A Lattice-based Framework for Joint Chinese Word Segmentation, POS Tagging and Parsing

Zhiguo Wang[1], **Chengqing Zong**[1] and **Nianwen Xue**[2]
[1]National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100190
[2]Computer Science Department, Brandeis University, Waltham, MA 02452
{zgwang, cqzong}@nlpr.ia.ac.cn xuen@brandeis.edu

Abstract

For the cascaded task of Chinese word segmentation, POS tagging and parsing, the pipeline approach suffers from error propagation while the joint learning approach suffers from inefficient decoding due to the large combined search space. In this paper, we present a novel lattice-based framework in which a Chinese sentence is first segmented into a word lattice, and then a lattice-based POS tagger and a lattice-based parser are used to process the lattice from two different viewpoints: sequential POS tagging and hierarchical tree building. A strategy is designed to exploit the complementary strengths of the tagger and parser, and encourage them to predict agreed structures. Experimental results on Chinese Treebank show that our lattice-based framework significantly improves the accuracy of the three sub-tasks.

1 Introduction

Previous work on syntactic parsing generally assumes a processing pipeline where an input sentence is first tokenized, POS-tagged and then parsed (Collins, 1999; Charniak, 2000; Petrov and Klein, 2007). This approach works well for languages like English where automatic tokenization and POS tagging can be performed with high accuracy without the guidance of the high-level syntactic structure. Such an approach, however, is not optimal for languages like Chinese where there are no natural delimiters for word boundaries, and word segmentation (or tokenization) is a non-trivial research problem by itself. Errors in word segmentation would propagate to later processing stages such as POS tagging and syntactic parsing. More importantly, Chinese is a language that lacks the morphological clues that help determine the POS tag of a word. For example, 调查 ("investigate/investigation") can either be a verb ("investigate") or a noun ("investigation"), and there is no morphological variation between its verbal form and nominal form. This contributes to the relatively low accuracy (95% or below) in Chinese POS tagging when evaluated as a stand-alone task (Sun and Uszkoreit, 2012), and the noun/verb ambiguity is a major source of error.

More recently, joint inference approaches have been proposed to address the shortcomings of the pipeline approach. Qian and Liu (2012) proposed a joint inference approach where syntactic parsing can provide feedback to word segmentation and POS tagging and showed that the joint inference approach leads to improvements in all three sub-tasks. However, a major challenge for joint inference approach is that the large combined search space makes efficient decoding and parameter estimation very hard.

In this paper, we present a novel lattice-based framework for Chinese. An input Chinese sentence is first segmented into a word lattice, which is a compact representation of a small set of high-quality word segmentations. Then, a lattice-based POS tagger and a lattice-based parser are used to process the word lattice from two different viewpoints. We next employ the dual decomposition method to exploit the complementary strengths of the tagger and parser, and encourage them to predict agreed structures. Experimental results show that our lattice-based framework significantly improves the accuracies of the three sub-tasks

2 The Lattice-based Framework

Figure 1 gives the organization of the framework. There are four types of linguistic structures: a Chinese sentence, the word lattice, tagged word sequence and parse tree of the Chinese sentence. An example for each structure is provided in Figure 2. We can see that the terminals and pre-terminals of a parse tree constitute a tagged word sequence. Therefore, we define a comparator between a tagged word sequence and a parse tree: if they contain the same word sequence and POS tags, they are equal, otherwise unequal.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 623–627,
Sofia, Bulgaria, August 4-9 2013.

Figure 1 also shows the workflow of the framework. First, the Chinese sentence is segmented into a word lattice using the word segmentation system. Then the word lattice is fed into the lattice-based POS tagger to produce a tagged word sequence S and into the lattice-based parser to separately produce a parse tree T. We then compare S with T to see whether they are equal. If they are equal, we output T as the final result. Otherwise, the guidance generator generates some guidance orders based on the difference between S and T, and guides the tagger and the parser to process the lattice again. This procedure may iterate many times until the tagger and parser predict equal structures.

