A Chinese Word Segmentation System Based on Structured Support Vector Machine Utilization of Unlabeled Text Corpus

Chongyang Zhang
Anhui Province Engineering Laboratory of Speech and Language, University of Science and Technology of China
cyzhang9@mail.ustc.edu.cn

Zhigang Chen
Anhui Province Engineering Laboratory of Speech and Language, University of Science and Technology of China
Chenzhigang@ustc.edu

Guoping Hu
Anhui Province Engineering Laboratory of Speech and Language, University of Science and Technology of China
Applecore@ustc.edu

Abstract

Character-based tagging method has achieved great success in Chinese Word Segmentation (CWS). This paper proposes a new approach to improve the CWS tagging accuracy by structured support vector machine (SVM) utilization of unlabeled text corpus. First, character N-grams in unlabeled text corpus are mapped into low-dimensional space by adopting SOM algorithm. Then new features extracted from these maps and another kind of feature based on entropy for each N-gram are integrated into the structured SVM methods for CWS. We took part in two tracks of the Word Segmentation for Simplified Chinese Text in bakeoff-2010: Closed track and Open track. The test corpora cover four domains: Literature, Computer Science, Medicine and Finance. Our system achieved good performance, especially in the open track on the domain of medicine, our system got the highest score among 18 systems.

1 Introduction

In the last decade, many statistics-based methods for automatic Chinese word segmentation (CWS) have been proposed with development of machine learning and statistical method (Huang and Zhao, 2007). Especially, character-based tagging method which was proposed by Nianwen Xue (2003) achieves great success in the second International Chinese word segmentation Bakeoff in 2005 (Low et al., 2005). The character-based tagging method formulates the CWS problem as a task of predicting a tag for each character in the sentence, i.e. every character is considered as one of four different types in 4-tag set: B (begin of word), M (middle of word), E (end of word), and S (single-character word).

Most of these works train tagging models only on limited labeled training sets, without using any unsupervised learning outcomes from unlabeled text. But in recent years, researchers begin to exploit the value of enormous unlabeled corpus for CWS, such as some statistics information on co-occurrence of sub-sequences in the whole text has been extracted from unlabeled data and been employed as input features for tagging model training (Zhao and Kit , 2007).

Word clustering is a common method to utilize unlabeled corpus in language processing research to enhance the generalization ability, such as part-of-speech clustering and semantic clustering (Lee et al., 1999 and B Wang and H Wang 2006). Character-based tagging method usually employs N-gram features, where an N-gram is an N-character segment of a string. We believe that there are also semantic or grammatical relationships between most of N-grams and these relationships will be useful in CWS. Intuitively, assuming the training data contains the bigram “色 / 列 ”(The last two

characters of the word “Israel” in Chinese), not contain the bigram “耳/其”(The last two characters of the word “Turkey” in Chinese), if we could cluster the two bigrams together according to unlabeled corpus and employ it as a feature for supervised training of tagging model, then maybe we will know that there should be a word boundary after “耳/其” though we only find the existence of word boundary after “色/列” in the training data. So we investigate how to apply clustering method onto unlabeled data for the purpose of improving CWS accuracy in this paper.

This paper proposes a novel method of using unlabeled data for CWS, which employs Self-Organizing Map (SOM) (Kohonen 1982) to organize Chinese character N-grams on a two-dimensional array, named as “N-gram cluster map” (NGCM), in which the character N-grams similar in grammatical structure and semantic meaning are organized in the same or adjacent position. Two different arrays are built based the N-gram’s preceding context and succeeding context respectively because normally N-gram is just part of Chinese word and doesn’t share similar preceding and succeeding context in the same time. Then NGCM-based features are extracted and applied to tagging model of CWS. Another kind of feature based on entropy for each N-gram is also introduced for improving the performance of CWS.

The rest of this paper is organized as follows: Section 2 describes our system; Section 3 describes structured SVM and the features which are obtained from labeled corpus and also unlabeled corpus; Section 4 shows experimental results on Bakeoff-2010 and Section 5 gives our conclusion.

2 System description

2.1 Open track:

The architecture of our system for open track is shown in Figure 1. For improving the cross-domain performance, we train and test with dictionary-based word segmentation outputs. On large-scale unlabeled corpus we use Self-Organizing Map (SOM) (Kohonen 1982) to organize Chinese character N-grams on a two-dimensional array, named as “N-gram cluster map” (NGCM), in which the character N-grams similar in grammatical structure and semantic meaning are organized in the same or adjacent position. Then new features are extracted from these maps and integrated into the structured SVM methods for CWS.

