Domain Adaptation for Conditional Random Fields

Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Lide Wu

Department of Computer Science and Engineering, Fudan University
{qi_zhang,xpqiu,xjhuang,ldwu}@fudan.edu.cn

Abstract. Conditional Random Fields (CRFs) have received a great amount of attentions in many fields and achieved good results. However, a case frequently encountered in practice is that the test data's domain is different with the training data's. It would affect negatively the performance of CRFs. This paper presents a novel technique for maximum a posteriori (MAP) adaptation of Conditional Random Fields model. The background model, which is trained on data from a domain, could be well adapted to a new domain with a small number of labeled domain specific data. Experimental results on tasks of chunking and capitalizing show that this technique can significantly improve performance on out-of-domain data. In chunking task, the relative improvement given by the adaptation technique is 56.9%. With two in-domain sentences, it also can achieve 30.2% relative improvement.

1 Introduction

Conditional Random Fields (CRFs) are undirected graphical models that were developed for labeling relational data [1]. A CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. Therefore, the weights of different features at different states can be traded off against each other. CRFs modeling technique has received a great amount of attentions in many fields, such as part-of-speech tagging [1], shallow parsing [2], named entity recognition [3,4], bioinfomatics [5], Chinese word segmentation [6,7], and Information Extraction [8]. It achieves good results in them.

Similar to most of the classification algorithms, CRFs also have the assumption that training and test data are drawn from the same underlying distributions. However, a case frequently encountered in practice is that the test data is drawn from a distribution that is related but not identical with the training data's. For example, one may wish to use a POS tagger trained with WSJ corpus to label email or bioinformatics research papers. This typically affects negatively the performance of a given model. From the experimental results we can know that the performance of the chunker trained with WSJ corpus can achieve 96.2% in different part of WSJ corpus. While performance of the same chunker in BROWN corpus is only 88.4%.

In order to achieve better results in a specific domain, labeled in-domain data is needed. Although large scale in-domain labeled corpus is hard to get, a small number of in-domain labeled data(*adaptation data*) and a large number of domain related labeled data(*background data*) is easier to get. For example Penn Treebanks [9] can be used as background training data for POS tagging, chunking, parsing and so on. This

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 192–202, 2008.

kind of adaptation technique is used in many fields, such as language modelling [10], capitalization [11], automatic speech recognition [12], parsing [13,14] and so on.

Directly combining background and adaptation data together is a way to use the in-domain data. But if the scale of adaptation data is much smaller than the background data, the adaptation data's impact would be low. It can be seen from the experimental results. Another disadvantage of this method is that this technique need to retrain the whole model. It would waste a lot of time. In order to take advantage of the in-domain labeled data, a maximum a-posteriori (MAP) adaptation technique for Conditional Random Fields models is developed, following the similar idea with adaptation of Maximum Entropy [11]. The adaptation procedure proves to be quite effective in further improving the classification result on different domains. We evaluate the performance of this adaptation technique in chunking, capitalizing. The relative chunking's performance improvement of the adapted model over the background model is 56.9%. In capitalization task, the adapted model achieves 29.6% relative improvement.

The remainder of this paper is organized as follows: Section 2 describes the related works. The CRFs modeling technique is briefly reviewed in Section 3. Section 4 describes the MAP adaptation technique used for CRFs. The experimental results are presented in Section 5. Conclusions are presented in the last section.

2 Related Works

Leggetter and Woodland [12] introduced a method of speaker adaptation for continuous density Hidden Markov Models (HMMs). Adaptation statistics are gathered from the available adaptation data and used to calculate a linear regression-based transformation for the mean vectors.

Several recent papers also presented their works on modifying learning approaches-boosting [15], naive Bayes [16], and SVMs [17] - to use domain knowledge in text classification. Those methods all modify the base learning algorithm with manually converted knowledge about words.

Chelba and Acero [11] presented a technique for maximum a posteriori (MAP) adaptation of maximum entropy (MaxEnt) and maximum entropy Markov models (MEMM). The technique was applied to the problem of recovering the correct capitalization of uniformly cased text. Our work has similarities to Chelba and Acero's.