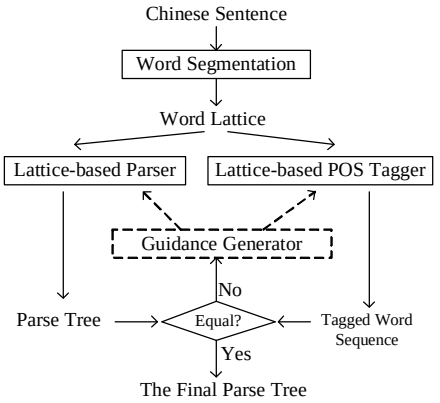

Figure 1: The lattice-based framework.

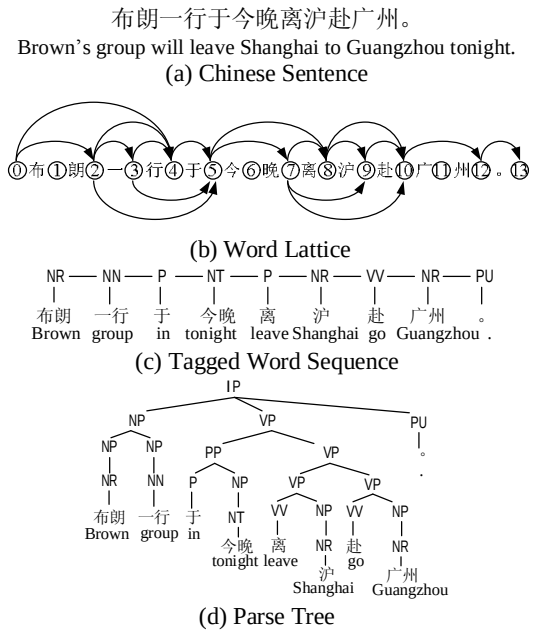

Figure 2: Linguistic structure examples.

The motivation to design such a framework is as follows. First, state-of-the-art word segmentation systems can now perform with high accuracy. We can easily get an F1 score greater than 96%, and an oracle (upper bound) F1 score greater than 99% for the word lattice (Jiang et al., 2008). Therefore, a word lattice provides us a good enough search space to allow sufficient interaction among word segmentation, POS tagging and parsing systems. Second, both the lattice-based POS tagger and the lattice-based parser can select word segmentation from the word lattice and predict POS tags, but they do so from two different perspectives. The lattice-based POS tagger looks at a path in a word lattice as a sequence and performs sequence labeling based on linear local context, while the lattice-based parser builds the parse trees in a hierarchical manner. They have different strengths with regard to word segmentation and POS tagging. We hypothesize that exploring the complementary strengths of the tagger and parser would improve each of the sub-tasks.

We build a character-based model (Xue, 2003) for the word segmentation system, and treat segmentation as a sequence labeling task, where each Chinese character is labeled with a tag. We use the tag set provided in Wang et al. (2011) and use the same feature templates. We use the Maximum Entropy (ME) model to estimate the feature weights. To get a word lattice, we first generate N-best word segmentation results, and then compact the N-best lists into a word lattice by collapsing all the identical words into one edge. We also assign a probability to each edge, which is calculated by multiplying the tagging probabilities of each character in the word.

The goal of the lattice-based POS tagger is to predict a tagged word sequence S for an input word lattice L:

$$\hat{S} = \underset{S \in cand(L)}{\operatorname{argmax}} \mathbf{w} \cdot \boldsymbol{f}(S)$$

where $cand(L)$ represents the set of all possible tagged word sequences derived from the word lattice L. $\boldsymbol{f}(S)$ is used to map S onto a global feature vector, and $\mathbf{w}$ is the corresponding weight vector. We use the same non-local feature templates used in Jiang et al. (2008) and a similar decoding algorithm. We use the perceptron algorithm (Collins, 2002) for parameter estimation.

