A Simple and Efficient Model Pruning Method for Conditional Random Fields

Hai Zhao and Chunyu Kit

Department of Chinese, Translation and Linguistics,
City University of Hong Kong,
83 Tat Chee Avenue, Kowloon, Hong Kong, China
haizhao@cityu.edu.hk, ctckit@cityu.edu.hk

Abstract. Conditional random fields (CRFs) have been quite successful in various machine learning tasks. However, as larger and larger data become acceptable for the current computational machines, trained CRFs Models for a real application quickly inflate. Recently, researchers often have to use models with tens of millions features. This paper considers pruning an existing CRFs model for storage reduction and decoding speedup. We propose a simple but efficient rank metric for feature group rather than features that previous work usually focus on. A series of experiments in two typical labeling tasks, word segmentation and named entity recognition for Chinese, are carried out to check the effectiveness of the proposed method. The results are quite positive and show that CRFs models are highly redundant, even using carefully selected label set and feature templates.

Key words: Conditional Random Fields, Model Pruning

1 Introduction

CRFs are a structure learning tool first introduced in [1]. CRFs often outperform maximum entropy Markov model (MEMM) [2], another popular structure learning method. The main reason is that, among directed graphical models, CRFs do not suffer from the label bias problem as much as MEMM and other conditional Markov models do [1]. So far, CRFs have been successful in a good number of applications, especially in natural language processing [3].

As any other general-purpose machine learning tool, feature engineering is also a central part in CRFs learning. Typically, selecting good and sufficient features from auto constructed candidate set is an open problem since [1]. However, most existing work is only concerned with feature refinement in training stage for training speedup or performance enhancement (forward feature selection) [4–7], and few existing work considers model pruning for the decoding requirement (backward feature elimination) [8]. We will consider the latter in this paper. Because of rapid progress of modern computer manufacture technology, larger and larger data are fed into machine learning to build larger and larger models. For example, tens of millions features will be encountered in recent research move,

but it is not always convenient to carry a model with so many features. In this study, we will consider to prune an existing CRFs model for storage reduction and decoding speedup. Our purpose is to reduce the given CRFs model as much as possible without or with least performance loss. Namely, we try to indicate those most necessary part in the model.

The most difference between our idea and previous work, either forward or backward feature pruning, is that structural factor is involved in our consideration. Thus a simple criterion is proposed to rank feature groups rather than features that previous work usually focused on.

The remainder of the paper is organized as follows. Section 2 proposes a criterion to ranking all groups of features in a given CRFs model. Section 3 presents our experimental results. Related work is discussed in Section 4. Section 5 concludes the paper and discusses future work.

2 The Proposed Method

2.1 CRFs

Given an input (observation) $\mathbf{x} \in X$ and parameter vector $\lambda = \lambda_1, ..., \lambda_M$, CRFs define the conditional probability $p(y|x)$ of a particular output $\mathbf{y} \in Y$ as being proportional to a product of potential functions on the cliques (namely, x) of a graph, which represents the interdependency of $\mathbf{y}$ and $\mathbf{x}$.

$$p(\mathbf{y}|\mathbf{x};\lambda) = Z_\lambda(\mathbf{x})^{-1} \prod_{c \in C(\mathbf{y},\mathbf{x})} \Phi_c(\mathbf{y},\mathbf{x};\lambda) \tag{1}$$

where $\Phi_c(\mathbf{y},\mathbf{x};\lambda)$ is a non-negative real value potential function on a clique $c \in C(\mathbf{y},\mathbf{x})$. $Z_\lambda(\mathbf{x}) = \sum_{\hat{\mathbf{y}} \in Y} \prod_{c \in C(\mathbf{y},\mathbf{x})} \Phi_c(\hat{\mathbf{y}},\mathbf{x};\lambda)$ is a normalization factor over all output values, Y.

