Using Part-of-Speech Reranking to Improve Chinese Word Segmentation

Mengqiu Wang **Yanxin Shi**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{mengqiu,yanxins}@cs.cmu.edu

Abstract

Chinese word segmentation and Part-of-Speech (POS) tagging have been commonly considered as two separated tasks. In this paper, we present a system that performs Chinese word segmentation and POS tagging simultaneously. We train a segmenter and a tagger model separately based on linear-chain Conditional Random Fields (CRF), using lexical, morphological and semantic features. We propose an approximated joint decoding method by reranking the N-best segmenter output, based POS tagging information. Experimental results on SIGHAN Bakeoff dataset and Penn Chinese Treebank show that our reranking method significantly improve both segmentation and POS tagging accuracies.

1 Introduction

Word segmentation and Part-of-speeching (POS) tagging are the most fundamental tasks in Chinese natural language processing (NLP). Traditionally, these two tasks were treated as separate and independent processing steps chained together in a pipeline. In such pipeline systems, errors introduced at the early stage cannot be easily recovered in later steps, causing a cascade of errors and eventually harm overall performance. Intuitively, a correct segmentation of the input sentence is more likely to give rise to a correct POS tagging sequence than an incorrect segmentation. Hinging on this idea, one way to avoid error propagation in chaining subtasks such as segmentation and POS tagging is to exploit the *learning transfer* (Sutton and McCallum, 2005) among subtasks, typically through joint inference. Sutton et al. (2004) presented dynamic conditional random fields (DCRF), a generalization of the traditional linear-chain CRF that allow representation of interaction among labels. They used loopy belief propagation for inference approximation. Their empirical results on the joint task of POS tagging and NP-chunking suggested that DCRF gave superior performance over cascaded linear-chain CRF. Ng and Low (2004) and Luo (2003) also trained single joint models over the Chinese segmentation and POS tagging subtasks. In their work, they brought the two subtasks together by treating it as a single tagging problem, for which they trained a maximum entropy classifier to assign a combined word boundary and POS tag to each character.

A major challenge, however, exists in doing joint inference for complex and large-scale NLP application. Sutton and McCallum (Sutton and McCallum, 2005) suggested that in many cases exact inference can be too expensive and thus formidable. They presented an alternative approach in which a linear-chain CRF is trained separately for each subtask at training time, but at decoding time they combined the learned weights from the CRF cascade into a single grid-shaped factorial CRF to perform joint decoding and make predictions for all subtasks. Similar to (Sutton and McCallum, 2005), in our system we also train a cascade of linear-chain CRF for the subtasks. But at decoding time, we experiment with an alternative approximation method to joint decoding, by taking the n-best hypotheses from the segmentation model and use the POS tagging model for reranking. We evaluated our system on the open tracks of SIGHAN Bakeoff 2006 dataset. Furthermore, to evaluate our reranking method's impact on the POS tagging task, we also performed 10-fold cross-validation tests on the 250k Penn

Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pages 205–208,
Sydney, July 2006.

Chinese Treebank (CTB) (Xue et al., 2002). Results from both evaluations suggest that our simple reranking method is very effective. We achieved a consistent performance gain on both segmentation and POS tagging tasks over linearly-cascaded CRF. Our official F-scores on the 2006 Bakeoff open tracks are 0.935 (UPUC), 0.964 (CityU), 0.952 (MSRA) and 0.949 (CKIP).

2 Algorithm

Given an observed Chinese character sequence $\mathbf{X} = \{C_1, C_2, ..., C_n\}$, let **S** and **T** denote a segmentation sequence and a POS tagging sequence over **X**. Our goal is to find a segmentation sequence $\hat{\mathbf{S}}$ and a POS tagging sequence $\hat{\mathbf{T}}$ that maximize the posterior probability :

$$P(\mathbf{S}, \mathbf{T}|\mathbf{X} = \{C_1, C_2, ..., C_n\}) \tag{1}$$

Applying chain rule, we can further derive from Equation 1 the following:

$$\begin{aligned} & < \hat{\mathbf{S}}, \hat{\mathbf{T}} > \\ = \; & \arg\max_{\mathbf{S},\mathbf{T}} P(\mathbf{T}|\mathbf{S}, \mathbf{X} = \{C_1, C_2, ..., C_n\}) \\ & \times P(\mathbf{S}|\mathbf{X} = \{C_1, C_2, ..., C_n\}) \end{aligned} \tag{2}$$

