Chinese Named Entity Recognition with the Improved Smoothed Conditional Random Fields[1]

Xiaojia Pu, Qi Mao, Gangshan Wu[2], and Chunfeng Yuan

Department of Computer Science and Technology, Nanjing University
{puxiaojia, maoq1984}@gmail.com, {gswu, cfyuan}@nju.edu.cn

Abstract. As a kind of state-of-the-art sequence classifier, Conditional Random Fields (CRFs) recently have been widely used for some natural language processing tasks which could be viewed as the sequence labeling problems such as POS tagging, named entity recognition(NER) etc. But CRFs suffer from the failing that they are prone to overfitting when the number of features grows. For NER task, the feature set is very large, especially for Chinese language, because of it's complex characteristics. Existing approaches to avoid overfitting include the regularization and feature selection. The main shortcoming of these approaches is that they ignore the so-called unsupported features which are the features appearing in the test set but with zero count in the training set. Actually, without the information of them, the generalization of the CRFs suffers. This paper describes a model called Improved Smoothed CRF which could capture the information of the unsupported features using the smoothing features. It provides a very effective and practical way to improve the generalization performance of CRFs. Experiments on Chinese NER proved the effectiveness of our method.

Keywords: Chinese named entity recognition; Conditional Random Fields; overfitting; generalization; Improved Smoothed CRF; smoothing feature

1 Introduction

Named entity recognition (NER) is one of the fundamental works in natural language processing and text processing. The aim of NER is to find the names of some special entities from the free texts, e.g. person, location, organization etc.

Compared with English, Chinese NER is more difficult because of its complex characteristics. For example, a sentence of English is a sequence of words, and the words will be separated by the space, but in Chinese, the sentence is a sequence of characters without any spaces between them.

Viewed as a sequence labeling problem, various sequence labeling models have been used to solve the NER problem such as Hidden Markov model (HMM) [1], Maximum Entropy (ME) [2], Maximum Entropy Markov model (MEMM) [3],

[1] This work was supported by the research grants from the Natural Science Foundation of China (60721002 and 60975043).

[2] Corresponding author.

© A. Gelbukh (Ed.)
Special issue: Natural Language Processing and its Applications.
Research in Computing Science 46, 2010, pp. 91-103

Received 05/12/09
Accepted 16/01/10
Final version 12/03/10

Conditional Random Fields (CRFs) [4-10],and so on. Many works have proven that CRFs get the excellent performance and are superior to the previous models [13].

A key advantage of CRFs is that they can support the use of complex features, e.g. long-range features that are overlapping and inter-dependent. This flexibility encourages the use of a rich collection of features. But with a large feature set, the CRFs are prone to overfitting [13] [15], e.g. in the Chinese NER task, because of the complex characteristics, the scale of the features will be more than millions. So it's really important to solve the overfitting problem.

There are two existing approaches to address this problem: regularization [12] [14] and feature selection [11] [15]. The regularization method is adding a regularization term to the objective function to penalize the large weight vectors. This method is also called smoothing [13] [20], similar with the smoothing methods in Maximum Entropy model [19].A typical feature selection approach is the feature induction [11], which induces the best features from the candidate sets in an iterative and approximate manner.

We conducted a detailed analysis of the issues which could influence the generalization ability of CRFs. After the study of these existing approaches, we found that the main disadvantage of these approaches is that they didn't take into account the so-called unsupported features [12], which were with zero count in the training set but occurred in the test set. Surely, these unsupported features will have an important impact on the generalization of CRFs [12]. For example, in the named entity recognition task, due to the lack of some efficient features which just appear in the test data set, some named entities could always be very difficult to recognize.

In this paper, we propose a new model called Improved Smoothed CRF, adopting a new smoothing method to help improve the generalization ability of CRFs. The insight of our method is that though the unsupported features are unknown, but we could use some high-level features to cover them. These high-level features, called smoothing features, are predefined based on the feature templates. Each smoothing feature is corresponding to a feature template. During the training procedure, in order to estimate the distribution of these smoothing features, a validation set, which is divided from the whole training set randomly, will be used to simulate the test set. Then, the ordinary features and smoothing features will be integrated together for training. During the decoding procedure, the unknown feature will all be mapped into the corresponding smoothing feature which will keep the information of unknown feature, rather than just omitting it.

In order to evaluate our method, we did some comparative experiments on Chinese NER task, and the result showed its effectiveness. Besides, we try to analyze why this method is efficient, and have a discussion about this.