Goldberg and Elhadad (2011) proposed a lattice-based parser for Heberw based on the PCFG-LA model (Matsuzaki et al., 2005). We adopted their approach, but found the unweighted word lattice their parser takes as input to be ineffective for our Chinese experiments. Instead, we use a weighted lattice as input and weigh each edge in the lattice with the word probability. In our model, each syntactic category A is split into multiple subcategories $A[x]$ by labeling a latent annotation x. Then, a parse tree

T is refined into $T[\boldsymbol{X}]$, where $\boldsymbol{X}$ is the latent annotation vector for all non-terminals in T. The probability of $T[\boldsymbol{X}]$ is calculated as:

$$p(T[\boldsymbol{X}]) = \prod p(A[x] \rightarrow B[y]C[z]) \times \prod p(D[x] \rightarrow w) \times \prod p(w)$$

where the three terms are products of all syntactic rule probabilities, lexical rule probabilities and word probabilities in $T[\boldsymbol{X}]$ respectively.

3 Combined Optimization Between The Lattice-based POS Tagger and The Lattice-based Parser

We first define some variables to make it easier to compare a tagged word sequence S with a parse tree T. We define $\mathcal{P}$ as the set of all POS tags. For S, we define $s(i,j,p)$=1 if S contains a POS tag $p \in \mathcal{P}$ spanning from the i-th character to the j-th character, otherwise $s(i,j,p) = 0$. We also define $s(i,j,\#) = 1$ if S contains the word spanning from the i-th character to the j-th character, otherwise $s(i,j,\#) = 0$. Similarly, for T, we define $t(i,j,p)$=1 if T contains a POS tag $p \in \mathcal{P}$ spanning from the i-th character to the j-th character, otherwise $t(i,j,p) = 0$. We also define $t(i,j,\#) = 1$ if T contains the word spanning from the i-th character to the j-th character, otherwise $t(i,j,\#) = 0$. Therefore, S and T are equal, only if $s(i,j,p) = t(i,j,p)$ for all $i \in [0,n]$, $j \in [i+1,n]$ and $p \in \mathcal{P} \cup \#$, otherwise unequal.

Our framework expects the tagger and the parser to predict equal structures and we formulate it as a constraint optimization problem:

$$(\hat{S},\hat{T}) = \underset{S,T}{\operatorname{argmax}}\ f_1(S) + f_2(T)$$

Such that for all $i \in [0,n]$, $j \in [i+1,n]$ *and* $p \in \mathcal{P} \cup \#$*:*

$$s(i,j,p) = t(i,j,p)$$

where $f_1(S) = \mathbf{w} \cdot \boldsymbol{f}(S)$ is a scoring function from the viewpoint of the lattice-based POS tagger, and $f_2(T) = \log p(T)$ is a scoring function from the viewpoint of the lattice-based parser.

The dual decomposition (a special case of Lagrangian relaxation) method introduced in Komodakis et al. (2007) is suitable for this problem. Using this method, we solve the primal constraint optimization problem by optimizing the *dual* problem. First, we introduce a vector of Lagrange multipliers $\mu(i,j,p)$ for each equality constraint. Then, the Lagrangian is formulated as:

$$L(S,T,\mu) = f_1(S) + f_2(T) + \sum_{i,j,p} \mu(i,j,p)(s(i,j,p) - t(i,j,p))$$

By grouping the terms that depend on S and T, we rewrite the Lagrangian as

$$L(S,T,\mu) = \left(f_1(S) + \sum_{i,j,p} \mu(i,j,p)s(i,j,p)\right) + \left(f_2(T) - \sum_{i,j,p} \mu(i,j,p)t(i,j,p)\right)$$

Then, the *dual objective* is

$$L(\mu) = \max_{S,T} L(S,T,\mu) = \max_{S}\left(f_1(S) + \sum_{i,j,p} \mu(i,j,p)s(i,j,p)\right) + \max_{T}\left(f_2(T) - \sum_{i,j,p} \mu(i,j,p)t(i,j,p)\right)$$

The dual problem is to find $\min_{\mu} L(\mu)$.

We use the subgradient method (Boyd et al., 2003) to minimize the dual. Following Rush et al. (2010), we define the subgradient of $L(\mu)$ as:

$\gamma(i,j,p) = s(i,j,p) - t(i,j,p)$ for all (i,j,p)

Then, adjust $\mu(i,j,t)$ as follows:

$\mu'(i,j,p) = \mu(i,j,p) - \delta(s(i,j,p) - t(i,j,p))$

where δ>0 is a step size.