Since we have factorized the joint probability in Equation 1 into two terms, we can now model these two components using conditional random fields (Lafferty et al., 2001). Linear-chain CRF models define conditional probability, $P(\mathbf{Z}|\mathbf{X})$, by linear-chain Markov random fields. In our case, **X** is the sequence of characters or words, and **Z** is the segmentation labels for characters (START or NON-START, used to indicate word boundaries) or the POS tagging for words (NN, VV, JJ, etc.). The conditional probability is defined as:

$$P(\mathbf{Z}|\mathbf{X}) = \frac{1}{N(X)} \exp\,(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{Z}, \mathbf{X}, t)) \tag{3}$$

where $N(X)$ is a normalization term to guarantee that the summation of the probability of all label sequences is one. $f_k(\mathbf{Z}, \mathbf{X}, t)$ is the k^{th} $local feature function$ at sequence position t. It maps a pair of **X** and **Z** and an index t to {0,1}. $(\lambda_1, ..., \lambda_K)$ is a weight vector to be learned from training set. A large positive value of λ_i means that the i^{th} feature function's value is frequent to be 1, whereas a negative value of λ_i means the i^{th} feature function's value is unlikely to be 1.

At decoding time, we are interested in finding the segmentation sequence $\hat{\mathbf{S}}$ and POS tagging sequence $\hat{\mathbf{T}}$ that maximizes the probability defined in Equation 2. Instead of exhaustively searching the whole space of all possible segmentations, we restrict our searching to $\mathcal{S} = \{\mathbf{S_1}, \mathbf{S_2}, ..., \mathbf{S_N}\}$, where $\mathcal{S}$ is the restricted search space consisting of N-best decoded segmentation sequences. This N-best list of segmentation sequences, $\mathcal{S}$, can be obtained using modified Viterbi algorithm and A* search (Schwartz and Chow, 1990).

3 Features

3.1 Features for Segmentation

We adopted the basic segmentation features used in (Ng and Low, 2004). These features are summarized in Table 1 ((1.1)-(1.7)). In these templates, C_0 refers to the current character, and C_{-n}, C_n refer to the characters *n* positions to the left and right of the current character, respectively. $Pu(C_0)$ indicates whether C_0 is a punctuation. $T(C_n)$ classifies the character C_n into four classes: numbers, dates (year, month, date), English letters and all other characters. $L_{Begin}(C_0)$, $L_{End}(C_0)$ and $L_{Mid}(C_0)$ represent the maximum length of words found in a lexicon[1] that contain the current character as either the first, last or middle character, respectively. $Single(C_0)$ indicates whether the current character can be found as a single word in the lexicon.

Besides the adopted basic features mentioned above, we also experimented with additional semantic features (Table 1 (1.8)). For (1.8), Sem_0 refers to the semantic class of current character, and Sem_{-1}, Sem_1 represent the semantic class of characters one position to the left and right of the current character, respectively. We obtained a character's semantic class from HowNet (Dong and Dong, 2006). Since many characters have multiple semantic classes defined by HowNet, it is a non-trivial task to choose among the different semantic classes. We performed contextual disambiguation of characters' semantic classes by calculating semantic class similarities. For example, let us assume the current character is 看(*look,read*) in a word context of 看报(*read*