The contribution of our work mainly includes: (1) a detailed analysis of the issues which influence the generalization of CRFs and some existing approaches to improve the generalization ability; (2) propose an Improved Smoothed CRF and show its effectiveness on Chinese NER task.

The paper is organized as follows. In Section 2, we will review Conditional Random Fields and analyze its generalization ability. In Section 3, we will have a detailed description about our Improved Smoothed CRF. Consequently, the experiment and the result analysis will be arranged in Section 4. Finally, we will give the conclusion and some possible future works.

2 Conditional Random Fields and Analysis of its Generalization

Conditional Random Fields (CRFs) are a class of discriminative probabilistic models trained to maximize a conditional probability. A common used special graph structure is a linear chain as shown in fig.1 and it avoids the label biased problem of Maximum Entropy Markov Models (MEMM) [3].

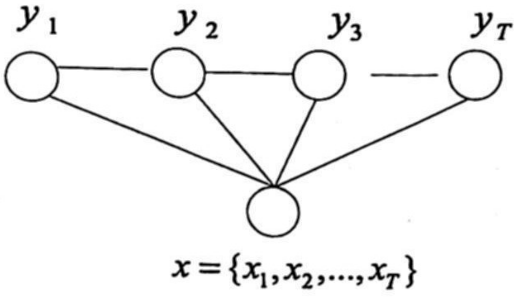

Fig. 1. A linear-chain CRF model

A linear-chain CRF[3] with parameters $\Lambda=\{\lambda_1,\lambda_2,...,\lambda_T\}$ defines a conditional probability for a state sequence $y=\{y_1,y_2,...,y_T\}$ given the observation sequence $x=\{x_1,x_2,...,x_T\}$ be the

$$p_\Lambda(y\mid x)=\frac{\exp(\sum_t\sum_k \lambda_k f_k(y_{t-1},y_t,x,t))}{Z(x)}, \tag{1}$$

where $Z(x)$ is the normalization constant that makes the probability sum to one. f_k is a feature function which is often binary-valued, and λ_k is a learned weight associated with feature f_k. Feature functions can measure any aspect of the a state transition, $y_{t-1}->y_t$,and the observation sequence, x ,centered at the current time step, t .For example, one feature function might have the value 1 when y_{t-1} is the sate B, y_t is the state I, and x_i is the character '国'.

2.1 Training of CRF

The training will be finished by maximizing the log-likelihood L_Λ on the given training set $D=\{(x_1,\mathrm{y}_1),(x_2,\mathrm{y}_2),...,(x_N,\mathrm{y}_\mathrm{N})\}$.

[3] For convince to describe our work, the CRF mentioned in this paper will all be linear-chain CRF.

$$\tilde{\Lambda} = \arg\max_{\Lambda \in R^k} L_\Lambda \tag{2}$$

where

$$L_\Lambda = \sum_{i=1}^{N} \left(\sum_t \sum_k \lambda_k f_k(y_{t-1}^i, y_t^i, x^i, t) - \log Z(x^i) \right). \tag{3}$$

Because the likelihood function is convex, we can get the optimization by seeking the zero of the gradient, i.e. the partial derivation

$$\frac{\partial L_\Lambda}{\partial \lambda_k} = \sum_{i=1}^{N} \sum_t f_k(y_{t-1}^i, y_t^i, x^i, t) - \sum_{i=1}^{N} \sum_t \sum_{y} \left(p_\Lambda(y \mid x^i) f_k(y_{t-1}, y_t, x^i, t) \right). \tag{4}$$

The first term is the expected value of f_k under the empirical distribution $\tilde{p}(x,y)$.The second term is the expectation of f_k under the model distribution $p(y \mid x)$.For the easy understanding, the formula (4) could be written as

$$\frac{\partial L_\Lambda}{\partial \lambda_k} = E_{\tilde{p}(x,y)}[f_k] - E_{p_\Lambda(y|x)}[f_k], \forall k\ . \tag{5}$$

Therefore, for the maximum likelihood solution, when the gradient is zero, the two expectations are equal. And setting this gradient to zero does not result in any closed form solution, so it typically resorts to iterative methods, such as the L-BFGS [16], which has been the method of choice since the work of [17].

