A Multi-layer Chinese Word Segmentation System Optimized for Out-of-domain Tasks

Qin Gao
Language Technologies Institute
Carnegie Mellon University
qing@cs.cmu.edu

Stephan Vogel
Language Technologies Institute
Carnegie Mellon University
stephan.vogel@cs.cmu.edu

Abstract

State-of-the-art Chinese word segmentation systems have achieved high performance when training data and testing data are from the same domain. However, they suffer from the generalizability problem when applied on test data from different domains. We introduce a multi-layer Chinese word segmentation system which can integrate the outputs from multiple heterogeneous segmentation systems. By training a second layer of large margin classifier on top of the outputs from several Conditional Random Fields classifiers, it can utilize a small amount of in-domain training data to improve the performance. Experimental results show consistent improvement on F1 scores and OOV recall rates by applying the approach.

1 Introduction

The Chinese word segmentation problem has been intensively investigated in the past two decades. From lexicon-based methods such as Bi-Directed Maximum Match (BDMM) (Chen et al., 2005) to statistical models such as Hidden Markove Model (HMM) (Zhang et al., 2003), a broad spectrum of approaches have been experimented. By casting the problem as a character labeling task, sequence labeling models such as Conditional Random Fields can be applied on the problem (Xue and Shen, 2003). State-of-the-art CRF-based systems have achieved good performance. However, like many machine learning problems, generalizability is crucial for a domain-independent segmentation system. Because the training data usually come from limited domains, when the domain of test data is different from the training data, the results are still not satisfactory.

A straight-forward solution is to obtain more labeled data in the domain we want to test. However this is not easily achievable because the amount of data needed to train a segmentation system are large. In this paper, we focus on improving the system performance by using a relatively small amount of manually labeled in-domain data together with larger out-of-domain corpus[1]. The effect of mingling the small in-domain data into large out-of-domain data may be neglectable due to the difference in data size. Hence, we try to explore an alternative way that put a second layer of classifier on top of the segmentation systems built on out-of-domain corpus (we will call them sub-systems). The classifier should be able to utilize the information from the sub-systems and optimize the performance with a small amount of in-domain data.

The basic idea of our method is to integrate a number of different sub-systems *whose performance varies on the new domain.* Figure 1 demonstrates the system architecture. There are two layers in the system. In the lower layer, the out-of-domain corpora are used, together with other resources to produce heterogeneous sub-systems. In the second layer the outputs of the sub-systems in the first layer are treated as input to the classifier. We train the classifier with small in-domain data. All the sub-systems should have

[1] From this point, we use the term *out-of-domain corpus* to refer to the general and large training data that are not related to the test domain, and the term *in-domain corpus* to refer to small amount of data that comes from the *same* domain of the test data

reasonable performance on all domains, but their performance on different domains may vary. The job of the second layer is to find the best decision boundary on the target domain, in presence of all the decisions made by the sub-systems.

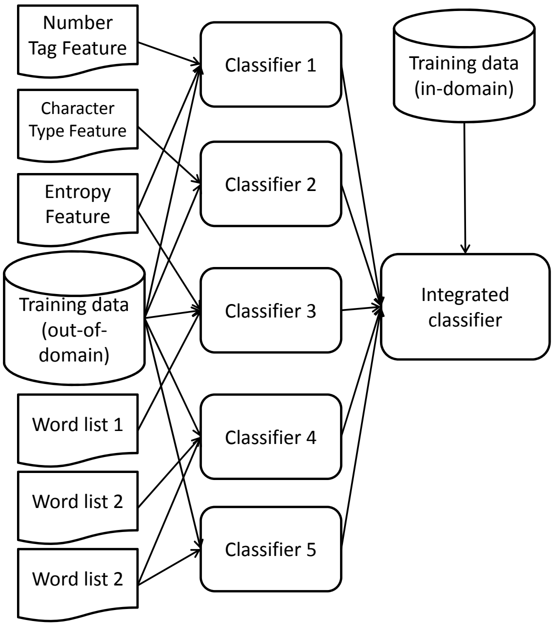

Figure 1: The architecture of the system, the first layer (sub-systems) is trained on general out-of-domain corpus and various resources, while the second layer of the classifier is trained on in-domain corpus.