A log-linear combination of weighted features,

$$\Phi_c(\mathbf{y},\mathbf{x};\lambda) = \exp(\lambda \mathbf{f}_c(\mathbf{y},\mathbf{x})), \tag{2}$$

is often used as individual potential functions, where $\mathbf{f}_c$ represents a feature vector obtained from the corresponding clique c. It has been proved that the form in equation (2) is a sufficient and necessary condition to guarantee the probability distribution over the graph Markovian. That is, $\prod_{c \in C(\mathbf{y},\mathbf{x})} \Phi_c(y,x) = \exp(\lambda F(y,x))$, where $F(\mathbf{y},\mathbf{x}) = \sum_c \mathbf{f}_c(\mathbf{y},\mathbf{x})$ is the CRF's global feature vector for $\mathbf{x}$ and $\mathbf{y}$.

The most probable output $\hat{\mathbf{y}}$ is given by $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in Y} p(\mathbf{y}|\mathbf{x};\lambda)$. However $Z_\lambda(\mathbf{x})$ never affects the decision of $\hat{\mathbf{y}}$ since $Z_\lambda(\mathbf{x})$ does not depend on $\mathbf{y}$. Thus, we can obtain the following discriminant function for CRFs:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in Y} \lambda F(\mathbf{y},\mathbf{x}) \tag{3}$$

2.2 Pruning via Ranking Feature Groups

In equations (1) and (2), $\Phi_c(\mathbf{y}, \mathbf{x}; \lambda)$ is often rewritten as two parts,

$$\Phi_c(\mathbf{y}, \mathbf{x}; \lambda) = \Phi_{c1}(\mathbf{y}, \mathbf{x}; \lambda)\Phi_{c2}(y, \mathbf{x}; \lambda), \tag{4}$$

where

$$\Phi_{c1}(\mathbf{y}, \mathbf{x}; \lambda) = \exp(\sum_k \lambda'_k f'_k(\mathbf{y}, x)), \tag{5}$$
$$\Phi_{c2}(y, \mathbf{x}; \lambda) = \exp(\sum_k \lambda_k f_k(y, x)).$$

In above equations, $f_k(y, x))$ is a state feature function that uses only the label at a particular position, and $f'_k(\mathbf{y}, x))$ is a transition feature function that depends on the current and the previous labels. Consider that state and transition features play quite different roles in decoding, the pruning will be respectively performed on them. In practice, state features often covers the most part of all ones in a given model. Thus, the pruning mostly aims at state features.

Prevailingly, a feature function, either state- or transition-, can be written as binary form,

$$f_H(y') = \begin{cases} 1, \text{ if } H \text{ holds and } y = y' \\ 0, \text{ otherwise,} \end{cases} \tag{6}$$

where H is a predefined condition (rule) around the current clique. Incorporated with their corresponding weight (score) λ, all features f consist of the model after training is completed.

Two natural ways are considered for the model pruning. One is based on the condition H that determines the feature. Feature count cut-off according to its occurrence in the training data is such a method.The other is based on feature weight statistics. In theory, λ value may range from negative infinite to positive infinite. The larger this value is, the more significant the respective feature is. It seems that we can rank all features simply according to λ value. However, decoding structural object is more sophisticated than multi-class classification procedure over a single clique because structure characteristics are additionally involved for the former. For example, Markovian characteristics should be often considered in structure learning, which cannot be effectively handled by most multi-classification algorithms. Without considering structural loss, direct filtering those low scored features in CRFs learning and decoding will inevitably lead to a dramatic decrease of performance in most cases.

Having sequence labeling task as an example, we may regard the decoding over the given structure defined by CRFs approximately as two-stage procedure. The first stage is to compute all boundary probabilities for each clique, namely, the probability distribution to output all possible labels over a clique. The second stage will find a series of labels with the maximal joint probability through searching a path over the matrix constructed by these boundary probabilities.