2.2 Analysis of the Generalization

In this section, we try to conduct a detailed analysis of the issues which will lead to the overfitting problem and decrease the generalization ability of CRF. These issues include:

Firstly, the method of CRF training can be viewed as maximum likelihood estimation (MLE), and like other maximum likelihood methods, this type of modeling is prone to overfitting, because of its inherent weaknesses, i.e. without any prior information about the parameter distribution [18]. The common method to avoid this is using the Maximum a Posterior (MAP) method instead.

Secondly, formula (1) tells us that CRF is an exponential linear model. The scale of the parameter is equal to the number of features. With the increasing of features, the parameter dimension will be very large, and thus, the freedom of parameters will be enlarged. For Chinese NER, the scale of the features reached millions or more than millions, and almost 35% of them are sparse features. In order to fit these sparse features, some parameters will become very large. This highly uneven distribution of the parameter values will lead to the overfitting problem.

Further, during the original CRF training, usually, the features used are simply aggregated from the training set following the feature templates. But, the diversity between the training set and test set is inevitable, the features occurred in the training set are not able to cover the full feature set. So, during the decoding period, the original CRF will omit the unsupported features [12], the features occurred in the test set but with zero count in the training set, ignoring the fact that these features surely contain the useful information. In [12], they found that the unsupported features can be extremely useful for pushing Viterbi inference away from certain paths by assigning such features negative weight. So if a model trained without the information of the unsupported features, when applied to the test set, of course, the generalization ability will decrease.

2.3 Existing Work

Focusing on part of the issues we talked above, there have been some approaches to avoid the overfitting problem, including feature selection and regularization.

Feature Selection. Wise choice of features is always vital in machine learning solutions. For CRF, this method aims at the second issue we talked above to reduce the parameter dimension. Typical method for the feature selection is feature induction [11] which induces the best features from the candidate sets in an iterative and approximate manner. But the computing cost will be very large when the scale of the features grows, so it's not very practical in some applications such as Chinese NER.

Regularization. As a common way to avoid overfitting, regularization is a penalty on the parameter vectors whose norm is too large. Instead of maximizing solely the likelihood of the training set, typically, a quadratic penalty term is added

$$L_\Lambda = \sum_{i=1}^{N}\left(\sum_{t}\sum_{k}\lambda_k f_k(y_{t-1}^i, y_t^i, x^i, t) - \log Z(x^i)\right) - \sum_{k}\frac{\lambda_k^2}{2\delta^2}, \tag{6}$$

where δ^2 specifies how much the penalty is applied.

In general, the penalty prevents the absolute values of parameters $\{|\lambda_k|\}$ from becoming too large. And this method has a Maximum a Posterior (MAP) interpretation that the parameter Λ follows a Gaussian prior distribution.

Following the expression of formula (5), the gradient of the objective function could be written as:

$$\frac{\partial L_\Lambda}{\partial \lambda_k} = E_{\tilde{p}(x,y)}[f_k] - E_{p_\Lambda(y|x)}[f_k] - \frac{\lambda_k}{\delta^2}, \forall k. \tag{7}$$

The regularization method is also called smoothing [13] [15] [20], because it is similar with the smoothing methods in the Maximum Entropy model [19]. It aims at the first and second issues we talked above, by adapting a MAP training method and tighten the values of the parameters.

So we can conclude that the existing approaches all ignored the third issues, i.e. the unsupported features, actually, the unsupported features have a great impact on the generalization performance of the CRF.

An intuitive approach is that we numerate the full feature set, containing all the possible features, and then use the regularization (smoothing) method. Then during the training, all the unsupported features will get the non-zero weight. However, for Chinese NER, it's hard to numerate the full feature set due to the high scale of Chinese characters and the complex characteristics of language, and besides, doing so often greatly increases the number of parameters which will cause the overfitting.

[12] presents a compromising method called incremental support, which will introduce just some heuristic unsupported features in a iterative way. However, it's not practical for the large feature set of Chinese NER.

3 Improved Smoothed Conditional Random Fields

We propose a new model called Improved Smoothed Conditional Random Field, which provides a practical way to capture the information of unsupported features, by the means of inducing the smoothing features.

The inspiration of our method is that though the unsupported features are unknown, but we could use some high-level features to cover them. Every high-level feature, called smoothing feature, is predefined corresponding to a feature template. Since the unsupported feature is also generated based on a special feature template, we could use the smoothing feature to replace it.