Conditional Random Fields (CRF) (Lafferty et al., 2001) has been applied on Chinese word segmentation and achieved high performance. However, because of its conditional nature the small amount of in-domain corpus will not significantly change the distributions of the model parameters trained on out-of-domain corpus, it is more suitable to be used in the sub-systems than in the second-layer classifier. Large margin models such as Support Vector Machine (SVM) (Vapnik, 1995) can be trained on small corpus and generalize well. Therefore we chose to use CRF in building sub-systems and SVM in building the second-layer. We built multiple CRF-based Chinese word segmentation systems using different features, and then use the marginal probability of each tag of all the systems as features in SVM. The SVM is then trained on small in-domain corpus, results in a decision hyperplane that minimizes the loss in the small training data. To integrate the dependencies of output tags, we use SVM-HMM (Altun et al., 2003) to capture the interactions between tags and features. By applying SVM-HMM we can bias our decision towards most informative CRF-based system w.r.t. the target domain. Our methodology is similar to (Cohen and Carvalho, 2005), who applied a cross-validation-like method to train sequential stacking models, while we directly use small amount of in-domain data to train the second-layer classifiers.

The paper is organized as follows, first we will discuss the CRF-based sub-systems we used in section 2, and then the SVM-based system combination method in section 3. Finally, in section 4 the experimental results are presented.

2 CRF-based sub-systems

In this section we describe the sub-systems we used in system. All of the sub-systems are based on CRF with different features. The tag set we use is the 6-tag (B1, B2, B3, M, E, S) set proposed by Zhao et al (2006). All of the sub-systems use the same tag set, however as we will see later, the second-layer classifier in our system does not require the sub-systems to have a common tag set. Also, all of the sub-systems include a common set of character features proposed in (Zhao and Kit, 2008). The offsets and concatenations of the six n-gram features (the feature template) are: $C_{-1}, C_0, C_1, C_{-1}C_0, C_0C_1, C_{-1}C_1$. In the remaining part of the section we will introduce other features that we employed in different sub-systems.

2.1 Character type features

By simply classify the characters into four types: Punctuation (P), Digits (D), Roman Letters (L) and Chinese characters (C), we can assign character type tags to every character. The idea is straight-forward. We denote the feature as CTF.

Similar to character feature, we also use different offsets and concatenations for character type features. The feature template is identical to character feature, i.e. CTF_{-1}, CTF_0, CTF_1, $CTF_{-1}CTF_0$, CTF_0CTF_1, $CTF_{-1}CTF_1$ are used as features in CRF training.

2.2 Number tag feature

Numbers take a large portion of the OOV words, which can easily be detected by regular expressions or Finite State Automata. However there are often ambiguities on the boundary of numbers. Therefore, instead of using detected numbers as final answers, we use them as features. The number detector we developed finds the longest substrings in a sentence that are:

- Chinese Numbers (N)
- Chinese Ordinals (O)
- Chinese Dates (D)

For each character of the detected numbers/ordinal/date, we assign a tag that reflects the position of the character in the detected number/ordinal/date. We adopt the four-tag set (B, M, E, S). The position tags are appended to end of the number/ordinal/date tags to form the number tag feature of that character. I.e. there are totally 13 possible values for the number tag feature, as listed in Table 1.[2]

	Number	Ordinal	Date	Other
Begin	NB	OB	DB	XX
Middle	NM	OM	DM	
End	NE	OE	DE	
Single	NS	OS^*	DS^*	

Table 1: The feature values used in the number tag feature, note that OS and DS are never observed because there is no single character ordinal/date by our definition.

Similar to character feature and character type feature, the feature template mention before is also applied on the number tag feature. We denote the number tag features as NF.