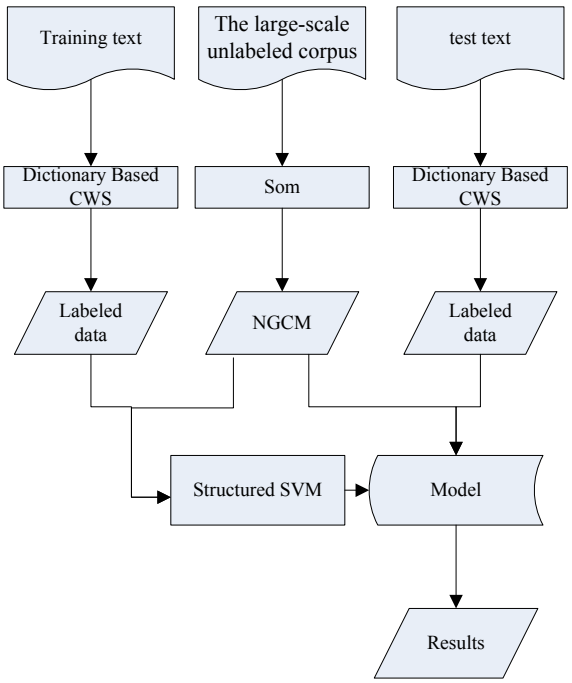

Figure 1: Open track system

2.2 Closed track:

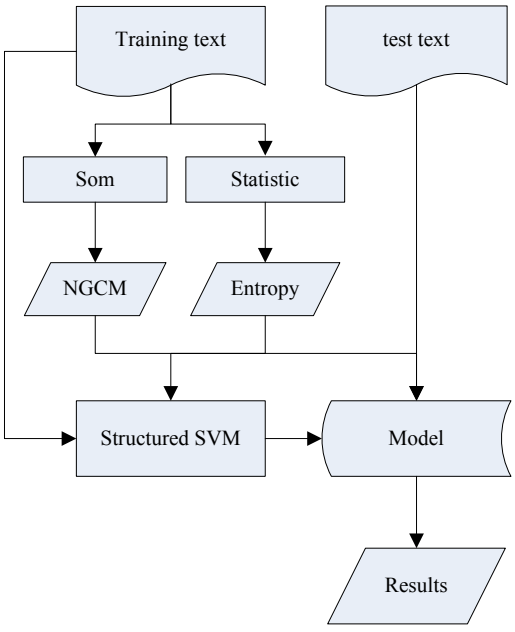

Figure 2: closed track system

Because the large-scale unlabeled corpus is forbidden to be used on closed track. We trained the SOM only on the data provided by

organizers. To make up for the deficiency of the sparse data on SOM, we add entropy-based features (ETF) for every N-gram to structured SVM model. The architecture of our system for close track is shown in Figure 2.

3 Learning algorithm

3.1 Structured support vector machine

The structured support vector machine can learn to predict structured y, such as trees sequences or sets, from x based on large-margin approach. We employ a structured SVM that can predict a sequence of labels $y=(y^1,...,y^T)$ for a given observation sequences $x=(x^1,...,x^T)$, where $y^t \in \Sigma$, Σ is the label set for y.

There are two types of features in the structured SVM: transition features (interactions between neighboring labels along the chain), emission features (interactions between attributes of the observation vectors and a specific label).we can represent the input-output pairs via joint feature map (JFM)

$$\psi(x,y)=\begin{pmatrix} \sum_{t=1}^{T}\phi(x^t)\otimes\Lambda^c(y^t) \\ \eta\sum_{t=1}^{T-1}\Lambda^c(y^t)\otimes\Lambda^c(y^{t+1}) \end{pmatrix}$$

where

$\Lambda^c(y)\equiv(\delta(y_1,y),\delta(y_2,y),...,\delta(y_K,y))'$

$\in\{0,1\}^K$, $y\in\{y_1,y_2,...,y_K\}=\Sigma$

Kronecker delta δ , $\delta_{i,j}=\begin{cases}1, i=j\\0, i\neq j\end{cases}$

$\phi(x)$ denotes an arbitrary feature representation of the inputs. The sign "$\otimes$" expresses tensor product defined as $\otimes : R^d\times R^k \to R^{dk}$, $[a\otimes b]_{i+(j-1)d}=[a]_i[b]_j$. T is the length of an observation sequence. $\eta\geq 0$ is a scaling factor which balances the two types of contributions.