During the training, in order to extract the smoothing features with the estimation of the distribution of them, we use a validation set as the simulation of test set. The validation set is divided from the whole training set randomly. And in the end, the ordinary features and smoothing features will be put together for the training. During the decoding, the unknown feature will all be mapped to the corresponding smoothing feature rather than solely omitted.

The advantage of our method is that we provide a practical way to capture the information of unsupported features, and meanwhile, the parameter dimension will not be enlarged too much because the smoothing feature set is small.

3.1 Smoothing Features

The so-called smoothing features are predefined based on the feature templates, as shown in Table 2.

For a given training set $Train\_set$, after the feature selection, we could get a feature vector $F_1 = \{f_1, f_2, \ldots, f_K\}$, and for a given validation set V_set, following the same feature selection method, we could get a feature vector $F_2 = \{f_1', f_2', \ldots, f_m'\}$ .we use the $T(f)$ to represent the template which generates the feature f . If $f \in F_2$ and $f \notin F_1$, we use a special feature $f^*(T)$ to describe $T(f)$, and $f^*(T)$ is called the smoothing feature for a given feature template.

3.2 Extraction of Features

For the training of CRF, the extraction of features is vital as the first step, including calculating the frequencies of each feature. The extraction sequence for the Improve Smoothed CRF is shown in Fig.2.

Different with the original CRF, during the extraction procedure, the Improved Smoothed CRF will divide the features into ordinary features and smoothing features. As shown in Fig. 2, the training set D is divided into two subsets, D_1 and D_2. D_1 is the data set, and D_2 is the validation set. F_3 is the collection of the smoothing features, and $F*$, containing the distribution of smoothing features, is the final smoothing feature set. Actually, the validation set is used to simulate the test set to get distribution of smoothing features. F_1 and $F_2 - F*$ are the ordinary features. F is the final full feature set with ordinary features and smoothing features including the distribution of them.

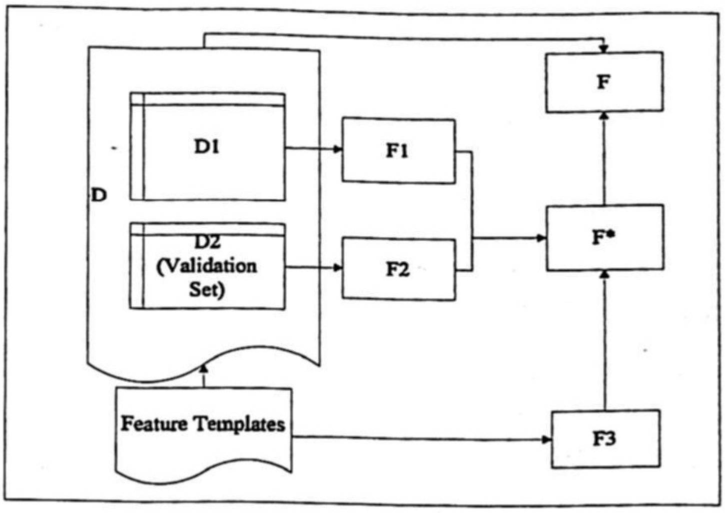

Fig. 2. The sequence of extracting features

3.3 The Definition of Improved Smoothed CRF

Following the definition of original CRF, the conditional probability of the state sequence $y = \{y_1, y_2, ..., y_T\}$ given the observation sequence $x = \{x_1, x_2, ..., x_T\}$ defined by the Improved Smoothed CRF could be formulized as

$$p(y|x) = \frac{\exp\left(\sum_t \left[\sum_k \lambda_k f_k(y_{t-1}, y_t, x, t) + \sum_m \mu_m g_m(y_{t-1}, y_t, x, t)\right]\right)}{Z(x)} \quad (8)$$

where $g_m(y_{t-1}, y_t x, t) \in F*$ is the smoothing feature at the time t, μ_m is the weight associated with the feature.

From formula (8), it seems that the new model is the same as the original CRF, but essentially, they are different. The new model use the predefined smoothing features to cover the unsupported features, and predict the distribution of unsupported features based on the validation set which is the simulation of the test data.

Incorporating the smoothing features, the model will more fit to the test data, and improve the generalization performance.

Because it can be seen as an extension of regularization (smoothing) in an improved smoothing way, we call it the Improved Smoothed CRF.