We will focus on the first stage since its output consists of the basis of the search in the second stage. As we cannot determine the exact label for a clique

before the decoding is completed, we have to consider a groups of activated features $f_H(y)$, for all $y \in Y$. Hereafter, we also call these features, $\{f_H(y), \forall y \in Y\}$ w.r.t some H, a feature group[1]. Here feature group pruning rather than feature pruning means that all features activated according to the predefined condition H over x will be discarded in decoding. When two groups of features, f_{H1} and f_{H2}, are activated for a clique c, our question will be, which one will be more informative? The answer is the one which can help us more confidently to predict a label to c. So, the group of features with more unbalanced weight scores can be more informative for the further prediction during search optimization. We take the variance of these scores as ranking metric of every groups of features,

$$v(\lambda_H) = N^{-1} \sum_y (\lambda_H(y) - \text{avg}(\lambda_H))^2, \tag{7}$$

where $\lambda_H(y)$ is the corresponding weight for feature $f_H(y)$, and $\text{avg}(\lambda_H) = N^{-1} \sum_y \lambda_H(y)$ and N is the number of $f_H(y)$ in the given feature group, it should not be larger than the number of label set, $|Y|$, because not all $f_H(y, x)$, $\forall y \in Y$ must occur in the training data. We hereafter will keep those groups of features with the highest scores (variance values) according to the pruning criterion formula (7) in the reduced model.

3 Experiments

3.1 Settings

A series of experiments are performed to check the effectiveness of the proposed pruning method through learning and decoding in order-1 linear-chain CRFs. Gaussian prior is adopted in all CRFs training to avoid overfitting [2]. Two typical sequence labeling tasks in Chinese, word segmentation (WS) and named entity recognition (NER), are evaluated. Two data sets of word segmentation, AS and MSRA, are from shared task Bakeoff-2 [3], and two data sets of named entity recognition, CityU and MSRA, are from Bakeoff-3 [4], as summarized in Table 1

[1] We take an example to explain what a feature group is. Assume that the label set is $\{A0, A1, A2\}$. $H = \{previous\_word = `fire'\}$, a feature group about H contains three features, $f_H(A0)$, $f_H(A1)$, and $f_H(A2)$, if all of them occur in the training corpus. Note that in some literatures a feature group defined here is also identified as a single feature [4]. Since CRFs model will assign three different weight scores for $f_H(A0)$, $f_H(A1)$, and $f_H(A2)$, respectively, we regard them three different features, and call the set, $\{f_H(A0), f_H(A1), f_H(A2)\}$, a feature group.

[2] We choose the best Gaussian prior according to a series of cross-validation experiments in the original model, and the corresponding values will be kept unchanged as pruning. Though some existing studies show that L_1 regularization is effective in producing a more sparse model than L_2 regularization, our empirical study shows that L_1 regularization cannot provide satisfied performance for these two labeling tasks as L_2 regularization does.

[3] http://www.sighan.org/bakeoff2005

[4] http://www.sighan.org/bakeoff2006

with corpus size in number of characters (tokens). The performance of both WS and NER is measured in terms of the F-measure $F = 2RP/(R + P)$, where R and P are the recall and precision of segmentation or NER.

Table 1. Corpora Statistics

Corpus	WS		NER	
	AS	MSRA	CityU	MSRA
Training(M)	8.39	4.05	2.71	2.17
Test(K)	198	184	364	173

Existing work shows that both WS and NER for Chinese can be effectively formulated as character tagging task [9–12]. According to these results, especially from the latter, we use a set of carefully selected label set and corresponding feature sets to train model for these two tasks, respectively. We will show that the model pruning is still effective even in these models that can bring up state-of-the-art performance. 6-tag set that represents character position in a word is kept using for word segmentation task as in [11, 12]. We have show that 6-tag set can bring state-of-the-art performance since our previous work in [10, 11]. Its six tags are B, B_2, B_3, M, E and S. For NER, we need to tell apart three types of NEs, namely, *person*, *location* and *organization* names. Correspondingly, the six tags are also adapted for characters in these NEs but distinguished by the prefixes Per-, Loc- and Org-. Plus an additional tag "O" for none NE characters, altogether we have 19 tags for NER. The following example illustrates how characters in words of various lengths are tagged in a sequence for word segmentation learning.