Note that both transition features and emission features can be extended by including higher-order interdependencies of labels (e.g. $\Lambda^c(y^t)\otimes\Lambda^c(y^{t+1})\otimes\Lambda^c(y^{t+2})$),by including input features from a window centered at the current position (e.g. replacing $\phi(x^t)$ with $\phi(x^{t-r},...,x^t,...x^{t+r})$)or by combining higher-order output features with input features (e.g. $\sum_t\phi(x^t)\otimes\Lambda^c(y^t)\otimes\Lambda^c(y^{t+1})$)

The w-parametrized discriminant function $F: X\times Y\to R$ interpreted as measuring the compatibility of x and y is defined as:

$$F(x,y;w)=\langle w,\psi(x,y)\rangle$$

So we can maximize this function over the response variable to make a prediction

$$f(x)=\arg\max_{y\in Y}F(x,y,w)$$

Training the parameters can be formulated as the following optimization problem.

$$\min_{w,\xi}\frac{1}{2}\langle w,w\rangle+\frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$s.t.\forall i,\forall y\in Y:$$

$$\langle w,\psi_i(x_i,y_i)-\psi_i(x_i,y)\rangle\geq\Delta(y_i,y)-\xi_i$$

where n is the number of the training set, ξ_i is a slack variable , $C\geq 0$ is a constant controlling the tradeoff between training error minimization and margin maximization, $\Delta(y^1,y)$ is the loss function ,usually the number of misclassified tags in the sentence.

3.2 Features set for tagging model

For a training sample denoted as $x=(x^1,...,x^T)$ and $y=(y^1,...,y^T)$. We chose first-order interdependencies of labels to be transition features, and dependencies between labels and N-grams (n=1, 2, 3, 4) at current position in observed input sequence to be emission features.

So our JFM is the concatenation of the follow vectors

$$\sum_{t=1}^{T-1}\Lambda^c(y^t)\otimes\Lambda^c(y^{t+1})$$

$$\sum_{t=1}^{T}\phi(x^{t+m})\otimes\Lambda^c(y^t), m\in\{-1,0,1\}$$

$$\sum_{t=1}^{T}\phi(x^{t+m}x^{t+m+1})\otimes\Lambda^c(y^t), m\in\{-2,-1,0,1\}$$

$$\sum_{t=1}^{T}\phi(x^{t+m-1}x^{t+m}x^{t+m+1})\otimes\Lambda^c(y^t),$$

$$m\in\{-2,-1,0,1,2\}$$

$$\sum_{t=1}^{T} \phi(x^{t+m-1}x^{t+m}x^{t+m+1}x^{t+m+2}) \otimes \Lambda^{c}(y^{t}),$$
$$m \in \{-3,-2,-1,0,1,2\}$$

Figure 3 shows the transition features and the emission features of N-grams (n=1, 2) at y_3. The emission features of 3-grams and 4-grams are not shown here because of the large number of the dependencies.

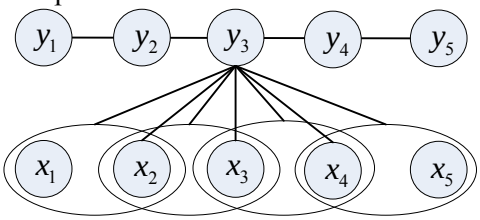

Figure 3: the transition features and the emission features at y_3 for structured SVM

3.3 SOM-based N-gram cluster maps and the NGCM *mapping* feature

The Self-Organizing Map (SOM) (Kohonen 1982), sometimes called Kohonen map, was developed by Teuvo Kohonen in the early 1980s.

Self-organizing semantic maps (Ritter and Kohonen 1989, 1990) are SOMs that have been organized according to word similarities, measured by the similarity of the short contexts of the words. Our algorithm of building N-gram cluster maps is similar to self-organizing semantic maps. Because normally N-gram is just part of Chinese word and do not share similar preceding and succeeding context in the same time, so we build two different maps according to the preceding context and the succeeding context of N-gram individually. In the end we build two NGCMs: NGCMP (NGCM according to preceding context) and NGCMS (NGCM according to succeeding context).

Due to the limitation of our computer and time we only get two 15×15 size 2GCMs for open track system from large-scale unlabeled corpus which was obtained easily from websites like Sohu, Netease, Sina and People Daily.

The 2GCMP and 2GCMS we got for the open track task are shown in Figure 4 and Figure 5 respectively.

