Adaptive Chinese Word Segmentation with Online Passive-Aggressive Algorithm

Wenjun Gao
School of Computer Science
Fudan University
Shanghai, China
wjgao616@gmail.com

Xipeng Qiu
School of Computer Science
Fudan University
Shanghai, China
xpqiu@fudan.edu.cn

Xuanjing Huang
School of Computer Science
Fudan University
Shanghai, China
xjhuang@fudan.edu.cn

Abstract

In this paper, we describe our system[1] for CIPS-SIGHAN-2010 bake-off task of Chinese word segmentation, which focused on the cross-domain performance of Chinese word segmentation algorithms. We use the online passive-aggressive algorithm with domain invariant information for cross-domain Chinese word segmentation.

1 Introduction

In recent years, Chinese word segmentation (CWS) has undergone great development (Xue, 2003; Peng et al., 2004). The popular method is to regard word segmentation as a sequence labeling problems. The goal of sequence labeling is to assign labels to all elements of a sequence.

Due to the exponential size of the output space, sequence labeling problems tend to be more challenging than the conventional classification problems. Many algorithms have been proposed and the progress has been encouraging, such as SVM^{struct} (Tsochantaridis et al., 2004), conditional random fields (CRF) (Lafferty et al., 2001), maximum margin Markov networks (M3N) (Taskar et al., 2003) and so on. After years of intensive researches, Chinese word segmentation achieves a quite high precision. However, the performance of segmentation is not so satisfying for out-of-domain text.

There are two domains in domain adaption problem, a source domain and a target domain. When we use the machine learning methods for Chinese word segmentation, we assume that training and test data are drawn from the same distribution. This assumption underlies both theoretical analysis and experimental evaluations of learning algorithms. However, the assumption does not hold for domain adaptation(Ben-David et al., 2007; Blitzer et al., 2006). The challenge is the difference of distribution between the source and target domains.

In this paper, we use online margin maximization algorithm and domain invariant features for domain adaptive CWS. The online learning algorithm is Passive-Aggressive (PA) algorithm(Crammer et al., 2006), which passively accepts a solution whose loss is zero, while it aggressively forces the new prototype vector to stay as close as possible to the one previously learned.

The rest of the paper is organized as follows. Section 2 introduces the related works. Then we describe our algorithm in section 3 and 4. The feature templates are described in section 5. Section 6 gives the experimental analysis. Section 7 concludes the paper.

2 Related Works

There are several approaches to deal with the domain adaption problem.

The first approach is to use semi-supervised learning (Zhu, 2005).

The second approach is to incorporate supervised learning with domain invariant information.

The third approach is to improve the present model with a few labeled domain data.

Altun et al. (2006) investigated structured classification in a semi-supervised setting. They presented a discriminative approach that utilizes the

[1] Available at http://code.google.com/p/fudannlp/

intrinsic geometry of inputs revealed by unlabeled data points and we derive a maximum-margin formulation of semi-supervised learning for structured variables.

Self-training (Zhu, 2005) is also a popular technology. In self-training a classifier is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data. Typically the most confident unlabeled points, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure repeated. Note the classifier uses its own predictions to teach itself. Yarowsky (1995) uses self-training for word sense disambiguation, e.g. deciding whether the word plant means a living organism or a factory in a given context.

Zhao and Kit (2008) integrated unsupervised segmentation and CRF learning for Chinese word segmentation and named entity recognition. They found word accessory variance (Feng et al., 2004) is useful to CWS.

3 Online Passive-Aggressive Algorithm

Sequence labeling, the task of assigning labels $\mathbf{y} = y_1, \ldots, y_L$ to an input sequence $\mathbf{x} = x_1, \ldots, x_L$.

Give a sample $(\mathbf{x}, \mathbf{y})$, we define the feature is $\Phi(\mathbf{x}, \mathbf{y})$. Thus, we can label $\mathbf{x}$ with a score function,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{z}} F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{z})), \tag{1}$$

where $\mathbf{w}$ is the parameter of function $F(\cdot)$.

The score function of our algorithm is linear function.