他 / 来自 / 阿根廷 / 首都 / 布宜诺斯艾利斯 /。
he / is from / Argentine / capital / Buenos Aires /.
S B E B B_2 E B E BB_2B_3MM M E S

And this is an example for NE tagging.

[马 拉 多 纳]/Per / 来 / 自 /[阿 根 廷]/Loc /。
Maradona / is from / Argentine /.
Per-B Per-B_2 Per-B_3 Per-E O O Loc-B Loc-B_2 Loc-E O

Six n-grams, C_{-1}, C_0, C_1, $C_{-1}C_0$, C_0C_1, and $C_{-1}C_1$, are selected as features for both tasks. As for NER, five unsupervised segmentation features generated by accessor variety criterion with respect to n-grams of different lengths are also introduced as in [12].

A performance comparison of our trained model (without any pruning) and other best existing results is given in Table 2. This comparison shows that we will start the model pruning experiments based on a system with state-of-the-art performance.

Table 2. Performance comparison and number of feature groups

Participant	WS		NER	
	AS	MSRA	CityU	MSRA
Bakeoff Best	.952	.964	.8903	.8651
Zhang et al. [13]	.951	.971		
Ours	.953	.973	.8918	.8630
#Feature group	2.60M	1.55M	1.46M	1.10M

3.2 Pruning Results

The numbers of feature groups in four models are given at the bottom of Table 2. Note that all these models contain millions of feature groups.

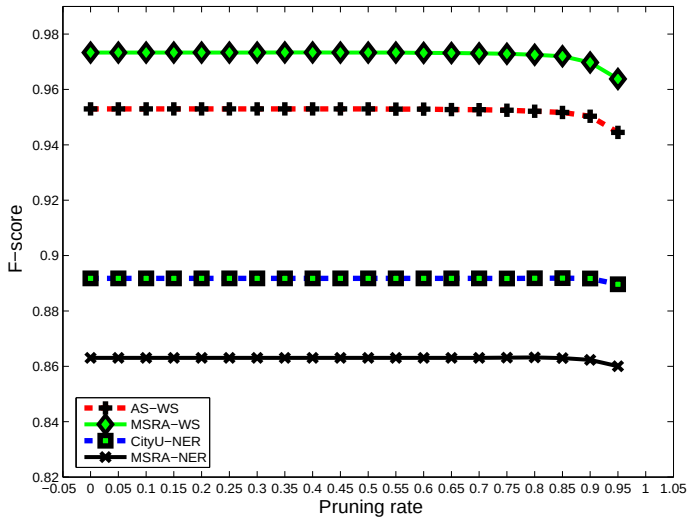
According to the ranking metric in (7), we remove the model step by step and observe how the performance changes. Our experimental results show that any performance loss is not encountered until pruning rate is larger than 65% for two WS tasks and 90% for two NER tasks. These results are shown in Figure 1(a). This indicates that these models are highly redundant.

In Figure 1(b), we keep few feature groups with top scores and observe how the performance varies. Still, we find few features help a great deal in performance. 1/50 features can give above 97% performance in all tasks. The value 97% and F-score rate in Figure 1(b) are computed in this way: divide F-score with 1/50 or some other amount of features by F-score with full features.

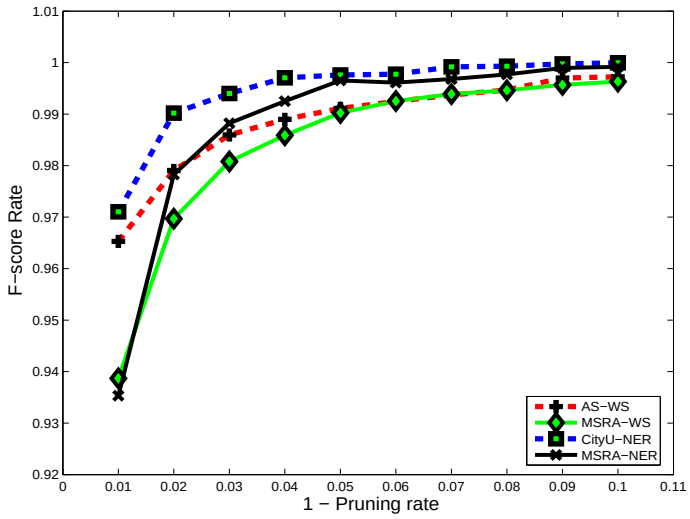
As a comparison, we compare the proposed method with feature count cut-off[5]. We prune the model according to the proposed ranking metric with the same rate as cut-off thresholds are set to 2, 3, 4, and 5, respectively. The pruning rates of each cut-off thresholds are given in Table 3. The performance comparison between our method and cut-off method are illustrated in Figures 2. We find that the simple cut-off according to the occurrence times of features may cause serious performance loss, while our pruning method only cause little for the same pruning rate.