Daume and Marcu [18] presented a framework for domain adaptation problem. They treat the in-domain data as drawn from a mixture of "truly in-domain" distribution and a "general domain" distribution. Similarly, the out-of-domain are also drawn from a "truly out-of-domain" distribution and a "general domain" distribution. Then they apply EM method to estimate parameters. However, this framework used in CRF is computationally expensive.

3 Conditional Random Fields

Conditional Random Fields (CRFs) are undirected graphical models trained to maximize a conditional probability [1]). CRFs avoid a fundamental limitation of maximum entropy Markov models (MEMMs), which can be biased towards states with few successor states.

Let $X = x_1 ... x_n$ and $Y = y_1 ... y_n$ represent the generic input sequence and label sequence. The cliques of the graph are now restricted to include just pairs of states (y_{i-1}, y_i) that are neighbors in the sequence. Linear-chain CRFs thus define the conditional probability of a state sequence given an input sequence to be

$$P_{\Lambda}(Y|X) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{n} \sum_{k=1}^{m} \lambda_k f_k(y_{i-1}, y_i, x, i)\right)$$

where Z_x is a normalization factor over all state sequences, $f_k(y_{i-1}, y_i, x, i)$ is an arbitrary feature function over its arguments, and λ_k (ranging from $-\infty$ to ∞) is a learned weight for each feature function. A feature function is either a state feature $s(y_i, x, i)$ or a transition feature $t(y_{i-1}, y_i, x, i)$.

Then, the CRF's global feature vector for input sequence X and label sequence Y is given by

$$F(Y, X) = \sum_{i} f(y_{i-1}, y_i, x, i)$$

where i ranges over input positions. Using the global feature vector, $P_{\Lambda}(Y|X) = \frac{1}{Z_X} \exp(\Lambda \cdot F(Y, X))$. The most probable path $\hat{Y}$ for input sequence X is then given by

$$\hat{Y} = \arg \max_{Y \in Y(x)} P(Y|X) = \arg \max_{Y} \lambda \cdot F(Y, X)$$

which can be found by Viterbi algorithm.

3.1 Parameter Estimation

CRFs can be trained by the standard maximum likelihood estimation, i.e., maximizing the log-likelihood $\mathcal{L}_{\Lambda}$ of a given training set $T = \{< X_j, Y_j >\}_{j=1}^{N}$.

$$\hat{\Lambda} = \arg \max_{\Lambda \in \mathbb{R}^k} \mathcal{L}_{\Lambda},$$

where

$$\begin{aligned} \mathcal{L}_{\Lambda} &= \sum_{j} \log(P(Y_j|X_j)) \\ &= \sum_{j} \left[\Lambda \cdot F(Y_j, X_j) - \log(Z_{X_j})\right]. \end{aligned}$$

To perform the optimization, we seek the zero of the gradient

$$\begin{aligned} \frac{\partial \mathcal{L}_{\Lambda}}{\partial \lambda_k} &= \sum_{j} \left(F_k(Y_j, X_j) - E_{P(Y|X)}[F_k(Y, X_j)]\right) \\ &= O_k - E_k = 0, \end{aligned}$$

where $O_k = \sum_j F_k(Y_j, X_j)$ is the count of feature k observed in the training data T, and $E_k = E_{P(Y|X)}[F_k(Y, X_j)]$ is the expectation of feature k over the model distribution $P(Y|X)$ and T. The expectation can be efficiency calculated using a variant of the forward-backword algorithm.

$$E_{P(Y|X)}[F_k(Y,X)] = \sum_i \frac{\alpha_i (f_i * M_i) \beta_i^T}{Z_X}$$
$$Z_X = \alpha_n \cdot 1^T$$

where α_i and β_i are the forward and backward state-cost vectors defined by

$$\alpha_i = \begin{cases} \alpha_{i-1} M_i & 0 < i \leq n \\ 1 & i = 0 \end{cases}$$
$$\beta_{i-1}^T = \begin{cases} M_{i+1} \beta_{i+1}^T & 0 \leq i < n \\ 1 & i = n \end{cases}$$

To avoid over fitting, we also use Gaussian weight prior [19]:

$$\mathcal{L}_\lambda{}' = \sum_j \log(P(Y_j|X_j) - \frac{\|\lambda\|^2}{2\sigma^2} + const$$

with gradient

$$\nabla \mathcal{L}_\lambda{}' = O_k - E_k - \frac{\lambda}{\sigma^2}$$

The optimal solutions can be obtained by using traditional iterative scaling algorithms (e.g., IIS or GIS [20]) or quasi-Newton methods(e.g., L-BFGS [21]).