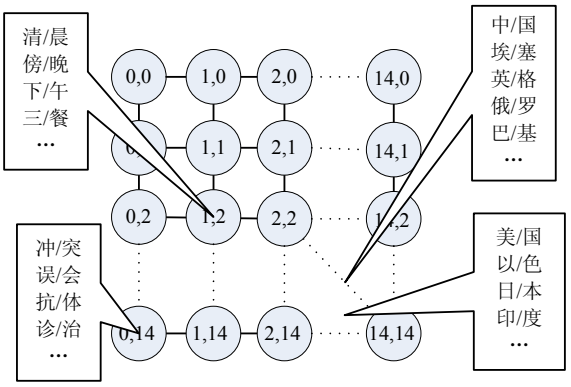

Figure 4: 2GCMP

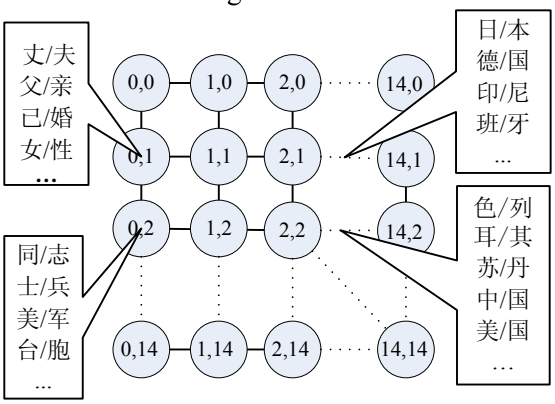

Figure 5: 2GCMS

After checking the results, we find that the 2GCMS have following characters:1) most of the meaningless bigrams that contain characters from more than one word, such as the bigram "京天" in "...北京天坛..." , are organized into the same neurons in the map, 2) most of the first or last bigrams of the country names are organized into a few adjacent neurons, such as “色/列”, “耳/其”, “中/国” and “美/国”in 2GCMS , “巴/基”, “埃/塞”, “英/格”, “俄/罗” , and “中/国” in 2GCMP.

Two 20×1 size 2GCMs are trained for the closed track system only on the data provided by organizers. The results are not as good as the results of the 15×15 size 2GCMs because of the less training data. The second character described above is no longer apparent as well as the 15×15 size 2GCMs, but it still kept the first character.

Then we adopt the position of the neurons which current N-gram mapped in the NGCM as a new feature. So every feature has D dimensions (D equals to the dimension of the NGCM, every dimension is corresponding to the coordinate value in the NGCM). In this way, N-gram which is originally represented as a

high dimensional vector based on its context is mapped into a very low-dimensional space. We call it NGCM mapping feature. So our previous JFM in section 3.2 is concatenated with the following features:

$$\sum_{t=1}^{T} \varphi^{2\text{GCMS}}(x^{t+m}x^{t+m+1}) \otimes \Lambda^c(y^t), m \in \{-2,-1\}$$

$$\sum_{t=1}^{T} \varphi^{2\text{GCMP}}(x^{t+m}x^{t+m+1}) \otimes \Lambda^c(y^t), m \in \{0,1\}$$

$$\sum_{t=1}^{T} \eta^{2\text{GCMS}}(x^{t+m}x^{t+m+1}) \otimes \Lambda^c(y^t), m \in \{-2,-1\}$$

$$\sum_{t=1}^{T} \eta^{2\text{GCMP}}(x^{t+m}x^{t+m+1}) \otimes \Lambda^c(y^t), m \in \{0,1\}$$

where $\varphi^{2\text{GCMS}}(x)$ and $\varphi^{2\text{GCMP}}(x) \in \{0,1,...,14\}^2$ denote the NGCM mapping feature from 2GCMS and 2GCMP respectively. $\eta^{\text{NGCM}}(x)$ denotes the quantization error of current N-gram x on its NGCM.

As an example, the process of import features from NGCMs at y_3 is presented in Figure 6.

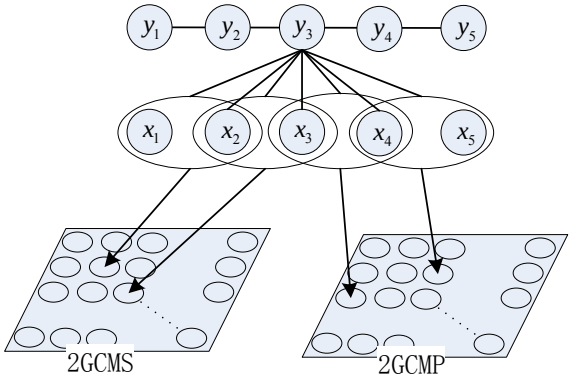

Figure 6: Using 2GCMS and 2GCMP as input to structured SVM

3.4 Entropy-based features

On closed track, the entropy of the preceding and succeeding characters conditional on the N-gram and also the self-information of the N-gram are used as features for the structured SVM methods. Then our previous JFM in section 3.2 is concatenated with the following features:

$$\sum_{t=1}^{T} H(P \mid N = x^{Ngram}) \otimes \Lambda^c(y^t),$$

$$x^{Ngram} \in \{x^{t+1}x^{t+2}, x^{t+1}x^{t+2}x^{t+3}, x^{t+1}x^{t+2}x^{t+3}x^{t+4}\}$$

$$\sum_{t=1}^{T} H(S \mid N = x^{Ngram}) \otimes \Lambda^c(y^t),$$

$$x^{Ngram} \in \{x^{t-2}x^{t-1}, x^{t-3}x^{t-2}x^{t-1}, x^{t-4}x^{t-3}x^{t-2}x^{t-1}\}$$

$$\sum_{t=1}^{T} I(N = x^{Ngram}) \otimes \Lambda^c(y^t)$$

$x^{Ngram} \in$ all the ngrams used in section 3.2

Where P and S denote the set of the preceding and succeeding characters respectively. The entropy: $H(X \mid N = x^{Ngram}) = -\sum_{X \in x^t} p(x^t \mid x^{Ngram}) \log p(x^t \mid x^{Ngram})$

The self-information of the N-gram $N = x^{Ngram}$: $I(x^{Ngram}) = -\log p(x^{Ngram})$

4 Applications and Experiments

4.1 Text Preprocessing

Text is usually mixed up with numerical or alphabetic characters in Chinese natural language, such as “我在 office 上班到晚上 9 点”. These numerical or alphabetic characters are barely segmented in CWS. Hence, we treat these symbols as a whole “character” according to the following two preprocessing steps. First replace one alphabetic character to four continuous alphabetic characters with E1 to E4 respectively, five or more alphabetic characters with E5. Then replace one numerical number to four numerical numbers with N1 to N4 and five or more numerical numbers with N5. After text preprocessing, the above examples will be “我在 E5 上班到晚上 N1 点”.

4.2 Character-based tagging method for CWS

Previous works show that 6-tag set achieved a better CWS performance (Zhao et al., 2006). Thus, we opt for this tag set. This 6-tag set adds ‘B2’ and ’B3’ to 4-tag set which stand for the type of the second and the third character in a Chinese word respectively. For example, the tag sequence for the sentence “上海世博会/将/持续/半

年(Shanghai World Expo / will / last / six months)" will be "B B2 B3 M E S B E B E".

4.3 Results in the bakeoff-2010

We use *svm*^{hmm} version 3.1 to build our structured SVM models. The cut-off threshold is set to 2. The precision parameter is set to 0.1. The tradeoff between training error minimization and margin maximization is set to 1000.

We took part in two tracks of the Word Segmentation for Simplified Chinese Text in bakeoff-2010: c (Closed track), o (Open track). The test corpora cover four domains: A (Literature), B (Computer Science), C (Medicine), D (Finance).

Precision(P),Recall(R),F-measure(F),Out-Of-Vocabulary Word Recall(OOV RR) and In-Vocabulary Word Recall(IV RR) are adopted to measure the performance of word segmentation system.

Table 1 shows the results of our system on the word segmentation task for simplified Chinese text in bakeoff-2010. Table 2 shows the comparision between our system results and best results in bakeoff-2010.

		R	P	F1	OOV RR	IV RR
A	c	0.932	0.935	0.933	0.654	0.953
	o	0.942	0.943	0.942	0.702	0.959
B	c	0.935	0.934	0.935	0.792	0.961
	o	0.948	0.946	0.947	0.812	0.973
C	c	0.937	0.934	0.936	0.761	0.959
	o	0.941	0.935	0.938	0.787	0.96
D	c	0.955	0.956	0.955	0.848	0.965
	o	0.948	0.955	0.951	0.853	0.957

Table 1: The results of our systems

		F1(Bakeoff-2010)	F1(Our system)
A	c	0.946	0.933
	o	0.955	0.942
B	c	0.951	0.935
	o	0.95	0.947
C	c	0.939	0.936
	o	0.938	0.938
D	c	0.959	0.955
	o	0.96	0.951

Tabel 2: The comparision between our system results and best results in bakeoff-2010

It is obvious that our systems are stable and reliable even in the domain of medicine when the F-measure of the best results was decreased. Our open track system performs better than closed track system, demonstrating the benefit of the dictionary-based word segmentation outputs and the NGCMs which are training on large-scale unlabeled corpus.