3.4 Training of the Improved Smoothed CRF

After the extraction of the features, the training procedure is the same as the original CRF, the entire training process is shown blow:

Algorithm: The training of Improved Smoothed CRF

```
Improved Smoothed CRF training ($D,F\_temp$)
  Input: $D=\{d_i\},di=\{x_i,y_i\},F\_temp$   //$D$ is the training set,$F\_temp$
are the feature templates
  Output: $\Lambda=\{\lambda_1,...,\lambda_k,\mu_1,...,\mu_m\}$  //$\Lambda$is the weights of the
parameters, $\lambda_1,...,\lambda_k$ for the ordinary features, $\mu_1,...,\mu_m$ for the
smoothing features
begin
    divide $D$ into $D_1$ and $D_2$ (validation set)
    extracting $F_1$, $F_2$ from $D_1$, $D_2$
    generate $F_3$ based on $F\_temp$
    extracting $F*$ with $F_1$, $F_2$ and $F_3$
    $\Lambda$= 0;//the initialization
    do {
    for $d=\{x,y\}$ in $D$ do
       calculate the $p_\Lambda(y|x)$of $d$ based on formula(1),
       for $\lambda/\mu$ in the features of $d$ do
            update $\partial L_\Lambda/\partial\lambda(\mu)$ based on formula(6)
       end do
       update $L_\Lambda$ based on $p_\Lambda(y|x)$
    end do
    $L-BFGS(\Lambda,L_\Lambda,\{\partial L/\partial\lambda(\mu)_i\})$
    //update the parameters of L-BFGS
    }until(converged)
end.
```

3.5 Decoding of the Improved Smoothed CRF

The decoding process is the same as the original CRF after replacing the unsupported features with the smoothing features.

4 Experiments

We did comparative experiments on Chinese NER task to evaluate our method in two different corpuses, MSR [21] and PKU [22]. We pick up 3000 sentences separately from the two corpuses as the test sets, the remains are used for training.

Chinese NER problem can be solved mainly in two levels: word level [8] [9] and character level [6] [10].In order to analyze the results clearly, we did our experiment in the character level.

We extend the CRF4 in Mallet toolkit [23] to implement our improved smoothed CRF, and the baseline model is CRF4, which is an implementation of original CRF.

The evaluation metrics is precision, recall and F measure.

4.1 Feature Templates and Tag Set

The feature templates are shown in Table 1. Since the purpose of our experiments is to compare the performance between the original CRF and Improved Smoothed CRF, so the features templates we used are the basic and simple ones, which will make the analysis of the results more clearly and persuasive.

Table 1. Feature templates used in the experiment

type	template
Base feature	$C_n(n=-2,-1,0,1,2)$
Bi-gram feature	$C_nC_{n+1}(n=-2,-1,0,1)$
	$C_{-1}C_1$

C_n is the character with the relative distance n from the observation position, these feature templates are the basic templates for Chinese NER.

Corresponding to the feature templates, some smoothing features are listed in Table 2. We just list the node features without the combination with the state transition features, actually during the training, the state transition should be considered.

Table 2. Some corresponding smoothing features

template	Smoothing features
C2	<Unknown>@2
C-1C0	<Unknown>_-1&_<Unknown>@0
C1C2	<Unknown>@1_&_<Unknown>@2
…	…

The tag set is the coding of the states, and we choose BIO as our tag set, the full states are $\{O,B-NR,I-NR,B-NS,I-NS,B-NT,I-NT\}$.

NR , NS and NT represent the name, location and organization respectively. O represents that it's not the named entity, and $B-NR$ represents that it's the first character of the named entity, $I-NR$ represents that it's the character of the named entity but not the first. The NS and NT are similar.

4.2 Comparative Results

The comparative experiments results are shown in Table 3 and Table 4.The three types of named entities to be recognized are person, location and organization.

Table 3. Comparative results based on PKU corpus

	Metric	Person	Loc.	Org.	All
Original CRF	Precision	92.83%	89.62%	85.21%	90.26%
	Recall	80.69%	81.45%	77.54%	80.13%
	F-measure	86.33%	85.34%	81.19%	84.89%
Imp. Smoothed CRF	Precision	93.05%	90.72%	87.66%	91.28%
	Recall	87.96%	85.38%	81.46%	85.90%
	F-measure	90.43%	87.97%	84.44%	88.51%

Table 4. Comparative results based on MSR corpus

	Metric	Person	Loc.	Org.	All
Original CRF	Precision	93.95%	86.02%	82.08%	87.44%
	Recall	72.14%	74.14%	64.25%	70.79%
	F-measure	81.62%	79.64%	72.08%	78.24%
Imp. Smoothed CRF	Precision	92.69%	87.72%	81.99%	87.84%
	Recall	82.36%	79.02%	70.49%	77.78%
	F-measure	87.22%	83.15%	75.81%	82.50%

For the experiments in PKU corpus, we used about 16 thousand sentences for training, and in MSR corpus, we used about 20 thousand sentences.