2.3 Conditional Entropy Feature

We define the *Forward Conditional Entropy* of a character C by the entropy of all the characters that follow C in a given corpus, and the *Backward Conditional Entropy* as the entropy of all the characters that precede C in a given corpus. The conditional entropy can be computed easily from a character bigram list generated from the corpus. Assume we have a bigram list $B = \{B_1, B_2, \cdots, B_N\}$, where every bigram entry $B_k = \{c_{i_k}, c_{j_k}, n_k\}$ is a triplet of the two consecutive characters c_{i_k} and c_{j_k} and the count of the bigram in the corpus, n_k. The Forward Conditional Entropy of the character C is defined by:

$$H_f(C) := \sum_{c_{i_k}=C} \frac{n_k}{Z} \log \frac{n_k}{Z}$$

where $Z = \sum_{c_{i_k}=C} n_k$ is the normalization factor.

And the Backward Conditional Entropy can be computed similarly.

We assign labels to every character based on the conditional entropy of it. If the conditional entropy value is less than 1.0, we assign feature value 0 to the character, and for region $[1.0, 2.0)$, we assign feature value 1. Similarly we define the region-to-value mappings as follows: $[2.0, 3.5) \rightarrow 2$, $[3.5, 5.0) \rightarrow 4$, $[5.0, 7.0) \rightarrow 5$, $[7.0, +\infty) \rightarrow 6$. The forward and backward conditional entropy forms two features. We will refer to these features as EF.

2.4 Lexical Features

Lexical features are the most important features to make sub-systems output different results on different domains. We adopt the definition of the features partially from (Shi and Wang, 2007). In our system we use only the $L_{begin}(C_0)$ and $L_{end}(C_0)$ features, omitting the $L_{mid}C_0$ feature. The two features represent the maximum length of words found in the lexicon that contain the current character as the first or last character, correspondingly. For feature values equal or greater than 6, we group them into one value.

Although we can find a number of Chinese lexicons available, they may or may not be generated according to the same standard as the training data. Concatenating them into one may bring in noise and undermine the performance. Therefore, every lexicon will generate its own lexical features.

3 SVM-based System Combination

Generalization is a fundamental problem of Chinese word segmentation. Since the training data may come from different domains than the test data, the vocabulary and the distribution can also

[2] Two of the tags, OS and DS are never observed.

be different. Ideally, if we can have labeled data from the same domain, we can train segmenters specific to the domain. However obtaining sufficient amount of labeled data in the target domain is time-consuming and expensive. In the mean time, if we only label a small amount of data in the target domain and put them into the training data, the effect may be too small because the size of out-of-domain data can overwhelm the in-domain data.

In this paper we propose a different way of utilizing small amount of in-domain corpus. We put a second-layer classifier on top of the CRF-based sub-systems, the output of CRF-based sub-systems are treated as features in an SVM-HMM (Altun et al., 2003) classifier. We can train the SVM-HMM classifier on a small amount of in-domain data. The training procedure can be viewed as finding the optimal decision boundary that minimize the hinge loss on the in-domain data. Because the number of features for SVM-HMM is significantly smaller than CRF, we can train the model with as few as several hundred sentences.

Similar to CRF, the SVM-HMM classifier still treats the Chinese word segmentation problem as character tagging. However, because of the limitation of training data size, we try to minimize the number of classes. We chose to adopt the two-tag set, i.e. class 1 indicates the character is the end of a word and class 2 means otherwise. Also, due to limited amount of training data, we do not use any character features, instead, the features comes directly from the output of sub-systems. The SVM-HMM can use any real value features, which enables integration of a wide range of segmenters. In this paper we use only the CRF-based segmenters, and the features are the marginal probabilities (Sutton and McCallum, 2006) of all the tags in the tag set for each character. As an example, for a CRF-based sub-system that outputs six tags, it will output six features for each character for the SVM-HMM classifier, corresponding to the marginal probability of the character given the CRF model. The marginal probabilities for the same tag (e.g. B1, S, etc) come from different CRF-based sub-systems are treated as distinct features.