The experimental results have shown that the proposed method is effective and CRFs models that we adopt at least are highly redundant. We don't give the results about decoding speedup after model pruning, because decoding speed is highly sensitive detailed decoding algorithm. However, feature reduction in a model surely helps speedup decoding since the search space for decoding is narrowed.

[5] Here, the term, 'feature count', aims at feature group. Thus it actually means the sum of feature count within a feature group. For example, as for a feature group, $f_H = \{f_H(A0), f_H(A1), f_H(A2)\}$, if three features, $f_H(A0)$, $f_H(A1)$, and $f_H(A2)$, occur 8, 6, and 5 times, respectively, then feature count for f_H should be 19. If cut-off threshold is set to 20, then this feature group will be discarded.



(a)



(b)

Fig. 1. Performance with different model pruning rates (F-score rate in (b) is obtained through divided by the F-score without any model pruning.)

4 Related Work and Discussions

Basically, the proposed method is different from those mentioned in [8]. In our scheme, not a single feature but a feature group is picked up for pruning. As to our best knowledge, little existing work is concerned with CRFs model pruning,

Table 3. The rates and number of Pruned feature groups for each cut-off thresholds

Cut-off	WS				NER			
	AS		MSRA		CityU		MSRA	
	Rate(%)	#group(M)	Rate	#group	Rate	#group	Rate	#group
≥ 1	00.0	0.00	00.0	0.00	00.0	0.00	00.0	0.00
≥ 2	47.5	1.24	49.2	0.76	52.8	0.77	52.8	0.58
≥ 3	63.3	1.65	65.0	1.01	68.9	1.01	68.6	0.75
≥ 4	71.3	1.85	72.9	1.13	76.8	1.12	76.3	0.84
≥ 5	76.3	1.98	77.8	1.21	81.5	1.19	81.0	0.89

either, though some work has carefully discussed so-called feature selection issue [4, 14].

Both model pruning and feature selection need a ranking metric to evaluate which feature is better among all candidates, so both of them share the similar idea in this sense. The differences, according to our understanding, are what rank metric is chosen and which kind of knowledge, posterior- or prior-, is adopted. In [4], the gain score of a new feature f_H with associated weight λ_H is given by:

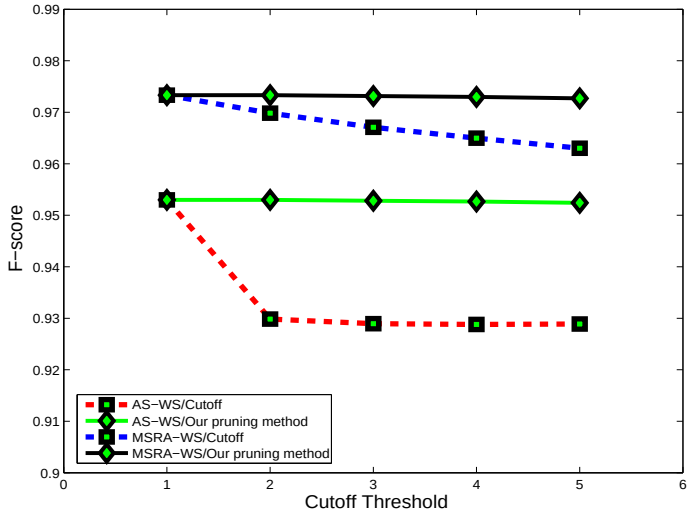
$$G_\lambda(f_H) = \max_{\lambda_H} L_{\lambda + f_H \lambda_H} - L_\lambda - (\lambda_H^2 / 2\sigma^2), \tag{8}$$