Because we aim to compare the performance of the models rather than get the best recognition result, so we didn't use the entire corpus for training. The scale of our validation set is about the 1/3 to 1/2 of the training corpus.

We can find that with the Improved Smoothed CRF, the F measure improved in both corpuses, e.g. 3.62% in PKU, 4.26% in MSR.

The recall got the largest increase, e.g. 5.77% in PKU, 6.99% in MSR. This increase indicates that the information of unsupported features is very useful, and our model could capture them efficiently. With this information, we could recognize some entities which couldn't be recognized correctly using the original model.

4.3 The Change of the Parameters

In order to know clearly about the impact on the parameters by the Improved Smoothed CRF, some values of the parameters $f(y_{t-1}, y_t, \text{'国'})$ are shown in Table 5.

Table 5. The Change of some example parameters

Features(国)	Original CRF	Imp. Smoothed CRF
O -> O	0.27270093137463947	-0.09193322956278925
O -> B-ns	0.5236346196173783	0.3431855304817065
B-nr -> O	-0.18969131328341626	0.11111461613932119
B-nr -> I-nr	0.14558200966493068	0.12232168710742694
B-nr -> B-ns	0.05574692403247397	-0.11382501213178316
B-nr -> B-nt	0.03156266967717719	0.11615382524919472
...	...	...

We can see that the parameters become more tightly, which could help to improve the generalization performance and the experiment result in deed showed this.

4.4 Different Training Size

We also did an experiment in MSR corpus to find how the size of training set influences the performance. The result is shown in fig. 3.

The 4 training set we used are respectively 25%, 50%, 75%, 100% of the whole corpus, for the 75% and 100% of the corpus, due to the limit of the memory of JVM and our machine, we discarded the sparse features, but the experiment result still proved that our model could improve the performance.

With the increase of the training scale, the improvement reduced relatively. This indicated that the unseen information of the unsupported features is more useful for the small training set, and our model could deal with this effectively.

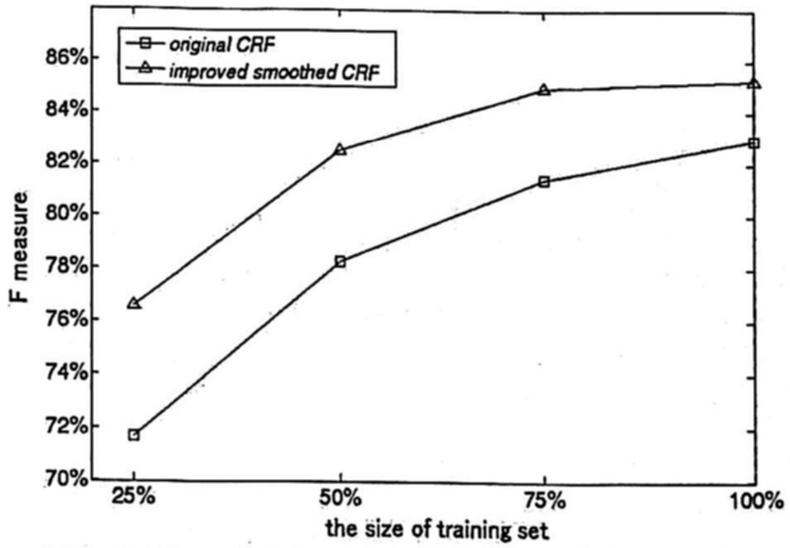

Fig. 3. The result based on different training set size

4.5 Analysis and discussion

The most particulars of our smoothing method are that they could capture some unknown features, and also could reduce the computation cost. We think that, for NER task, the sparse unknown features have something in common, and our method could make the best use of the common characteristics.

Anyway, for Chinese NER, this special task, our smoothing method give a very effective and practical way to improve the generalization of CRF. And in the future works, we want to do some experiments in other tasks to analyze that whether this smoothing method is limited to some special tasks or depends on factors like text genre or text domain.

5 Conclusion

In this paper, we take a detailed analysis of the factors influencing the generalization ability, and then we propose an Improved Smoothed CRF. The substance of our work is that using smoothing features to capture the information of unsupported features and using the validation set to simulate the test set.

