Enhancing domain portability of Chinese segmentation model using chi-square statistics and bootstrapping

Baobao Chang, Dongxu Han
Institute of Computational Linguistics, Peking University
Key Laboratory of Computational Linguistics(Peking University), Ministry Education, China
Beijing, 100871, P.R.China
chbb@pku.edu.cn,hweibo@126.com

Abstract

Almost all Chinese language processing tasks involve word segmentation of the language input as their first steps, thus robust and reliable segmentation techniques are always required to make sure those tasks well-performed. In recent years, machine learning and sequence labeling models such as Conditional Random Fields (CRFs) are often used in segmenting Chinese texts. Compared with traditional lexicon-driven models, machine learned models achieve higher F-measure scores. But machine learned models heavily depend on training materials. Although they can effectively process texts from the same domain as the training texts, they perform relatively poorly when texts from new domains are to be processed. In this paper, we propose to use χ^2 statistics when training an SVM-HMM based segmentation model to improve its ability to recall OOV words and then use bootstrapping strategies to maintain its ability to recall IV words. Experiments show the approach proposed in this paper enhances the domain portability of the Chinese word segmentation model and prevents drastic decline in performance when processing texts across domains.

1 Introduction

Chinese word segmentation plays a fundamental role in Chinese language processing tasks, because almost all Chinese language processing tasks are assumed to work with segmented input. After intensive research for more than twenty years, the performance of Chinese segmentation made considerable progress. The bakeoff series hosted by the Chinese Information Processing Society (CIPS) and ACL SIGHAN shows that an F measure of 0.95 can be achieved in the closed test tracks, in which only specified training materials can be used in learning segmentation models[1].

Traditional word segmentation approaches are lexicon-driven (Liang, 1987) and assume predefined lexicons of Chinese words are available. Segmentation results are obtained by finding a best match between the input texts and the lexicons. Such lexicon-driven approaches can be rule-based, statistic-based or in some hybrid form.

Xue (2003) proposed a novel way of segmenting Chinese texts, and it views the Chinese word segmentation task as a character tagging task. According to Xue's approach, no predefined Chinese lexicons are required; a tagging model is learned by using manually segmented training texts. The model is then used to assign each character a tag indicating the position of this character within a word. Xue's approach has become the most popular approach to Chinese word segmentation for its high performance and unified way of dealing with out-of-vocabulary (OOV) issues. Most segmentation work began to follow this approach later. Major improvements in this line of research include: 1) More sophisticated learning models were introduced other than the maximum entropy model that Xue used, such as the conditional random fields (CRFs) model which fits the sequence tagging tasks much better than the maximum entropy model (Tseng et al.,2005). 2) More tags were in-

[1] http://www.sighan.org/bakeoff2005/data/results.php.htm

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 789–798,
MIT, Massachusetts, USA, 9-11 October 2010.

troduced, as Zhao et al. (2006) shows 6 tags are superior to 4 tags. 3) New feature templates were added, such as the templates that were used in representing numbers, dates, letters etc. (Low et al., 2005)

Character tagging approaches require manually segmented training texts to learn models usually in a supervised way. The performance is always evaluated on a test set from the same domain as the training set. Such evaluation does not reveal its ability to deal with domain variation. Actually, when test set is from other domains than the domain where training set is from, the learned model normally underperforms substantially.

One of the main reasons of such performance degradation lies in the model's ability to cope with OOV words. Actually, even when the test set has the same domain properties as the training set, the ability of the model to recall OOV words is still the main obstacle to achieve better performance of segmentation. However, when the test set is different with the training set in nature, the OOV recall normally drops much more substantially, and becomes much lower.

Apart from the supervised approach, Sun et al. (2004) proposed an unsupervised way of Chinese word segmentation. The approach did not use any predefined lexicons or segmented texts. A statistic named as *md*, combining the mutual information and *t* score, was proposed to measure whether a string of characters forms word. The unsupervised nature of the approach means good ability to deal with domain variation. However, the approach did not show a segmentation performance as good as that of the supervised approach. The approach was not evaluated in F measurement, but in accuracy of word break prediction. As their experiment showed, the approach successfully predicted 85.88% of the word breaks, which is much lower than that of the character tagging approach if in terms of F measurement.