Given an example $(\mathbf{x}, \mathbf{y})$, $\hat{\mathbf{y}}$ is denoted as the incorrect label with the highest score,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{z} \neq \mathbf{y}} w^T \Phi(\mathbf{x}, \mathbf{z}). \tag{2}$$

The **margin** $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$ is defined as

$$\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) - \mathbf{w}^T \Phi(\mathbf{x}, \hat{\mathbf{y}}). \tag{3}$$

Thus, we calculate the **hinge loss**.

$$\ell(\mathbf{w}; (\mathbf{x}, \mathbf{y}) = \begin{cases} 0, & \gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) > 1 \\ 1 - \gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})), & \text{otherwise} \end{cases} \tag{4}$$

We use the online PA learning algorithm to learn the weights of features. In round t, we find new weight vector $\mathbf{w}_{t+1}$ by

$$\begin{aligned} \mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{2} ||\mathbf{w} - \mathbf{w}_t||^2 + \mathcal{C} \cdot \xi, \\ \textbf{s.t. } \ell(\mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) <= \xi \textbf{ and } \xi >= 0 \end{aligned} \tag{5}$$

where C is a positive parameter which controls the influence of the slack term on the objective function.

The algorithms goal is to achieve a margin at least 1 as often as possible, thus the Hamming loss is also reduced indirectly. On rounds where the algorithm attains a margin less than 1 it suffers an instantaneous loss.

We abbreviate $\ell(\mathbf{w_t}; (x, y))$ to ℓ_t. If $\ell_t = 0$ then w_t itself satisfies the constraint in Eq. (5) and is clearly the optimal solution. We therefore concentrate on the case where $\ell_t > 0$.

First, we define the Lagrangian of the optimization problem in Eq. (5) to be

$$\begin{aligned} \mathcal{L}(\mathbf{w}, \xi, \alpha, \beta) = \frac{1}{2} ||\mathbf{w} - \mathbf{w}_t||^2 + \mathcal{C} \cdot \xi \\ + \alpha(\ell_t - \xi) - \beta \xi \\ \textbf{s.t. } \alpha >= 0, \beta >= 0. \end{aligned} \tag{6}$$

where α, β is a Lagrange multiplier.

Setting the partial derivatives of $\mathcal{L}$ with respect to the elements of ξ to zero gives

$$\alpha + \beta = \mathcal{C}. \tag{7}$$

The gradient of $\mathbf{w}$ should be zero,

$$\mathbf{w} - \mathbf{w}_t - \alpha(\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})) = 0, \tag{8}$$

we get

$$\mathbf{w} = \mathbf{w}_t + \alpha(\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})). \tag{9}$$

Substitute Eq. (7) and Eq. (9) to dual objective function Eq. (6), we get

$$\begin{aligned} \mathcal{L}(\alpha) = -\frac{1}{2} ||\alpha(\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}}))||^2 \\ - \alpha(\mathbf{w_t}^T(\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})) + \alpha \end{aligned} \tag{10}$$

Differentiate with α, and set it to zero, we get

$$\alpha||\Phi(\mathbf{x},\mathbf{y})-\Phi(\mathbf{x},\hat{\mathbf{y}})||^2 \\ +\mathbf{w_t}^T\left(\Phi(\mathbf{x},\mathbf{y})-\Phi(\mathbf{x},\hat{\mathbf{y}})\right)-1=0. \quad (11)$$

So,

$$\bar{\alpha} \quad = \quad \frac{1-\mathbf{w_t}^T\left(\Phi(\mathbf{x},\mathbf{y})-\Phi(\mathbf{x},\hat{\mathbf{y}})\right)}{||\Phi(\mathbf{x},\mathbf{y})-\Phi(\mathbf{x},\hat{\mathbf{y}})||^2}. \quad (12)$$

From $\alpha+\beta=\mathcal{C}$, we know that $\alpha<\mathcal{C}$, so

$$\bar{\alpha}^*=\min(\mathcal{C},\bar{\alpha}). \quad (13)$$

Finally, we get update strategy,

$$\mathbf{w_{t+1}}=\mathbf{w}_t+\bar{\alpha}^*(\Phi(\mathbf{x},\mathbf{y})-\Phi(\mathbf{x},\hat{\mathbf{y}})). \quad (14)$$