where L_λ is the conditional log-likelihood for training, and σ^2 is a Gaussian prior. In order to make the gain computations tractable, the likelihood is approximated by a pseudo likelihood. In feature selection, those feature candidates with highest gain are added into the optimal subset. Recently, boosting techniques are paid more and more attention and applied to CRFs training speedup [6, 7]. [6] proposes a method that simultaneously performs feature selection and parameter estimation for CRFs. In their formulation, to choose a good feature, a weighted least-square-error (WLSE) problem should be solved,

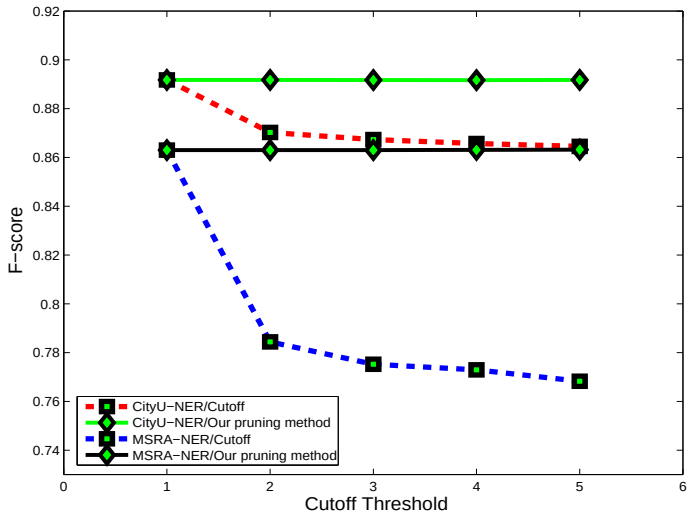
$$f_m(\mathbf{x}) = \text{argmin}_f \sum_{i=1}^{N} w_i E(f(\mathbf{x}_i) - z_i)^2, \tag{9}$$

where w_i and z_i are two parameters that can be computed as in LogitBoost algorithm. Our ranking metric is some similar to [6] in formulation though quite different from the latter. In addition, the feature candidate of the latter is implicitly derived rather than explicitly ranking all possible features according to a metric score.

CRFs learning is not often an easy computational job in many cases as we need to train larger and larger labeled data. Feature selection, namely, to find an optimal feature subset for CRFs is even harder task than CRFs training itself. For example, in [4], sophisticated techniques are used to make feature selection tractable in computation. Thus, we can regard pruning an existing model with millions of features more practical than feature selection task defined by [1] in current computational machine settings. Especially, we start our work based on



(a)



(b)

Fig. 2. Comparison of our pruning method and count cut-off method, (a) WS (b) NER

models obtained through training with carefully selected label set and feature template set by human observation, which is surely a tractable computational task.

Our results shows that a few features contribute a great deal to the performance and existing CRFs models that we have examined in this work at least contain quite an amount of redundancy even they can achieve state-of-the-art performance.

5 Conclusions and Future Work

We propose a posterior pruning method for CRFs model. CRFs Models for a real application may dramatically inflate as training data is enlarged. This study tries to alleviate this difficulty. Our idea is to remove those insignificant feature groups according to a proposed ranking metric. We carry out a series of experiments in two sequence labeling tasks, namely, sequence segmentation and named entity recognition, to verify the effectiveness of the proposed method. The results are quite positive. Our results show that CRFs models that are examined in this work are highly redundant, even using carefully selected label set and feature templates.