Aiming at preventing the OOV recall from dropping sharply and still maintaining an overall performance as good as that of the state-of-art segmenter when working with heterogeneous test sets, we propose in this paper to use a semi-supervised way for Chinese word segmentation task. Specifically, we propose to use χ^2 statistics together with bootstrapping strategies to build Chinese word segmentation model. The experiment shows the approach can effectively promote the OOV recall and lead to a higher overall performance. In addition, instead of using the popular CRF model, we use another sequence labeling model in this paper --- the hidden Markov Support Vector Machines (SVM-HMM) Model (Altun et al., 2003). We just wish to show that there are alternatives other than CRF model to use and comparable results can be obtained.

Our work differs from the previous supervised work in its ability to cope with domain variation and differs from the previous unsupervised work in its much better overall segmentation performance.

The rest of the paper is organized as follows: In section 2, we give a brief introduction to the hidden Markov Support Vector Machines, on which we rely to build the segmentation model. In section 3, we list the segmentation tags and the basic feature templates we used in the paper. In section 4 we show how χ^2 statistics can be encoded as features to promote OOV recall. In section 5 we give the bootstrapping strategy. In section 6, we report the experiments and in section 7 we present our conclusions.

2 The hidden Markov support vector machines

The hidden Markov support vector machine (SVM-HMM) is actually a special case of the structural support vector machines proposed by Tsochantaridis et al. (2005). It is a powerful model to solve the structure predication problem. It differs from support vector machine in its ability to model complex structured problems and shares the max-margin training principles with support vector machines. The hidden Markov support vector machine model is inspired by the hidden Markov model and is an instance of structural support vector machine dedicated to solve sequence labeling learning, a problem that CRF model is assumed to solve. In the SVM-HMM model, the sequence labeling problem is modeled by learning a discriminant function F: $X \times Y \rightarrow R$ over the pairs of input sequence and label sequence, thus the prediction of the label sequence can be derived by maximizing F over all possible label sequences for a specific given input sequence x.

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

In the structural SVMs, F is assumed to be linear in some combined feature representation of the input sequence and the label sequence $\boldsymbol{\psi}(\mathbf{x},\mathbf{y})$, i.e.

$$F(\mathbf{x},\mathbf{y};\mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\psi}(\mathbf{x},\mathbf{y})\rangle$$

Where **w** denotes a parameter vector, for the SVM-HMMs, the discriminant function is defined as follows.

$$F(x,y;\mathbf{w}) = \sum_{t=1..T}\sum_{y\in\Sigma}\langle \overline{\mathbf{w}}_y, \boldsymbol{\Phi}(\mathbf{x}^t)\rangle \delta(y^t,y)$$
$$+\eta \sum_{t=1..T-1}\sum_{y\in\Sigma}\sum_{y'\in\Sigma}\hat{\mathbf{w}}_{y,y'}\delta(y^t,y)\delta(y^{t+1},y')$$

Here $\mathbf{w} = (\overline{\mathbf{w}},\hat{\mathbf{w}})$, $\boldsymbol{\Phi}(\mathbf{x}^t)$ is the vector of features of the input sequence. $\delta(y^t, y)$ is the Kronecker function, i.e.,

$$\delta(y^t,y) = \begin{cases} 1 & \text{if } y^t = y \\ 0 & \text{if } y^t \neq y \end{cases}$$

The first term of the discriminant function is used to model the interactions between input features and labels, and the second term is used to model interactions between nearby labels. $\eta > 0$ is a scaling factor which balances the two types of contributions. (Tsochantaridis et al., 2005)

Like SVMs, parameter vector **w** is learned with the maximum margin principle by using training data. To control the complexity of the training problem, the cutting plane method is used to solve the resulted constrained optimization problem. Thus only a small subset of constraints from the full-sized optimization is checked to ensure a sufficiently accurate solution. Roughly speaking, SVM-HMM differs from CRF in its principle of training, and both of them could be used to deal with sequence labeling problem like Chinese word segmentation.

