSR corpus from SIGHAN bake-off 2005 (86,924 sentences) and People's Daily (PD) corpus from Peking University (294,239 sentences) as the heterogeneous corpora.

As the source files of Chinese Wikipedia contain both simplified and traditional Chinese characters and the translation method is not straightforward, we turn to use another wiki-like site Baike from Baidu[3] containing only simplified characters. One million sentences with brackets are used, which is still a small part of the total sentences with brackets that can be extracted.

We use a word-based Chinese word segmentation system [23] which won the first place in the CIPS-SIGHAN bakeoff 2012 [27] as our baseline model[4]. The feature templates are listed in table 1. In all the semi-supervised experiments the parameter k in the self-training algorithm (Figure 3) is set to 10. Since there are no hyper-parameters that are tuned, we directly show the results on the test set instead of the development set.

F-score [28] is used for the evaluation.

5.1 Supervised Learning with Partially Annotated Data

We mix the training data of CTB and the partially annotated sentences generated from other resources together as the training data for supervised learning.

[3] http://baike.baidu.com/

[4] https://github.com/zhangkaixu/isan

action-based	$\langle \mathbf{a01}, a_{i-2}, a_{i-1} \rangle$
character-based	$\langle \mathbf{c01}, c_{i-2}, a_{i-1} \rangle$, $\langle \mathbf{c02}, c_{i-1}, a_{i-1} \rangle$, $\langle \mathbf{c03}, c_i, a_{i-1} \rangle$
	$\langle \mathbf{c04}, c_{i-3}, c_{i-2}, a_{i-1} \rangle$, $\langle \mathbf{c05}, c_{i-2}, c_{i-1}, a_{i-1} \rangle$,
	$\langle \mathbf{c06}, c_{i-1}, c_i, a_{i-1} \rangle$, $\langle \mathbf{c07}, c_i, c_{i+1}, a_{i-1} \rangle$
word-based	$\langle \mathbf{w01}, \mathbf{w}_0 \rangle$, $\langle \mathbf{w02}, \lvert\mathbf{w}_0\rvert \rangle$
	$\langle \mathbf{w03}, \lvert\mathbf{w}_0\rvert, \mathbf{w}_0[0] \rangle$, $\langle \mathbf{w04}, \lvert\mathbf{w}_0\rvert, \mathbf{w}_0[-1] \rangle$, $\langle \mathbf{w05}, \mathbf{w}_0[0], \mathbf{w}_0[-1] \rangle$
	$\langle \mathbf{w06}, \mathbf{w}_{-1}[-1], \mathbf{w}_0[-1] \rangle$, $\langle \mathbf{w07}, \lvert\mathbf{w}_{-1}\rvert, \mathbf{w}_0 \rangle$, $\langle \mathbf{w08}, \mathbf{w}_{-1}, \lvert\mathbf{w}_0\rvert \rangle$
	$\langle \mathbf{w09}, \mathbf{w}_0[0], c_i \rangle$, $\langle \mathbf{w10}, \mathbf{w}_0[-1], c_i \rangle$

Table 1. Feature templates.

Training Set	F1
CTB	0.9745
CTB + $\text{MSR}_{\text{partial}}$	0.9767
CTB + $\frac{1}{4}\text{PD}_{\text{partial}}$	0.9773
CTB + $\text{PD}_{\text{partial}}$	0.9752
CTB + $\text{Baike\_50K}_{\text{partial}}$	0.9761

Table 2. Results of supervised learning with partially annotated sentences.

Table 2 shows the results. Partially annotated sentences generated from both heterogeneous corpus and wikitexts can improve the performance. Note that the PD corpus is much large than the MSR corpus. Probably because the converging is harder when the rate of partially annotated sentence is high, we find that using a quarter of these sentences are even better than using all at once.