5 Conclusion

This paper proposes a new approach to improve the CWS tagging accuracy by structured support vector machine (SVM) utilization of unlabeled text corpus. We use SOM to organize Chinese character N-grams on a two-dimensional array, so that the N-grams similar in grammatical structure and semantic meaning are organized in the same or adjacent position. Then new features extracted from these maps and another kind of feature based on entropy for each N-gram are integrated into the structured SVM methods for CWS. Our system achieved good performance, especially in the open track on the domain of medicine, our system got the highest score among 18 systems.

In future work, we will try to organizing all the N-grams on a much larger array, so that every neuron will be labeled by a single N-gram. The ultimate objective is to reduce the dimension of input features for supervised CWS learning by replacing N-gram features with two-dimensional NGCM mapping features.

References

B.Wang, H.Wang 2006.A Comparative Study on Chinese Word Clustering. *Computer Processing of Oriental Languages.* Beyond the Orient: The Research Challenges Ahead, pages 157-164

Chang-Ning Huang and Hai Zhao. 2007. Chinese word segmentation: A decade review. Journal of Chinese Information Processing, 21(3):8–20.

Chung-Hong Lee & Hsin-Chang Yang.1999, *A Web Text Mining Approach Based on Self-Organizing Map*, ACM-library

G.Bakir, T.Hofmann, B.Scholkopf, A.Smola, B. Taskar, and S. V. N. Vishwanathan, editors. 2007 *Predicting Structured Data*. MIT Press, Cambridge, Massachusetts.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. *Effective tag set selection*

inChinese word segmentation via conditional random field modeling. In Proceedings of PACLIC-20, pages 87–94. Wuhan, China.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006.*An improved Chinese word segmentation system with conditional random field*. In SIGHAN-5, pages 162–165, Sydney, Australia, July 22-23.

Hai Zhao and Chunyu Kit. 2007. *Incorporating global information into supervised learning for Chinese word segmentation*. In PACLING-2007, pages 66–74, Melbourne,Australia, September 19-21.

H.Ritter, and T.Kohonen, 1989. *Self-organizing semantic maps*. Biological Cybernetics, vol. 61, no. 4, pp. 241-254.

I.Tsochantaridis,T.Joachims,T.Hofmann,and Y.Altun. 2005. *Large Margin Methods for Structured and Interdependent Output Variables*, Journal of Machine Learning Research (JMLR), 6(Sep):1453-1484.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo.2005. *A maximum entropy approach to Chinese word segmentation*. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 161–164. Jeju Island,Korea.

J.Lafferty,A.McCallum, F.Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In Proceedings of the International Conference on Machine Learning (ICML). San Francisco: Morgan Kaufmann Publishers, 282−289.

Nianwen Xue and Susan P. Converse., 2002, *Combining Classifiers for Chinese Word Segmentation*, In Proceedings of First SIGHAN Workshop on Chinese Language Processing.

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

R.Sproat and T.Emerson. 2003.*The first international Chinese word segmentation bakeoff*. In The Second SIGHAN Workshop on Chinese Language Processing, pages 133–143.Sapporo, Japan.

S.Haykin, 1994. Neural Networks: *A Comprehensive Foundation*. NewYork: MacMillan.

T.Joachims, T.Finley, Chun-Nam Yu. 2009, *Cutting-Plane Training of Structural SVMs*, Machine Learning Journal,77(1):27-59.

T.Joachims. 2008 . svm^{hmm} *Sequence Tagging with Structural Support Vector Machines*, http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html

T.Honkela, 1997. *Self-Organizing Maps in Natural Language Processing. PhD thesis, Helsinki University of Technology*, Department of Computer Science and Engineering, Laboratory of Computer and Information Science.

T.Kohonen. 1982.*Self-organized formation of topologically correct feature maps*. Biological Cybernetics, 43, pp. 59-69.

T.Kohonen., J.Hynninen, J.Kangas, J.Laaksonen, 1996 ,*SOM_PAK: The Self-Organizing Map Program Package*,Technical Report A31, Helsinki University of Technology , http://www.cis.hut.fi/nnrc/nnrc-programs.html

Y.Altun, I.Tsochantaridis, T.Hofmann. 2003. *Hidden Markov Support Vector Machines*. In Proceedings of International Conference on Machine Learning (ICML).