Our final algorithm is shown in Algorithm 1. In order to avoiding overfitting, the averaging technology is employed.

```
input  : training data set:
         $(\mathbf{x}_n,\mathbf{y}_n), n=1,\cdots,N$, and
         parameters: $\mathcal{C}, K$
output: $\mathbf{w}$
Initialize: $\mathbf{cw} \leftarrow 0$,;
for $k=0\cdots K-1$ do
    $\mathbf{w}_0 \leftarrow 0$ ;
    for $t=0\cdots T-1$ do
        receive an example $(\mathbf{x}_t,\mathbf{y}_t)$;
        predict:
        $\hat{\mathbf{y}}_t=\arg\max_{\mathbf{z}\neq\mathbf{y}_t}\langle\mathbf{w}_t,\Phi(\mathbf{x}_t,\mathbf{z})\rangle$;
        calculate $\ell(\mathbf{w};(\mathbf{x},\mathbf{y}))$;
        update $\mathbf{w}_{t+1}$ with Eq.(14);
    end
    $\mathbf{cw}=\mathbf{cw}+\mathbf{w}_T$ ;
end
$\mathbf{w}=\mathbf{cw}/K$ ;
```

Algorithm 1: Labelwise Margin Maximization Algorithm

4 Inference

The PA algorithm is used to learn the weights of features in training procedure. In inference procedure, we use Viterbi algorithm to calculate the maximum score label.

Let $\omega(n)$ be the best score of the partial label sequence ending with y_n. The idea of the Viterbi algorithm is to use dynamic programming to compute $\omega(n)$:

$$\omega(n)=\max_{n-1}\left(\omega(n-1)+\mathbf{w}^T\phi(x,y_n,y_{n-1})\right) \quad (15) \\ +\mathbf{w}^t\phi(x,y_n)$$

Using this recursive definition, we can evaluate $\omega(N)$ for all y_N, where N is the input length. This results in the identification of the best label sequence.

The computational cost of the Viterbi algorithm is $O(NL^2)$, where L is the number of labels.

5 Feature Templates

All feature templates used in this paper are shown in Table 1. C represents a Chinese character while the subscript of C indicates its position in the sentence relative to the current character, whose subscript is 0. T represents the character-based tag: “B”, “B2”, “B3”, “M”, “E” and “S”, which represent the beginning, second, third, middle, end or single character of a word respectively.

The type of character includes: digital, letter, punctuation and other.

We also use the word accessor variance for domain adaption. Word accessor variance (AV) was proposed by (Feng et al., 2004) and was used to evaluate how independently a string is used, and thus how likely it is that the string can be a word. The accessor variety of a string s of more than one character is defined as

$$AV(s)=\min\{L_{av}(s),R_{av}(s)\} \quad (16)$$

$L_{av}(s)$ is called the left accessor variety and is defined as the number of distinct characters (predecessors) except “S” that precede s plus the number of distinct sentences of which s appears at the beginning. Similarly, the right accessor variety $R_{av}(s)$ is defined as the number of distinct characters (successors) except “E” that succeed s plus the number of distinct sentences in which s appears at the end. The characters “S” and “E” are defined as the begin and end of a sentence. The word accessor variance was found effective for CWS with unsegmented text (Zhao and Kit, 2008).

Table 1: Feature templates

$C_i, T_0, (i = -1, 0, 1, 2)$
$C_i, C_{i+1}, T_0, (i = -2, -1, 0, 1)$
$T_{-1,0}$
T_c: Type of Character
AV: word accessor variance

6 CIPS-SIGHAN-2010 Bakeoff

CIPS-SIGHAN-2010 bake-off task of Chinese word segmentation focused on the cross-domain performance of Chinese word segmentation algorithms. There are two subtasks for this evaluation:

(1) Word Segmentation for Simplified Chinese Text;

(2) Word Segmentation for Traditional Chinese Text.

The test corpus of each subtask covers four domains: literature, computer science, medicine and finance.