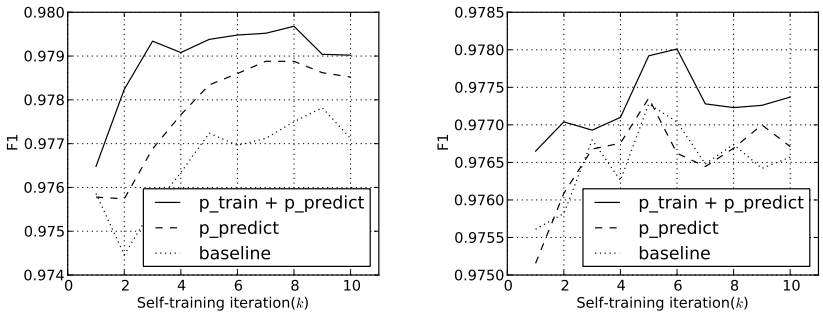
5.2 Self-Training with Partially Annotated Data

Three different self-training algorithms are performed and compared. The conventional self-training algorithm without using any partially annotated information is denoted as "baseline". The self-training algorithm proposed by us in Figure 4 is denoted as "p_train+p_predict". We also use an algorithm like "p_train+p_predict" but does not use partially annotated sentences in the training process which is denoted as "p_predict".

Experiment results are shown in Figure 4. All these three self-training algorithms outperform the supervised algorithm. The algorithm "p_train+p_predict" can always improve the performance. For a corpus like MSR where the partial annotation is relatively rich, only using "p_predict" can also improve the performance. But for Baike_50K where annotated information is rare, the difference between the performances of "p_predict" and "baseline" is not obvious.

5.3 Distributed Learning with Large Data

From the experiment results of the supervised learning we already find that simply using large partially annotated dataset is not always helpful.



(a) CTB ← $\text{MSR}_{\text{partial}}$ (b) CTB ← $\text{Baike\_40K}_{\text{partial}}$

Fig. 4. Results of the self-training algorithm.

In this subsection we first divide the PD corpus into four parts and use the distributed learning to train the models.

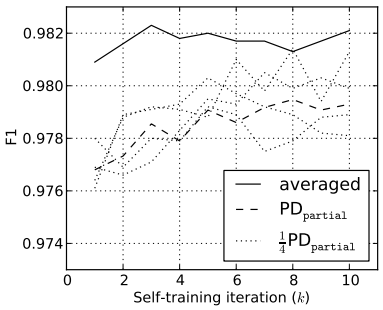


Fig. 5. Results of self-training using PD corpus. Distributed learning outperforms the baseline.

Figure 5 shows the results. The dotted line is the curve of using the whole PD corpus in the self-training algorithm. Slashed lines are four curves of using a quarter of the PD corpus, respectively. The solid line is the curve of the averaged model based on these four small models using Equation 2.

Finally, we perform the distributed self-training with one million sentences from Baike (divided into 25 sets). Table 3 shows the final results of our method and the results of related work. It is not surprise that our method do not outperform the annotation adaptation method [1], since we only treat the heterogeneous corpus as a partial annotated corpus. But with the same method, we can make use of the partial annotation information in the wikitexts. Our word segmentation model using one million Baike sentences is comparative to the joint

word segmentation and part-of-speech tagging model [20] using approximately 208 million additional words from Xinhua newswire.

Training Set	F1
CTB	0.9745
CTB + $PD_{partial}$	0.9821
CTB + $Baike\_1M_{partial}$	0.9810
CTB + PD [16]	0.9815
CTB + PD [1]	0.9843
CTB + Gigaword [20]	0.9811

Table 3. Final results of our method are compared with related work.

6 Discussion and Conclusion

We presented a learning method with partially annotated sentences for Chinese word segmentation. Naturally annotated data such as wikitexts can be treated as partially annotated sentences and be used as training data together with fully annotated sentences. Our method is potentially suitable for domain adaptation where in-domain fully segmented sentences are limited.

Our work is just a primary study that uses the partial annotation information for Chinese language processing. In the future, we will try to use similar method for word-based active learning and syntax parsing.

Acknowledgments

The authors want to thank ZHANG Junsong from the cognitive lab and SHI Xiaodong and CHEN Yidong from the NLP lab of Xiamen University for the support of experiments.

The authors are supported by NSFC (Grant No. 61273338), Research Fund for the Doctoral Program of Higher Education of China
(Grant No. 20120121120046) and Natural Science Fundation of Fujian Province (Grant No. 2010J01351).