3 The tag set and the basic feature templates

As in most other work on segmentation, we use a 4-tag tagset, that is S for the character being a single-character-word by itself, B for the character beginning a multi-character-word, E for the character ending a multi-character-word and M for a character occurring in the middle of a multi-character-word.

We use the following feature templates, as are widely used in most segmentation work:

(a) C_n (n = -2, -1, 0, 1, 2)
(b) C_nC_{n+1} (n = -2, -1, 0, 1)
(c) $C_{-1}C_{+1}$

Here C refers to a character; n refers to the position index relative to the current character. By setting the above feature templates, we actually set a 5-character window to extract features, the current character, 2 characters to its left and 2 characters to its right.

In addition, we also use the following feature templates to extract features representing the character type:

(d) T_n (n = -2, -1, 0, 1, 2)
(e) T_nT_{n+1} (n = -2, -1, 0, 1)
(f) $T_{-1}T_{+1}$

Here T refers to a character type, and its value can be digit, letter, punctuation or Chinese character. The type feature is important, for there are two versions of Arabic numbers, Latin alphabets and punctuations in the Chinese texts. This is because all three kinds of characters have their internal codes defined in ASCII table, but the Chinese encoding standard like GB18030 assigns them with other double-byte codes. This causes problems for model learning as we encounter in the experiment. The training data we adopt in this paper only use numbers, letters and punctuation of double-byte codes. But the test data use both the double-byte and single-byte codes. If the type features are not introduced, most of the numbers, letters and punctuation of single-byte can not be segmented correctly. The type feature establishes links between the two versions of codes, for both versions of a digit, a letter or punctuation share the same type feature value. Actually, the encoding problem could be alternatively solved by a character normalization process. That is the mapping all single-byte versions of digits, letters and punctuations in the test sets into their double-byte counterparts as in the training set. We use the type features here to avoid any changes to the test sets.

4 The χ2 statistic features

χ^2 test is one of hypothesis test methods, which can be used to test if two events co-occur just by chance or not. A lower χ^2 score normally means the two co-occurred events are independent; otherwise they are dependent on each other. χ^2 score is widely used in computational linguistics to extract collocations or terminologies. Unsupervised segmentation approach also mainly relies on mutual information and t-score to identify words in Chinese texts (Sun et al., 2004). Inspired by their work, we believe that χ^2 statistics could also be incorporated into supervised segmentation models to deal with the OOV issue. The idea is very straightforward. If two continuous characters in the test set have a higher χ^2 score, it is highly likely they form a word or are part of a word even they are not seen in the training set.

The χ^2 score of a character bigram (i.e. two continuous characters in the text) C_1C_2 can be computed by the following formula.

$$\chi^2(C_1, C_2) = \frac{n \times (a \times d - b \times c)^2}{(a+b) \times (a+c) \times (b+d) \times (c+d)}$$

Here,

a refers to all counts of bigram C_1C_2 in the text;

b refers to all counts of bigrams that C_1 oc-curs but C_2 does not;

c refers to all counts of bigrams that C_1 does not occur but C_2 occurs;

d refers to all counts of bigrams that both C_1 and C_2 do not occur.

n refers to total counts of all bigrams in the text, apparently, $n = a + b + c + d$.

We do the χ^2 statistics computation to the training set and the test set respectively. To make the χ^2 statistics from the training set and test set comparable, we normalize the χ^2 scores by the following formula.

$$\chi^2_{norm}(C_1, C_2) = \left\lfloor \frac{\chi^2(C_1, C_2) - \chi^2_{\min}}{\chi^2_{\max} - \chi^2_{\min}} \times 10 \right\rfloor$$

To make the learned model sensitive to the χ^2 statistics, we then add two more feature templates as follows:

(g) X_nX_{n+1} (n = -2, -1, 0, 1)

(h) $X_{-1}X_{+1}$

The value of the feature X_nX_{n+1} is the normalized χ^2 score of the bigram C_nC_{n+1}. Note we also compute the normalized χ^2 score to bigram $C_{-1}C_{+1}$, which is to measure the association strength of two intervened characters.