4 MAP Adaptation of Conditional Random Fields

The overview of adaptation stages is shown in Figure 1. A simple way to accomplish this is to use MAP adaptation using a prior distribution on the model parameters [11]. A Gaussian prior for the model parameters Λ has been previously used to smooth CRFs models. The prior has 0 mean and diagonal covariance: $\Lambda \sim \mathcal{N}(0, diag(\sigma_i^2))$. In the adaptation part, the prior distribution is centered at the parameter Λ^0 estimated from the background data:$\Lambda \sim \mathcal{N}(\Lambda^0, diag(\sigma_i^2))$. For the features generated only from the adaptation, the prior distribution is still centered at 0. In our experiments the variances were tied to $\sigma_i = \sigma$ whose value was determined by line search on development data drawn from the background data or adaptation data.

Different from the Chelba and Acero's method [11], we use both σ_a and σ_m here. In their method, σ is used not only to balance the background and adaptation data, but also to represent the variance of the adaptation data. However they are different in most of circumstance. In order to overcome this problem we use two σ in adaptation step.

The log-likelihood $\mathcal{L}_\Lambda$ of the given adaptation data set becomes:

$$\mathcal{L}_\lambda{}' = \sum_j \log(P(y_j|x_j) - \sum_{i=1}^{F_{background}} \frac{\|\lambda_i - \lambda_i^0\|^2}{2\sigma_m^2} - \sum_{i=1}^{F_{adaptation}} \frac{\|\lambda_i\|^2}{2\sigma_a^2}$$

Therefore the gradient becomes:

$$\nabla\mathcal{L}_\lambda{}' = O_k - E_k - \sum_{i=1}^{F_{background}} \frac{\lambda_i - \lambda_i^0}{\sigma_m^2} - \sum_{i=1}^{F_{adaptation}} \frac{\lambda_i}{\sigma_a^2},$$

where $F_{background}$ is the features generated from the background data, $F_{adaptation}$ is the features generated only from the adaptation data, σ_a represents the variance of the adaptation data, and σ_m is used to balance the background and adaptation model. A small variance σ_m will keep the weight λ_m close to the background model, while a large variance σ_m will make the model sensitive to adaptation data. With $\mathcal{L}_\lambda{}'$ and $\nabla\mathcal{L}_\lambda{}'$, λ can be iteratively calculated through L-BFGS.

```
Algorithm MAP Adaptation of CRFs
$\mathcal{F}_{background}$ = Feature set generated from background data
$\mathcal{F}_{adaptation}$ = Feature set generated from adaptation data
$\lambda_i$ = $f_i$'s corresponding weight

    Generate $\mathcal{F}_{background}$ from background data

    Estimate $\lambda_i^0$ for $\mathcal{F}_{background}$

    Generate $\mathcal{F}_{adaptation}$ from adaptation data

    Let $\mathcal{F} = \mathcal{F}_{background} \bigcup \mathcal{F}_{adaptation}$

    Let $\lambda_i = \lambda_i^0$ if $f_i \in \mathcal{F}_{background}$
        $\lambda_i = 0$, otherwise

    Estimate $\lambda_i$ with equation $\mathcal{L}_\lambda{}'$ and $\nabla\mathcal{L}_\lambda{}'$
```