Algorithm 1: Combined Optimization

1: Set $\mu^{(0)}(i,j,p)$=0, for all $\mu(i,j,p)$
2: **For** k=1 **to** K
3: $\hat{S}^{(k)} \leftarrow \operatorname{argmax}_S\left(f_1(S) + \sum_{i,j,p} \mu^{(k-1)}(i,j,p)s(i,j,p)\right)$
4: $\hat{T}^{(k)} \leftarrow \operatorname{argmax}_T\left(f_2(T) - \sum_{i,j,p} \mu^{(k-1)}(i,j,p)t(i,j,p)\right)$
5: **If** $s^{(k)}(i,j,p) = t^{(k)}(i,j,p)$ for all (i,j,p)
6: **Return** $(\hat{S}^{(k)}, \hat{T}^{(k)})$
7: **Else**
8: $\mu^{(k)}(i,j,p) = \mu^{(k-1)}(i,j,p) - \delta(s^{(k)}(i,j,p) - t^{(k)}(i,j,p))$

Algorithm 1 presents the subgradient method to solve the dual problem. The algorithm initializes the Lagrange multiplier values with 0 (line 1) and then iterates many times. In each iteration, the algorithm finds the best $\hat{S}^{(k)}$ and $\hat{T}^{(k)}$ by running the lattice-based POS tagger (line 3) and the lattice-based parser (line 4). If $\hat{S}^{(k)}$ and $\hat{T}^{(k)}$ share the same tagged word sequence (line 5), then the algorithm returns the solution (line 6). Otherwise, the algorithm adjusts the Lagrange multiplier values based on the differences between $\hat{S}^{(k)}$ and $\hat{T}^{(k)}$ (line 8). A crucial point is that the argmax problems in line 3 and line 4 can be solved efficiently using the original decoding algorithms, because the Lagrange multiplier can be regarded as adjustments for lexical rule probabilities and word probabilities.

4 Experiments

We conduct experiments on the Chinese Treebank Version 5.0 and use the standard data split

(Petrov and Klein, 2007). The traditional evaluation metrics for POS tagging and parsing are not suitable for the joint task. Following with Qian and Liu (2012), we redefine *precision* and *recall* by computing the span of a constituent based on character offsets rather than word offsets.

4.1 Performance of the Basic Sub-systems

We train the word segmentation system with 100 iterations of the Maximum Entropy model using the OpenNLP toolkit. Table 1 shows the performance. It shows that our word segmentation system is comparable with the state-of-the-art systems and the upper bound F1 score of the word lattice exceeds 99.6%. This indicates that our word segmentation system can provide a good search space for the lattice-based POS tagger and the lattice-based parser.

	P	R	F
(Kruengkrai et al., 2009)	97.46	98.29	97.87
(Zhang and Clark, 2010)	-	-	97.78
(Qian and Liu, 2012)	97.45	98.24	97.85
(Sun, 2011)	-	-	98.17
Our Word Seg. System	96.97	98.06	97.52
Word Lattice Upper Bound	99.55	99.75	99.65

Table 1: Word segmentation evaluation.

To train the lattice-based POS tagger, we generate the word lattice for each sentence in the training set using cross validation approach. We divide the entire training set into 18 folds on average (each fold contains 1,000 sentences). For each fold, we segment each sentence in the fold into a word lattice by compacting 20-best segmentation list produced with a model trained on the other 17 folds. Then, we train the lattice-based POS tagger with 20 iterations of the average perceptron algorithm. Table 2 presents the joint word segmentation and POS tagging performance and shows that our lattice-based POS tagger obtains results that are comparable with state-of-the-art systems.

	P	R	F
(Kruengkrai et al., 2009)	93.28	94.07	93.67
(Zhang and Clark, 2010)	-	-	93.67
(Qian and Liu, 2012)	93.1	93.96	93.53
(Sun, 2011)	-	-	94.02
Lattice-based POS tagger	93.64	93.87	93.75

Table 2: POS tagging evaluation.

We implement the lattice-based parser by modifying the Berkeley Parser, and train it with 5 iterations of the split-merge-smooth strategy (Petrov et al., 2006). Table 3 shows the performance, where the "Pipeline Parser" represents the system taking one-best segmentation result from our word segmentation system as input and "Lattice-based Parser" represents the system taking the compacted word lattice as input. We find the lattice-based parser gets better performance than the pipeline system among all three sub-tasks.