By using the χ^2 features, statistics from the test set are introduced into segmentation model, and it makes the resulted model more aware of the test set and therefore more robust to test domains other than training domains.

Because the normalized χ^2 score is one of 11 possible values 0, 1, 2, …, 10, templates (g)-(h) generate 55 features in total.

All features generated from the templates (a)-(f) together with the 55 χ^2 features form the whole feature set. The training set and test set are then converted into their feature representations. The feature representation of the training set is then used to learn the model and the feature representation of the test set is then used for segmentation and evaluated by comparison with gold standard segmentation. The whole process is shown in Figure-1.

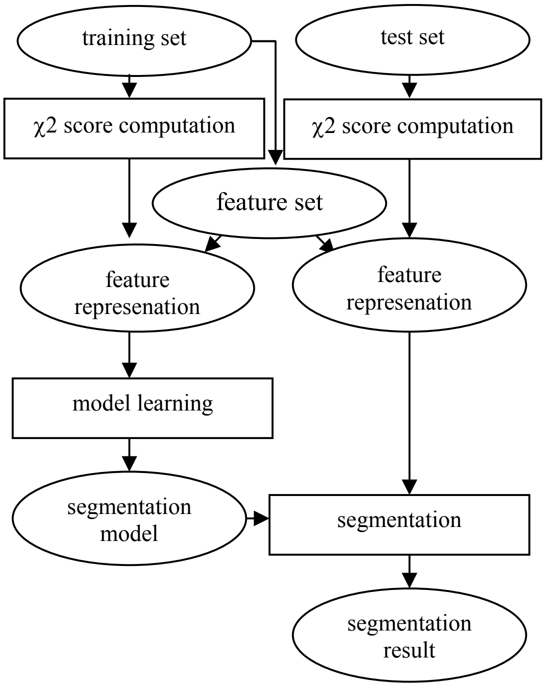

Figure-1. The workflow

By this way, an OOV word in the test set might be found by the segmentation model if the bigrams extracted from this word take higher χ^2 scores.

5 the bootstrapping strategy

The addition of the χ^2 features can be also problematic as we will see in the experiments. Even though it could promote the OOV recall significantly, it also leads to drops in in-vocabulary (IV) recall.

We are now in a dilemma. If we use χ^2 features, we get high OOV recall but a lower IV recall. If we do not use the χ^2 feature, we get a lower OOV recall but a high IV recall. To keep the IV recall from falling, we propose to use a bootstrapping method. Specifically, we choose to use both models with χ^2 features and without χ^2 features. We train two models firstly, one is χ^2-based and the other not. Then we do the segmentation for the test set with the two models simultaneously. Two segmentation results can be obtained. One result is produced by the χ^2-based model and has a high OOV recall. The other result is produced by the non-χ^2-based model and has a higher IV recall. Then we compare the two results and extract all sentences that have equal segmentations with the two models as the intersection of the two results. It is not difficult to understand that the intersection of the two results has both high OOV recall and high IV recall, if we also extract these sentences from the gold standard segmentation and perform evaluations. We then put the intersection results into the training set to form a new training set. By this new training set, we train again to get two new models, one χ^2-based and the other not. Then the two new models are used to segment the test set. Then we do again intersection to the two results and their common parts are again put into the training set. We repeat this process until a plausible result is obtained.

The whole process can be informally described as the following algorithm:

```
1. let training set T to be the original training set;
2. for I = 0 to K
    1) train the $\chi^2$-based model by using training
       set T;
    2) train the non-$\chi^2$-based model by using
       training set T;
    3) do segmentation by using the $\chi^2$-based
       model;
    4) do segmentation by using the non-$\chi^2$-
       based model;
    5) do intersection to the two segmentation re-
       sults
    6) put the intersection results into the training
       set and get the enlarged training set T
3. train the non-$\chi^2$-based model using training
   set T, and take the output of this model as the
   final output;
4. end.
```

6 The experiments and discussions

6.1 On the training set and test set

For training the segmentation model, we use the training data provided by Peking University for bakeoff 2005[2]. The training set has about 1.1 million words in total. The PKU training data is actually consisted of all texts of the People's Daily newspaper in January of 1998. So the training data represents very formal written Chinese and mainly are news articles. A characteristic of the PKU data is that all Arabic numbers, Latin letters and punctuations in the data are all double-byte GB codes; there are no single-byte ASCII versions of these characters in the PKU training data.