	Features	Lexicons
S1	CF, CTF	None
S2	CF, NF	ADSO, CTB6
S3	CF, CTF, NF	ADSO
S4	CF, CTF, NF, EF	ADSO, CTB6
S5	CF, EF	None
S6	CF, NF	None
S7	CF, CTF	ADSO
S8	CF, CTF	CTB6

Table 2: The configurations of CRF-based sub-systems. S1 to S4 are used in the final submission of the Bake-off, S5 through S8 are also presented to show the effects of individual features.

When we encounter data from a new domain, we first use one of the CRF-based sub-system to segment a portion of the data, and manually correct obvious segmentation errors. The manually labeled data are then processed by all the CRF-based sub-systems, so as to obtain features of every character. After that, we train the SVM-HMM model using these features.

During decoding, the Chinese input will also be processed by all of the CRF-based sub-systems, and the outputs will be fed into the SVM-HMM classifier. The final decisions of word boundaries are based solely on the classified labels of SVM-HMM model.

For the Bake-off system, we labeled two hundred sentences in each of the unsegmented training set (A and B). Since only one submission is allowed, the SVM-HMM model of the final system was trained on the concatenation of the two training sets, i.e. four hundred sentences.

The CRF-based sub-systems are trained using CRF++ toolkit (Kudo, 2003), and the SVM-HMM trained by the SVM^{struct} toolkit (Joachims et al., 2009).

4 Experiments

To evaluate the effectiveness of the proposed system combination method, we performed two experiments. First, we evaluate the system combination method on provided training data in the way that is similar to cross-validation. Second, we experimented with training the SVM-HMM model with the manually labeled data come from cor-

	Micro-Average				Macro-Average			
	P	R	F1	OOV-R	P	R	F1	OOV-R
S1	0.962	0.960	0.961	0.722	0.962	0.960	0.960	0.720
S2	0.965	0.966	0.966	0.725	0.965	0.966	0.966	0.723
S3	0.966	0.967	0.967	0.731	0.966	0.967	0.967	0.729
S4	0.968	0.969	0.968	0.731	0.967	0.969	**0.969**	0.729
S5	0.962	0.960	0.961	0.720	0.962	0.960	0.960	0.718
S6	0.963	0.961	0.962	0.730	0.963	0.961	0.961	0.729
S7	0.966	0.967	0.966	0.723	0.966	0.967	0.967	0.720
S8	0.963	0.960	0.962	0.727	0.963	0.960	0.960	0.726
CB	0.969	0.969	**0.969**	**0.741**	0.969	0.969	**0.969**	**0.739**

Table 3: The performance of individual sub-systems and combined system. The Micro-Average results come from concatenating all the outputs of the ten-fold systems and then compute the scores, and the Macro-Average results are calculated by first compute the scores in every of the ten-fold systems and then average the scores.

	Set A				Set B			
	P	R	F1	OOV-R	P	R	F1	OOV-R
S1	0.925	0.920	0.923	0.625	0.936	0.938	0.937	0.805
S2	0.934	0.934	0.934	0.641	0.941	0.930	0.935	0.751
S3	0.940	0.937	0.938	0.677	0.938	0.926	0.932	0.752
S4	0.942	0.940	0.941	0.688	0.944	0.929	0.936	0.776
CB1	0.943	0.941	0.942	0.688	0.948	0.936	0.942	0.794
CB2	0.941	0.940	0.941	0.692	0.939	0.949	0.944	**0.821**
CB3	0.943	0.939	**0.941**	**0.699**	0.950	0.950	**0.950**	0.820

Table 4: The performance of individual systems and system combination on Bake-off test data, CB1, CB2, and CB3 are system combination trained on labeled data from domain A, B, and the concatenation of the data from both domains.

responding domains, and tested the resulting systems on the Bake-off test data.