		P	R	F
Pipeline Parser	Seg.	96.97	98.06	97.52
	POS	92.01	93.04	92.52
	Parse	80.86	81.47	81.17
Lattice-based Parser	Seg.	97.73	97.66	97.70
	POS	93.24	93.18	93.21
	Parse	81.83	81.71	81.77

Table 3: Parsing evaluation.

4.2 Performance of the Framework

For the lattice-based framework, we set the maximum iteration in Algorithm 1 as $K = 20$. The step size δ is tuned on the development set and empirically set to be 0.8. Table 4 shows the parsing performance on the test set. It shows that the lattice-based framework achieves improvement over the lattice-based parser alone among all three sub-tasks: 0.16 points for word segmentation, 1.19 points for POS tagging and 1.65 points for parsing. It also outperforms the lattice-based POS tagger by 0.65 points on POS tagging accuracy. Our lattice-based framework also improves over the best joint inference parsing system (Qian and Liu, 2012) by 0.57 points.

		P	R	F
(Qian and Liu, 2012)	Seg.	97.56	98.36	97.96
	POS	93.43	94.2	93.81
	Parse	83.03	82.66	82.85
Lattice-based Framework	Seg.	97.82	97.9	97.86
	POS	94.36	94.44	**94.40**
	Parse	83.34	83.5	**83.42**

Table 4: Lattice-based framework evaluation.

5 Conclusion

In this paper, we present a novel lattice-based framework for the cascaded task of Chinese word segmentation, POS tagging and parsing. We first segment a Chinese sentence into a word lattice, then process the lattice using a lattice-based POS tagger and a lattice-based parser. We also design a strategy to exploit the complementary strengths of the tagger and the parser and encourage them to predict agreed structures. Experimental results show that the lattice-based framework significantly improves the accuracies of the three tasks. The parsing accuracy of the framework also outperforms the best joint parsing system reported in the literature.

Acknowledgments

The research work has been funded by the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2011AA01A207, 2012AA011101, and 2012AA011102 and also supported by the Key Project of Knowledge Innovation Program of Chinese Academy of Sciences under Grant No.KGZD-EW-501. This work is also supported in part by the DAPRA via contract HR0011-11-C-0145 entitled "Linguistic Resources for Multilingual Processing".

References

S. Boyd, L. Xiao and A. Mutapcic. 2003. Subgradient methods. Lecture notes of EE392o, Stanford University.

E. Charniak. 2000. A maximum–entropy–inspired parser. In NAACL '00, page 132–139.

Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proc. of EMNLP2002, pages 1-8.

Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFG-LA lattice parser. In Proc. of ACL2011.

Wenbin Jiang, Haitao Mi and Qun Liu. 2008. Word lattice reranking for Chinese word segmentation and part-of-speech tagging. In Proc. of Coling 2008, pages 385-392.

Komodakis, N., Paragios, N., and Tziritas, G. 2007. MRF optimization via dual decomposition: Message-passing revisited. In ICCV 2007.

C. Kruengkrai, K. Uchimoto, J. Kazama, Y. Wang, K. Torisawa and H. Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In Proc. of ACL2009, pages 513-521.

Takuya Matsuzaki, Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In Proc. of ACL2005, pages 75-82.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In Proc. of ACL2006, pages 433-440.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In Proc. of NAACL2007, pages 404-411.

Xian Qian and Yang Liu. 2012. Joint Chinese Word segmentation, POS Tagging Parsing. In Proc. of EMNLP 2012, pages 501-511.

Alexander M. Rush, David Sontag, Michael Collins and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In Proc. of EMNLP2010, pages 1-11.

Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In Proc. of ACL2011, pages 1385-1394.

Weiwei Sun and Hans Uszkoreit. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate Chinese part-of-speech tagging. In Proc. of ACL2012.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang and Kentaro Torisawa. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In Proc. of IJCNLP2011, pages 309-317.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. Computational Linguistics and Chinese Language Processing, 8 (1). pages 29-48.

Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In Proc. of EMNLP2010, pages 843-852.