We use three different test sets. The first one (denoted by A) is all texts of the People's Daily of February in 1998[3]. Its size and the genre of the texts are very similar to the training data. We use this test set to show how well the SVM-HMM can be used to model segmentation problem and the performance that a segmentation model achieves when applied to the texts from the same domain.

The second and the third test sets are set to test how well the segmentation model can apply to texts from other domains. The second test set (denoted by B) is from the literature domain and the third (denoted by C) from computer domain. We segmented them manually according to the guidelines of Peking University[4] to use as gold standard segmentations. The genres of the two test set are very different from the training set. There are even typos in the texts. In the computer test set, there are many numbers and English words. And most of the numbers and letters are single-byte ASCII codes.

The sizes and the OOV rates of the three test sets are shown in Table-1.

Table-1. Test sets statistics

test set	domain	word count	OOV rate
A	Newspaper	1,152,084	0.036
B	Literature	72,438	0.058
C	Computer	69,671	0.159

For all the experiments, we use the same evaluation measure as most of previous work on segmentation, that is the Recall(R), Precision(P), F measure (F=2PR/(P+R)), IV word recall and OOV word recall. In addition, we also evaluate all the test results with sentence accuracies (SA), which is the proportion of the correctly segmented sentences in the test set.

[2] can be download from http://www.sighan.org/bakeoff2005/

[3] The corpus can be licensed from Peking University.

[4] See http:// www.sighan.org/bakeoff2005/

Table-2. Performance of the SVM-HMM and CRF model

Models	P	R	F	Riv	Roov	SA
SVM-HMM	0.9566	0.9528	0.9547	0.9620	0.7041	0.5749
CRF	0.9541	0.9489	0.9515	0.9570	0.7185	0.5570

Table-3. Performance of the basic model

test set	P	R	F	Riv	Roov	SA
A	0.9566	0.9528	0.9547	0.9620	0.7041	0.5749
B	0.9135	0.9098	0.9116	0.9295	0.5916	0.4698
C	0.7561	0.8394	0.7956	0.9325	0.3487	0.2530

Table-4. Performance of the type sensitive model

test set	P	R	F	Riv	Roov	SA
A	0.9576	0.9522↓	0.9549	0.9610↓	0.7161	0.5766
B	0.9176	0.9095↓	0.9136	0.9273↓	0.6228	0.4832
C	0.9141	0.8975	0.9057	0.9381	0.6839	0.4287

Table-5. Performance of the χ^2-based model

test set	P	R	F	Riv	Roov	SA
A	0.9585	0.9518↓	0.9552	0.9602↓	0.7274	0.5736↓
B	0.9211	0.8971↓	0.9090↓	0.9104↓	0.6825	0.4648↓
C	0.9180	0.8895↓	0.9035↓	0.9209↓	0.7239	0.4204↓

Table-6. Performance of the bootstrapping model

test set	P	R	F	Riv	Roov	SA
B	0.9260	0.9183	0.9221	0.9329	0.6830	0.5120
C	0.9113↓	0.9268	0.9190	0.9482	0.8138	0.5039

6.1 SVM-HMM vs. CRF

To show how well the SVM-HMM model can be used to model segmentation tasks and its performance compared to that of CRF model, we use the training set to train two models, one with SVM-HMM and the other with CRF.

The implementations of SVM-HMM and CRF model we use in the paper can be found and downloaded respectively via Internet. [5]

To make the results comparable, we use the same feature templates, that is feature template (a)-(c). However, SVM-HMM takes interactions between nearby labels into the model, which means there is a label bigram feature template implicitly used in the SVM-HMM. So when training the CRF model we also use explicitly the label bigram feature template to model interactions between nearby labels[6].