Compared to the existing, the proposed pruning method is efficient because only a local metric of each feature group needs to be computed before a sorting operation is performed. For higher performance and a more compact system, it is natural to consider combining our technique with existing ones, which mostly are forward feature selection methods and whose metrics are derived according to the observations in information theory[6]. Since this work requires a great deal of computational resources, we have to leave this as one of our future work. However, we may still expect the effectiveness of the proposed metric in these possible ensemble schemes, as it is motivated from a local structural factor of CRFs learning rather than global statistical information as most others.

Though we check the effectiveness of the proposed rank metric only for CRFs, its principle may be extended to other similar learning schemes such as MEMM. In fact, our early results have shown that it is also effective for these kinds of learning schemes.

Another issue about future work is that there are many other learning techniques that naturally produce sparse solutions such as some lazy-update algorithms. Typically, the structured averaged perceptron of [15] generally yields models with very few active features (usually between 1% and 10% active according to their report), since the parameters are updated only in the case that a training error occurs. Thus, one could simply train the perceptron and discard all features with zero weight, obtaining a model with identical behavior and far fewer features. However, this is beyond what we intend to study about CRFs model pruning in this work, and further comparison and technique ensemble will be also left as future work.

References

1. Lafferty, J. D., McCallum, A. and Pereira, F. C. N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, pages 282-289, San Francisco, CA, USA, 2001.
2. Rosenfeld, B., Feldman, R., and Fresko, M.: A systematic cross-comparison of sequence classifiers. In SDM 2006, pages 563-567, Bethesda, Maryland, 2006.

[6] Apparently, [6] should be an exception.

3. Sha, F., and Pereira, F.: Shallow parsing with conditional random fields. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, volume 1, pages 134-141, Edmonton, Canada, 2003.
4. McCallum, A.: Efficiently inducing features of conditional random fields. In Proceedings of the 19th Conference in Uncertainty in Articifical Intelligence (UAI-2003), Acapulco, Mexico, August 7-10, 2003.
5. Qi, Y., Szummer, M., and Minka, T. P.: Diagram structure recognition by bayesian conditional random fields. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pages 191-196, San Diego, CA, USA, June 20-25, 2005.
6. Liao, L., Choudhury, T., Fox, D. and Kautz, H.: Training conditional random fields using virtual evidence boosting. In The Twentieth International Joint Conferfence on Artificial Intelligence (IJCAI 2007), pages 2530-2535, Hyderabad, India, January 6 – 12, 2007.
7. Gutmann, B. and Kersting, K.: Stratified gradient boosting for fast training of confiditional random fields. In D. Malerba, A. Appice, and M. Ceci, editors, Proceedings of the 6th International Workshop on Multi-Relational Data Mining, pages 56-68, Warsaw, Poland, September 17, 2007.
8. Guyon, I. and Elisseeff A.: An introduction to variable and feature selection. Journal of Machine Learning Research, 3:1157-1182, 2003.
9. Peng, F., Feng, F. and McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In COLING 2004, pages 562-568, Geneva, Switzerland, August 23-27, 2004.
10. Zhao, H., Huang, C.-N., and Li, M.: An improved Chinese word segmentation system with conditional random field. In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pages 162-165, Sydney, Australia, July 22-23, 2006.
11. Zhao, H., Huang, C.-N., Li, M., and Lu, B.-L.: Effective tag set selection in Chinese word segmentation via conditional random field modeling. In Proceedings of the 20th Asian Pacific Conference on Language, Information and Computation, pages 87-94, Wuhan, China, November 1-3, 2006.
12. Zhao, H., and Kit, C.: Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In The Sixth SIGHAN Workshop on Chinese Language Processing, pages 106-111, Hyderabad, India, January 11-12, 2008.
13. Zhang, R., Kikui, G., and Sumita, E.: Subword-based tagging by conditional random fields for Chinese word segmentation. In Proceedings of Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2006), pages 193-196, New York, 2006.
14. Pietra, S. D., Pietra, V. D., and Lafferty, J.: Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:380 – 393, 1997.
15. Collins, M: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pages 1-8, University of Pennsylvania, Philadelphia, PA, USA, July 6-7 2002.