References

1. Jiang, W., Meng, F., Liu, Q., L ü , Y.: Iterative annotation transformation with predict-self reestimation for chinese word segmentation. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, Association for Computational Linguistics (July 2012) 412–420

2. Sun, W., Wan, X.: Reducing approximation and estimation errors for chinese lexical processing with heterogeneous annotations. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jeju Island, Korea, Association for Computational Linguistics (July 2012) 232–241
3. Fernandes, E., dos Santos, C., Milidi ú , R.: Latent structure perceptron with feature induction for unrestricted coreference resolution. In: Joint Conference on EMNLP and CoNLL - Shared Task, Jeju Island, Korea, Association for Computational Linguistics (July 2012) 41–48
4. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. (2002) 1–8
5. Klein, D., Manning, C.D.: A generative constituent-context model for improved grammar induction. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, Association for Computational Linguistics (July 2002) 128–135
6. Lou, X., Hamprecht, F.: Structured learning from partial annotations. arXiv:1206.6421 (June 2012)
7. Neubig, G., Mori, S.: Word-based partial annotation for efficient corpus construction. In Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (2010)
8. Tsuboi, Y., Kashima, H., Mori, S., Oda, H., Matsumoto, Y.: Training conditional random fields using incomplete annotations. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, Coling 2008 Organizing Committee (August 2008) 897–904
9. Mirroshandel, S.A., Nasr, A.: Active learning for dependency parsing using partially annotated sentences. In: Proceedings of the 12th International Conference on Parsing Technologies, Dublin, Ireland, Association for Computational Linguistics (October 2011) 140–149
10. Flannery, D., Miayo, Y., Neubig, G., Mori, S.: Training dependency parsers from partially annotated corpora. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, Asian Federation of Natural Language Processing (November 2011) 776–784
11. Fernandes, E.R., Brefeld, U.: Learning from partially annotated sequences. In: Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I. ECML PKDD'11, Berlin, Heidelberg, Springer-Verlag (2011) 407–422
12. Yu, C.N.J., Joachims, T.: Learning structural SVMs with latent variables. In: Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09, New York, NY, USA, ACM (2009) 1169–1176
13. Zettlemoyer, L., Collins, M.: Online learning of relaxed CCG grammars for parsing to logical form. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, Association for Computational Linguistics (June 2007) 678–687
14. McClosky, D., Charniak, E., Johnson, M.: Effective self-training for parsing. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, New York City, USA, Association for Computational Linguistics (June 2006) 152–159

15. Sarkar, A.: Applying co-training methods to statistical parsing. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. NAACL '01, Stroudsburg, PA, USA, Association for Computational Linguistics (2001) 1–8
16. Jiang, W., Huang, L., Liu, Q.: Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging - a case study. In: Proceedings of the 47th ACL, Suntec, Singapore, Association for Computational Linguistics (August 2009) 522–530
17. Feng, H., Chen, K., Deng, X., Zheng, W.: Accessor variety criteria for chinese word extraction. Computational Linguistics **30**(1) (2004) 75–93
18. Zhao, H., Kit, C.: Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In: The Sixth SIGHAN Workshop on Chinese Language Processing. (2008) 106–111
19. Sun, W., Xu, J.: Enhancing chinese word segmentation using unlabeled data. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK., Association for Computational Linguistics (July 2011) 970–979
20. Wang, Y., Kazama, J., Tsuruoka, Y., Chen, W., Zhang, Y., Torisawa, K.: Improving chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, Asian Federation of Natural Language Processing (November 2011) 309–317
21. Li, Z., Sun, M.: Punctuation as implicit annotations for chinese word segmentation. Computational Linguistics **35**(4) (2009) 505–512
22. Spitkovsky, V.I., Jurafsky, D., Alshawi, H.: Profiting from mark-up: Hyper-text annotations for guided parsing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, Association for Computational Linguistics (July 2010) 1278–1287
23. Zhang, K., Sun, M., Zhou, C.: Word segmentation on chinese mirco-blog data with a linear-time incremental model. In: Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, Tianjin, China, Association for Computational Linguistics (December 2012) 4146
24. Zhang, Y., Clark, S.: Chinese segmentation with a word-based perceptron algorithm, Prague, Czech Republic, Association for Computational Linguistics (June 2007) 840–847
25. Zhang, Y., Clark, S.: Syntactic processing using the generalized perceptron and beam search. Computational Linguistics (Early Access) (2011) 1–47
26. Huang, L., Sagae, K.: Dynamic programming for linear-time incremental parsing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, Association for Computational Linguistics (July 2010) 1077–1086
27. Duan, H., Sui, Z., Tian, Y., Li, W.: The cips-sighan clp 2012 chineseword segmentation onmicroblog corpora bakeoff. In: Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, Tianjin, China, Association for Computational Linguistics (December 2012) 35–40
28. Emerson, T.: The second international chinese word segmentation bakeoff. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea (2005) 123–133