This Improved Smoothed CRF provides a practical and effective way to increase the generalization performance of CRF. The experiments on Chinese NER proved its effectiveness. We will incorporate more meaningful feature templates such as surname list, location ending list, etc. in [5-10] to achieve a better result for the Chinese NER.

And we believe that our method could also be useful in other NLP tasks, e.g. POS tagging, Chinese word segmentation, etc.

References

1. L. R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: Proceedings of IEEE, pp. 257--285. (1989)
2. A. L. Berger, S. A. D. Pietra, V. J. D. Pietra: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, pp. 39-71. (1996)
3. A. McCallum, D. Freitag, F. Pereira: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: Proceedings of ICML'2000.
4. J. Lafferty, A. McCallum, F. Pereira: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of ICML' 2001, pages 282-289.
5. A. McCallum, W. Li: Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: Proceedings of 7th Conference on Natural Language Learning (CoNLL). (2003)
6. W. Chen, Y. Zhang, H. Isahara: Chinese Named Entity Recognition with Conditional Random Fields. In: Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing & the 3rd International Chinese Language Processing Bakeoff. (2006)
7. A. Chen, F. Peng, R. Shan, G. Sun: Chinese Named Entity Recognition with Conditional Probabilistic Models. In: Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing& the 3rd International Chinese Language Processing Bakeoff. (2006)

8. Z. Xu, X. Qian, Y. Zhang, Y. Zhou: CRF-based Hybrid Model for Word Segmentation, NER and even POS Tagging. In: The 4th International Chinese Language Processing Bakeoff & the First CIPS Chinese Language Processing Evaluation. (2008)
9. H. Zhao, C. Kit: Unsupervised Segmentation Helps Supervised Learning of Character Tagging for Word Segmentation and Named Entity Recognition. In: The 4th International Chinese Language Processing Bakeoff & the First CIPS Chinese Language Processing Evaluation. (2008)
10. Yuanyong Fen, Le Sun, Wenbo Li, Dakun Zhang: A Rapid Algorithm to Chinese Named Entity Recognition Based on Single Character Hints. Journal of Chinese Information Processing, Vol.22, No.1, pp.104-110. (2008)
11. A. McCallum: Efficiently Inducing Features of Conditional Random Fields. In: Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI). (2003)
12. F. Peng, A. McCallum: Accurate Information Extraction from Research Papers using Conditional Random Fields. In: Proceedings of Human Language Technologies: The 11thAnnual Conference of the North American Chapter of the Association for Computational Linguistics. (2004)
13. Roamn Klinger, Katrin Tomanek: Classical Probabilistic Models and Conditional Random Fields. Algorithm Engineering Report TR07-2-013, Dortmund University of Technology. (2007)
14. Charles Sutton, Andrew McCallum: An introduction to Conditional Random Fields for Relational Learning. In: Lise Getoor, Benjamin Taskar (Editors.): Introduction to Statistical Relational Learning, MIT Press, Chap.1, pp.93-127. (2006)
15. Trevor A. Cohn: Scaling Conditional Random Fields for Natural Language Processing. Ph.D Thesis, University of Melbourne. (2007)
16. D.C. Liu, J. Nocedal: On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming, pp. 49-55. (1989)
17. F. Sha, F. Pereira: Shallow Parsing with Conditional Random Fields. In: Proceedings of Human Language Technologies: The 11thAnnual Conference of the North American Chapter of the Association for Computational Linguistics. (2003)
18. Stanley F. Chen, Rosenfeld Ronald: A Survey of Smoothing Techniques for ME Models. In: IEEE Transactions on Speech and Audio Processing, Vol.8, No.1, pp. 37-50. (2000)
19. Stanley F. Chen, Rosenfeld Ronald: A Gaussian Prior for Smoothing Maximum Entropy Models. Technical Report CMU-CS-99-108, Carnegie Mellon University. (1999)
20. D. L. Vail, J. D. Lafferty, M. M. Veloso: Feature Selection in Conditional Random Fields for Activity Recognition. In: Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, Oct 29- Nov 2.(2007)
21. Corpus of MSR, http:// www.sighan.org/bakeoff2006/
22. Corpus of Peking University, http://icl.pku.edu.cn/icl_groups/corpus/dwldform1.asp
23. Mallet Toolkit, http://mallet.cs.umass.edu