For experiment 1, We divide the training set into 11 segments, segment 0 through 9 contains 1733 sentences, and segment 10 has 1724 sentence. We perform 10-fold cross-validation on segment 0 to 9. Every time we pick one segment from segment 0 to 9 as test set and the remaining 9 segments are used to train CRF-based subsystems. Segment 10 is used as the training set for SVM-HMM model. The sub-systems we used is listed in Table 2.

In Table 3 we provide the micro-level and macro-level average of performance the ten-fold evaluation, including both the combined system and all the individual sub-systems. Because the system combination uses more data than its sub-systems (segment 10), in order to have a fair comparison, when evaluating individual subsystems, segment 10 is appended to the training data of CRF model. Therefore, the individual subsystems and system combination have exactly the same set of training data.

As we can see in the results in Table 3, the system combination method (Row CB) has improvement over the best sub-system (S4) on both F1 and OOV recall rate, and the OOV recall rate improved by 1%. We should notice that in this experiment we actually did not deal with any data from different domains, the advantage of the proposed method is therefore not prominent.

We continue to present the experiment results of the second experiment. In the experiment we labeled 200 sentences from each of the unlabeled bake-off training set A and B, and trained the SVM-HMM model on the labeled data. We compare the performance of the four sub-systems and the performance of the system combination method trained on: 1) 200 sentences from A, 2) 200 sentences from B, and 3) the concatenation of the 400 sentences from both A and B. We show the scores on the bake-off test set A and B in Table 4.

As we can see from the results in Table 4, the system combination method outperforms all the individual systems, and the best performance is observed when using both of the labeled data from domain A and B, which indicates the potential of further improvement by increasing the amount of in-domain training data. Also, the individual subsystems with the best performance on the two domains are different. System 1 performs well on Set B but not on Set A, so does System 4, which tops on Set A but not as good as System 1 on Set B. The system combination results appear to be much more stable on the two domains, which is a preferable characteristic if the segmentation system needs to deal with data from various domains.

5 Conclusion

In this paper we discussed a system combination method based on SVM-HMM for the Chinese word segmentation problem. The method can utilize small amount of training data in target domains to improve the performance over individual sub-systems trained on data from different domains. Experimental results show that the method is effective in improving the performance with a small amount of in-domain training data.

Future work includes adding more heterogeneous sub-systems other than CRF-based ones into the system and investigate the effects on the performance. Automatic domain adaptation for Chinese word segmentation can also be an outcome of the method, which may be an interesting research topic in the future.

References

Altun, Yasemin, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden markov support vector machines. In *Proceedings of International Conference on Machine Learning (ICML)*.

Chen, Yaodong, Ting Wang, and Huowang Chen. 2005. Using directed graph based bdmm algorithm for chinese word segmentation. pages 214–217.

Cohen, William W. and Vitor Carvalho. 2005. Stacked sequential learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJ-CAI)*.

Joachims, Thorsten, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59.

Kudo, Taku. 2003. CRF++: Yet another crf toolkit. Web page: http://crfpp.sourceforge.net/.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning (ICML)*.

Shi, Yanxin and Mengqiu Wang. 2007. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJ-CAI)*.

Sutton, Charles and Andrew McCallum, 2006. *Introduction to Statistical Relational Learning*, chapter An Introduction to Conditional Random Fields for Relational Learning. MIT Press.

Vapnik, Vladimir N. 1995. *The Nature of Statistical Learning Theory*. Springer.

Xue, Nianwen and Libin Shen. 2003. Chinese word segmentation as lmr tagging. In *Proceedings of the second SIGHAN workshop on Chinese language processing*, pages 176–179.

Zhang, Huaping, Qun Liu, Xueqi Cheng, Hao Zhang, and Hongkui Yu. 2003. Chinese lexical analysis using hierarchical hidden markov model. In *Proceedings of the second SIGHAN workshop on Chinese language processing*, pages 63–70.

Zhao, Hai and Chunyu 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 (SIGHAN-6)*, pages 106–111.

Zhao, Hai, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC-20)*, pages 87–94.