For the SVM-HMM model, we set ε to 0.25. This is a parameter to control the accuracy of the solution of the optimization problem. We set C to half of the number of the sentences in the training data according to our understanding to the models. The C parameter is set to trade off the margin size and training error. For CRF model, we use all parameters to their default value. We do not do parameter optimizations to both models with respect their performances.

We use test set A to test both models. For both models, we use the same cutoff frequency to feature extraction. Only those features that are seen more than three times in texts are actually used in the models. The performances of the two models are shown in Table-2, which shows SVM-HMM can be used to model Chinese segmentation tasks

[5] http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html, and http://sourceforge.net/projects/crfpp/

[6] specified by the B template as the toolkit requires.

and comparable results can be achieved like CRF model.

6.2 The baseline model

To test how well the segmentation model applies to other domain texts, we only use the SVM-HMM model with the same parameters as in section 6.1 and the same cutoff frequency.

For a baseline model, we only use feature templates (a)-(c), the performances of the basic model on the three test sets are shown in Table-3.

For the test set A, which is from the same domain as the training data, an F-score 0.95 is achieved.

For test set B and C, both are from different domains with the training data, the F-scores drop significantly. Especially the OOV recalls fall drastically, which means the model is very sensitive to the domain variation. Even the IV recalls fall significantly. This also shows the domain portability of the segmentation model is still an obstacle for the segmentation model to be used in cross-domain applications.

6.3 The type features

As we noted before, there are different encoding types for the Arabic numbers, Latin letters and punctuations. Especially, test set C is full of single-byte version of such numbers, letters and punctuations. The introduction of type features may improve performance of the model to the test set. Therefore, we use the feature tem-plates (a)-(f) to train a type sensitive model with the training data. This gives segmentation results shown in table-4. (The symbol ↓ means performance drop compared with a previous model)

As we can see, for test set A, the type features almost contribute nothing; the F-score has a very slight change. The IV recall even has a slight fall while the OOV recall rises a little.

For test set C, the type features bring about very significant improvement. The F-score rises from 0.7956 to 0.9057, and the OOV recall rises from 0.3487 to 0.6839. Different with the test set A, even the IV recall for test set C rises slightly. The reason of such a big improvement lies in that there are many single-byte digits, letters and punctuations in the texts.

Unlike test set C, there are not so many single-byte characters in test set B. Even though the OOV recall does rise significantly, the change in OOV recall for test set B is not as much as that for test set B. Type features contribute much to cross domain texts.

6.4 The χ2-based model

Compared with OOV recall for test set A, the OOV recall for test set B and C are still lower. To promote the OOV recall, we use the feature templates (a)-(h) to train a χ^2-based model with the training data. This gives segmentation results shown in table-5.

As we see from table-5, the introduction of the χ^2 features does not improve the overall performance. Only F-score for test set A improves slightly, the other two get bad. But the OOV recall for the three test sets does improve, especially for test set B and C. The IV recalls for the three test sets drop, especially for test set B and C. That's why the F scores for test B and C drop.

6.5 Bootstrapping

To increase the OOV recall and prevent the IV recall from falling, we use the bootstrapping strategy in section 5.

We set K = 3 and run the algorithm shown in section 5. We just do the bootstrapping to test set B and C, because what we are concerned with in this paper is to improve the performance of the model to different domains. This gives results shown in Table-6. As we see in Table-6, almost all evaluation measurements get improved. Not only the OOV recall improves significantly, but also the IV recall improves compared with the type-sensitive model.

To illustrate how the bootstrapping strategy works, we also present the performance of the intermediate models on test set C in each pass of the bootstrapping in table-7 and table-8. Table-7 is results of the intermediate χ^2-based models for test set C. Table-8 is results of the intermediate non-χ^2-based models for test set C. Figure-2 illustrates changes in OOV recalls of both non-χ^2-based models and χ^2-based models as the bootstrapping algorithm advances for test set C. Figure-3 illustrates changes in IV re-calls of both non-χ^2-based models and χ^2-based models for test set C. As we can see from Figure-2 and Figure-3, the ability of non-χ^2-based model gets improved to the OOV