[1] We compiled our lexicon from three external resources. HowNet: www.keenage.com; On-Line Chinese Tools: www.mandarintools.com; Online Dictionary from Peking University: http://ccl.pku.edu.cn/doubtfire/Course/Chinese%20Information%20Processing/Source_Code/Chapter_8/Lexicon_full_2000.zip

newspaper). The character 看(*look*) has two semantic classes in HowNet, i.e. 读(*read*) and 医治(*doctor*). To determine which class is more appropriate, we check the example words illustrating the meanings of the two semantic classes, given by HowNet. For 读(*read*), the example word is 看书(*read book*); for 医治(*doctor*), the example word is 看病(*see a doctor*). We then calculated the semantic class similarity scores between 报(*newspaper*) and 书(*book*), and 报(*newspaper*) and 病(*illness*), using HowNet's built-in similarity measure function. Since 报(*newspaper*) and 书(*book*) both have semantic class 文书(*document*), their maximum similarity score is 0.95, where the maximum similarity score between 报(*newspaper*) and 病(*illness*) is 0.03478. Therefore, Sem_0Sem_1 =读(*read*),文书(*document*). Similarly, we can figure out $Sem_{-1}Sem_0$. For Sem_0, we simply picked the top four semantic classes ranked by HowNet, and used ”‘NONE”’ for absent values.

Segmentation features
(1.1) $C_n, n \in [-2, 2]$
(1.2) $C_nC_{n+1}, n \in [-2, 1]$
(1.3) $C_{-1}C_1$
(1.4) $Pu(C_0)$
(1.5) $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$
(1.6) $L_{Begin}(C_0)$, $L_{End}(C_0)$
(1.7) $Single(C_0)$
(1.8) $Sem_0, Sem_nSem_{n+1}, n \in -1, 0$
POS tagging features
(2.1) $W_n, n \in [-2, 2]$
(2.2) $W_nW_{n+1}, n \in [-2, 1]$
(2.3) $W_{-1}W_1$
(2.4) $W_{n-1}W_nW_{n+1}, n \in [-1, 1]$
(2.5) $C_n(W_0), n \in [-2, 2]$
(2.6) $Len(W_0)$
(2.7) Other morphological features

Table 1: Feature templates list

3.2 Features for POS Tagging

The bottom half of Table 1 summarizes the feature templates we employed for POS tagging. W_0 denotes the current word. W_{-n} and W_n refer to the words *n* positions to the left and right of the current word, respectively. $C_n(W_0)$ is the n^{th} character in current word. If the number of characters in the word is less than 5, we use ”NONE” for absent characters. $Len(W_0)$ is the number of characters in the current word. We also used a group of binary features for each word, which are used to represent the morphological properties of current word, e.g. whether the current word is punctuation, number, foreign name, etc.

4 Experimental Results

We evaluated our system's segmentation results on the SIGHAN Bakeoff 2006 dataset. To evaluate our reranking method's impact on the POS tagging part, we also performed 10-fold cross-validation tests on the 250k Penn Chinese Treebank (CTB 250k). The CRF model for POS tagging is trained on CTB 250k in all the experiments. We report recall (R), precision (P), and F1-score (F) for both word segmentation and POS tagging tasks. N value is chosen to be 20 for the N-best list reranking, based on cross validation. For CRF learning and decoding, we use the CRF++ toolkit[2].

4.1 Results on Bakeoff 2006 Dataset

	R	***P***	***F***	***R_{oov}***	***R_{iv}***
UPUC	0.942	0.928	0.935	0.711	0.964
CityU	0.964	0.964	0.964	0.787	0.971
MSRA	0.949	0.954	0.952	0.692	0.958
CKIP	0.953	0.946	0.949	0.679	0.965

Table 2: Performance of our system on open tracks of SIGHAN Bakeoff 2006.

We participated in the open tracks of the SIGHAN Bakeoff 2006, and we achieved F-scores of 0.935 (UPUC), 0.964 (CityU), 0.952 (MSRA) and 0.949 (CKIP). More detailed performances statistics including in-vocabulary recall ($\boldsymbol{R_{iv}}$) and out-of-vocabulary recall ($\boldsymbol{R_{oov}}$) are shown in Table 2.