We participate in closed training evaluation of both subtasks.

Firstly, we calculate the word accessor variance $AV_L(s)$of the continuous string s from labeled corpus. Here, we set the largest length of string s to be 4.

Secondly, we train our model with feature temples and $AV_L(s)$.

Thirdly, when we process the different domain unlabeled corpus, we recalculate the word accessory variance $AV_U(s)$ from the corresponding corpus.

Fourthly, we segment the domain corpus with new word accessory variance $AV_U(s)$ instead of $AV_L(s)$.

The results are shown in Table 2 and 3. The results show our method has a poor performance in OOV (Out-Of-Vocabulary) word.

The running environment is shown in Table 4.

Table 4: Experimental environment

OS	Win 2003
CPU	Intel Xeon 2.0G
Memory	4G

We set the max iterative number is 20. Our running time is shown in Table 5. “s” represents second, “chars” is the number of Chinese character, and “MB” is the megabyte. In practice, we found the system can achieve the same performance after 7 loops. Therefore, we just need less half the time in Table 5 actually.

7 Conclusion

In this paper, we describe our system in CIPS-SIGHAN-2010 bake-off task of Chinese word segmentation. Although our method just achieve a consequence of being average and not outstanding, it has an advantage of faster training than other batch learning algorithm, such as CRF and M3N.

In the future, we wish to improve our method in the following aspects. Firstly, we will investigate more effective domain invariant feature representation. Secondly, we will integrate our algorithm with self-training and other semi-supervised learning methods.

Acknowledgments

This work was (partially) funded by 863 Program (No. 2009AA01A346), 973 Program (No. 2010CB327906), and Shanghai Science and Technology Development Funds (No. 08511500302).

References

Altun, Y., D. McAllester, and M. Belkin. 2006. Maximum margin semi-supervised learning for structured variables. *Advances in neural information processing systems*, 18:33.

Ben-David, S., J. Blitzer, K. Crammer, and F. Pereira. 2007. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems*, 19:137.

Blitzer, J., R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128. Association for Computational Linguistics.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Table 2: Evaluation results on simplified corpus

		R	P	F1	OOV RR	IV RR
Literature	Best	0.945	0.946	0.946	0.816	0.954
	Our	0.915	0.925	0.92	0.577	0.94
Computer	Best	0.953	0.95	0.951	0.827	0.975
	Our	0.934	0.919	0.926	0.739	0.969
Medicine	Best	0.942	0.936	0.939	0.75	0.965
	Our	0.927	0.924	0.925	0.714	0.953
Finance	Best	0.959	0.96	0.959	0.827	0.972
	Our	0.94	0.942	0.941	0.719	0.961

Table 3: Evaluation results on traditional corpus

		R	P	F1	OOV RR	IV RR
Literature	Best	0.942	0.942	0.942	0.788	0.958
	Our	0.869	0.91	0.889	0.698	0.887
Computer	Best	0.948	0.957	0.952	0.666	0.977
	Our	0.933	0.949	0.941	0.791	0.948
Medicine	Best	0.953	0.957	0.955	0.798	0.966
	Our	0.908	0.932	0.92	0.771	0.919
Finance	Best	0.964	0.962	0.963	0.812	0.975
	Our	00.925	0.939	0.932	0.793	0.935

Table 5: Execution time of training and test phase.

Training	Task	A	B	C	D
	Simp	817.2s	795.6s	774.0s	792.0s
	Trad	903.6s	889.2s	885.6s	874.8s
Test		20327 chars/s, or 17.97 s/MB			

Feng, H., K. Chen, X. Deng, and W. Zheng. 2004. Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*.

Peng, F., F. Feng, and A. McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. *Proceedings of the 20th international conference on Computational Linguistics*.

Taskar, Ben, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *Proceedings of Neural Information Processing Systems*.

Tsochantaridis, I., T. Hofmann, T. Joachims, and Y Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning(ICML)*.

Xue, N. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Zhao, H. and C. Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *The Sixth SIGHAN Workshop on Chinese Language Processing*, pages 106–111. Citeseer.

Zhu, Xiaojin. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison. http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.