Table-7. Performance of the intermediate χ^2-based models for test set C

I	P	R	F	Riv	Roov	SA
0	0.9180	0.8895	0.9035	0.9209	0.7239	0.4204
1	0.9084	0.9186	0.9134	0.9387	0.8126	0.4762
2	0.9083	0.9187	0.9134	0.9386	0.8138	0.4822
3	0.9068	0.9208	0.9137	0.9412	0.8131	0.4816

Table-8. Performance of the intermediate non-χ^2-based models for test set C

I	P	R	F	Riv	Roov	SA
0	0.9141	0.8975	0.9057	0.9381	0.6839	0.4287
1	0.9070	0.9249	0.9159	0.9478	0.8044	0.4869
2	0.9093	0.9254	0.9173	0.9476	0.8087	0.4947
3	0.9111	0.9266	0.9188	0.9481	0.8133	0.5030
4	0.9113	0.9268	0.9190	0.9482	0.8138	0.5039

Table-9. Performance of the intersection of the intermediate χ^2-based model and non-χ^2-based model for test C

I	P	R	F	Riv	Roov	SA
0	0.9431	0.9539	0.9485	0.9664	0.8832	0.6783
1	0.9259	0.9434	0.9345	0.9609	0.8491	0.5992
2	0.9178	0.9379	0.9277	0.9582	0.8316	0.5724
3	0.9143	0.9347	0.9244	0.9559	0.8250	0.5616

recall of the χ^2-based model as the bootstrapping algorithm advances. The abilities to recall IV words of both models improve, and even the final IV recall of the χ^2-based model surpasses the IV recall of the type sensitive model shown in Table-3. (0.9412 vs. 0.9381). To save the space of the paper, we do not list all the intermediate results for test set B. We just show the changes in OOV recalls and IV recalls as illustrated in Figure-4 and Figure-5. One can see from Figure-4 and Figure-5, the bootstrapping strategy also works for test set B in a similar way as it works for test set C.

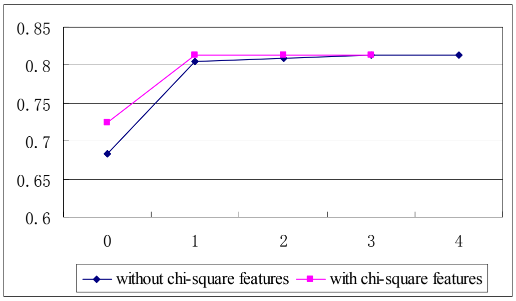

Figure-2 the Changes in OOV recalls for test set C as boot-strapping algorithm advances

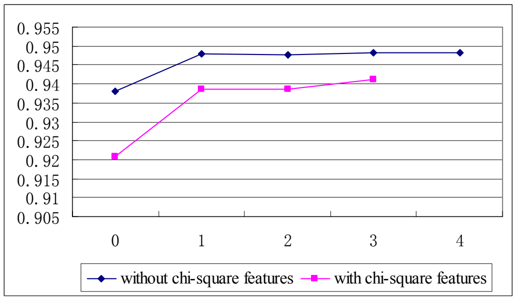

Figure-3 the Changes in IV recalls for test set C as boot-strapping algorithm advances

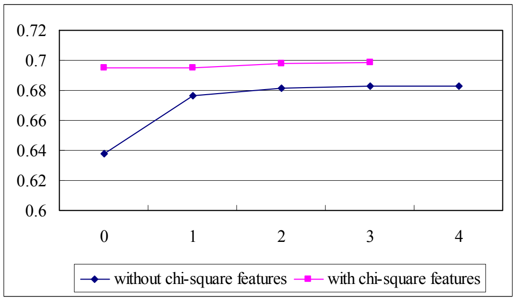

Figure-4 the Changes in OOV recalls for test set B as boot-strapping algorithm advances