More interesting to us is how much the N-best list reranking method using POS tagging helped to increase segmentation performance. For comparison, we ran a linear-cascade of segmentation and POS tagging CRFs without reranking as the baseline system, and the results are shown in Table 3. We can see that our reranking method consistently improved segmentation scores. In particular, there is a greater improvement gained in recall than precision across all four tracks. We observed the greatest improvement from the UPUC track. We think it is because our POS tagging model is trained on CTB 250k, which could be drawn from the same corpus as the UPUC training data, and therefore there is a closer mapping between segmentation standard of the POS tagging training data and the segmentation training data (at this

[2] http://chasen.org/ taku/software/CRF++/

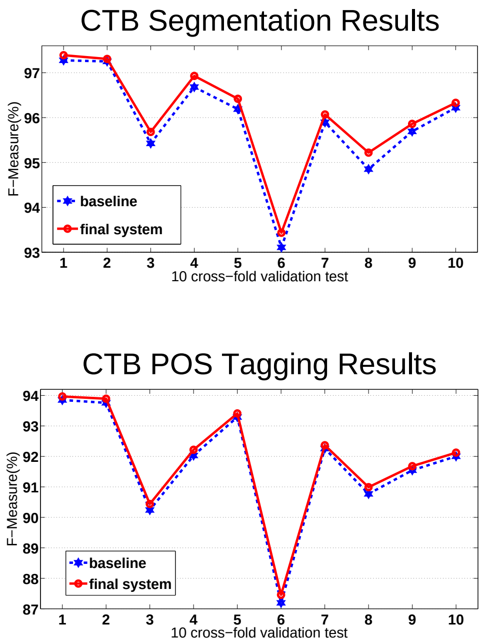

Figure 1: Segmentation and POS tagging results on CTB corpus.

point we are not sure if there exists any overlap between the UPUC test data and CTB 250k).

	Baseline system			Final system		
	R	***P***	***F***	***R***	***P***	***F***
UPUC	0.910	0.924	0.917	0.942	0.928	0.935
CityU	0.954	0.963	0.958	0.964	0.964	0.964
MSRA	0.935	0.953	0.944	0.949	0.954	0.952
CKIP	0.932	0.942	0.937	0.953	0.946	0.949

Table 3: Comparison of the baseline system (without POS reranking) and our final system.

4.2 Results on CTB Corpus

To evaluate our reranking method's impact on the POS tagging task, we also tested our systems on CTB 250k corpus using 10-fold cross-validation. Figure 1 summarizes the results of segmentation and POS tagging tasks on CTB 250k corpus. From figure 1 we can see that our reranking method improved both the segmentation and tagging accuracies across all 10 tests. We conducted pairwise t-tests and our reranking model was found to be statistically significantly better than the baseline model under significance level of 5.0^{-4} (p-value for segmentation) and 3.3^{-5} (p-value for POS tagging).

5 Conclusion

Our system uses conditional random fields for performing Chinese word segmentation and POS tagging tasks simultaneously. In particular, we proposed an approximated joint decoding method by reranking the N-best segmenter output, based POS tagging information. Our experimental results on both SIGHAN Bakeoff 2006 datasets and Chinese Penn Treebank showed that our reranking method consistently increased both segmentation and POS tagging accuracies. It is worth noting that our reranking method can be applied not only to Chinese segmentation and POS tagging tasks, but also to many other sequential tasks that can benefit from learning transfer, such as POS tagging and NP-chunking.

Acknowledgment

This work was supported in part by ARDA's AQUAINT Program.

References

Zhengdong Dong and Qiang Dong. 2006. *HowNet And The Computation Of Meaning*. World Scientific.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML '01*.

Xiaoqiang Luo. 2003. A maximum entropy Chinese character-based parser. In *Proceedings of EMNLP '03*.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of EMNLP '04*.

Richard Schwartz and Yen-Lu Chow. 1990. The n-best algorithm: An efficient and exact procedure for finding the n most likely sentence hypotheses. In *Proceedings of ICASSP '90*.

Charles Sutton and Andrew McCallum. 2005. Composition of conditional random fields for transfer learning. In *Proceedings of HLT/EMNLP '05*.

Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of ICML '04*.

Nianwen Xue, Fu-Dong Chiou, and Martha Stone Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Proceedings of COLING '02*.