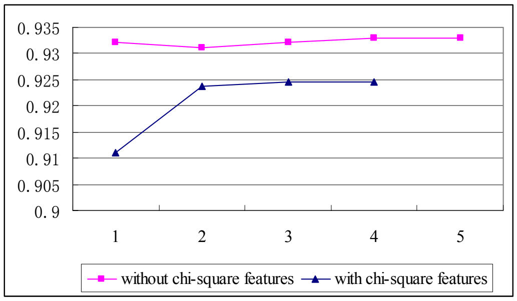

Figure-5 the Changes in IV recalls for test set B as boot-strapping algorithm advances

As we mentioned in section 5, the intersection of the results produced by χ^2-based model and non-χ^2-based model has both high OOV recall and high IV recall, that's the reason why bootstrapping strategy works. This can be seen from Table-9. However, as the algorithm progresses, both the OOV recall and IV recall of the intersection results fall, but are still higher than OOV recall and IV recall of the final results on the whole test set.

As we said before, we give also sentence accuracies of all segmentation models. With the χ^2 statistics and bootstrapping strategies, the sentence accuracy also rises. 2.8% more sentences on test set B and 7.5% more sentences on test set C are correctly segmented, compared with the type-sensitive model.

7 Conclusions

Sequence labeling models are widely used in Chinese word segmentation recently. High performance can be achieved when the test data is from the same domain as the training data. However, if the test data is assumed to be from other domains than the domain of the training data, the segmentation models always underperform substantially. To enhance the portability of the sequence labeling segmentation models to other domains, this paper proposes to use χ^2 statistics and bootstrapping strategy. The experiment shows the approach significantly increases both IV recall and OOV recall when processing texts from different domains.

We also show in this paper that hidden Markov support vector machine which is also a sequence labeling model like CRF can be used to model the Chinese word segmentation problem, by which high F-score results can be obtained like those of CRF model.

One concern to the bootstrapping approach in this paper is that it takes time to work with, which will make it difficult to be incorporated into language applications that need to responses in real time. However, we believe that such an approach can be used in offline contexts. For online use in a specified domain, one can first train models by using the approach in the paper with prepared raw texts from the specified domain and then use the final non-χ^2-based model to segment new texts of the same domain, since statistics of the target domain are more or less injected into the model by the iteration of bootstrapping.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant No. 60975054 and National Social Science Foundation of China under Grant No. 06BYY048.

We would like to give thanks to Prof. Duan Huiming for her work in preparing the gold standard segmentation and to the anonymous reviewers for their comments to the paper.

References

Altun,Yasemin et al.,2003, Hidden Markov Support Vector Machines. Proceedings of the Twentieth Iternational Conference on Machine Learning (ICML-2003), Washington DC, 2003.

Gao, Jianfeng et al., 2005, Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach, Computational Linguis-tics,Vol.31, No.4, pp531-574.

Huang, Changning et al. 2007, Chinese word segmentation: a decade review. Journal of Chinese Information Processing, Vol.21, NO.3,pp8–19.(in Chinese)

Liang, Nanyuan, 1987. ‘‘written Chinese text segmentation system--cdws”. Journal of Chinese Information Processing, Vol.2, NO.2, pp44–52.(in Chinese)

Low, Jin Kiat et al.,2005, A Maximum Entropy Approach to Chinese Word Segmentation. Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Ko-rea,. pp161-164

Sun, Maosong et al., 2004, Chinese word segmentation without using dictionary based on unsupervised learning strategy. Chinese Journal of Computers. Vol.27, No.6, pp736-742. (in Chinese)

Tseng, Huihsin et al., 2005, A conditional random field word segmenter for SIGHAN 2005, Proceedings of the fourth SIGHAN workshop on Chinese language processing. Jeju Island, Korea. pp168-171

Tsochantaridis,Ioannis et al., 2005, Large Margin Methods for Structured and Interdependent Output Variables, Journal of Machine Learning Research (JMLR), No.6, pp1453-1484.

Xue, Nianwen, 2003, Chinese Word Segmentation as Character Tagging, Computational Linguistics and Chinese Language Processing. Vol.8, No.1, pp29-48.

Zhao, Hai et al., 2006, Effective tag set selection in Chinese word segmentation via conditional random field modeling, Proceedings of the 20th Pacific Asia Conference on language, Information and Computation (PACLIC-20), Wuhan, China, pp87-94