Fig. 1. Algorithm of MAP Adaptation of CRFs

Table 1. Feature templates used by Chunker

type	template
Base features	w_{-2}, w_{-1}, w_0,w_1, w_2 p_{-2}, p_{-1}, p_0,p_1, p_2
Bi-gram features	$w_{-2}w_{-1}$, $w_{-1}w_0$, w_0w_1,w_1w_2 $p_{-2}p_{-1}$, $p_{-1}p_0$, p_0p_1,p_1p_2 $p_{-2}w_{-1}$, $p_{-1}w_0$, p_0w_1,p_1w_2 $w_{-2}p_{-1}$, $w_{-1}p_0$, w_0p_1,w_1p_2
Tri-gram features	$w_{-2}w_{-1}w_0$, $w_{-1}w_0w_1$, $w_0w_1w_2$ $p_{-2}p_{-1}p_0$, $p_{-1}p_0p_1$, $p_0p_1p_2$ $p_{-2}p_{-1}w_0$, $p_{-1}w_0p_1$, $p_0w_1p_2$ $w_{-2}w_{-1}p_0$, $w_{-1}p_0w_1$, $w_0p_1w_2$

w_0 is the word at current position, w_1 is the word instant after w_0, w_{-1} is the word instant before it, p_* represents word's POS tags.

5 Experiments

To evaluate the MAP adaptation of CRFs, we did several experiments on chunking and capitalizing. Penn Treebanks III [9] is used to train chunker. Capitalizer's training data comes from Tipster corpus [22]. We will introduce the detail steps and features used in the following parts.

5.1 Experiments on Chunking

The goal of chunking is to group sequences of words together and classify them by syntactic labels. Various NLP tasks can be seen as a chunking task, such as English base noun phrase identification (base NP chunking), English base phrase identification (chunking), and so on. Because chunking technique is used in many different fields, we choose chunking task to evaluate the adaptation methods.

The background data used for chunker is generated from WSJ data(wsj_0200.mrg - wsj_2172.mrg). The in-domain test data is from wsj_0000.mrg to wsj_0199.mrg. The others are used to tune parameters. Bracketed representation is converted into IOB2 representation [23,24].

For adaptation experiments we use BROWN data in Penn Treebanks III. As Brown Corpus dataset contains eight types of articles, we extract one article from each type(C* _01.mrg), which are used as adaptation data. The second articles from each type(C* _02. mrg) are used as development data. The others are used for evaluations.

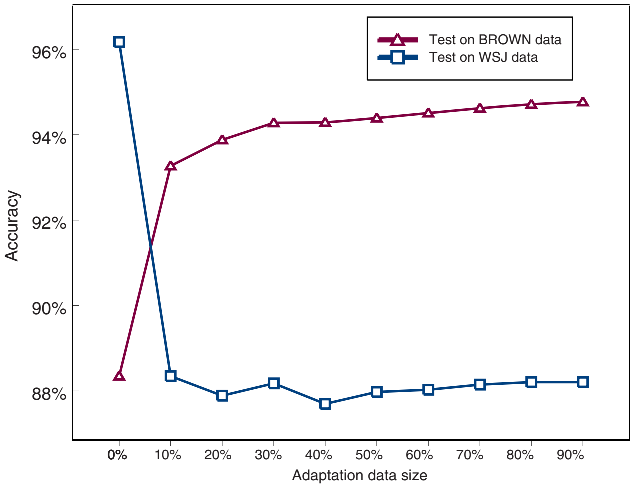

Fig. 2. The impact of the adaptation data's size

Table 2. Chunking Results on in-domain(WSJ) and out-of-domain data(BROWN)

Background data	Adaptation data	Evaluation data	Accuracy
WSJ	NONE	B-tst	88.4%
WSJ	B-ada	B-tst	**95.0%**
WSJ	NONE	wsj-tst	**96.2%**
WSJ	B-ada	wsj-tst	88.4%

where "wsj-tst" represents the test part of WSJ, "B-tst" represents the test part of BROWN.

The templates used in chunking experiments are shown is Table 1.

Results of both in-domain and out-of-domain are shown in Table 2. The σ^2 used in background model is selected by in-domain development data. σ_a^2, and σ_m^2 are selected by development data extracted from BROWN data. From the result we observe that the performance of background model in in-domain data is significantly better than in out-of-domain data. Adaptation improves the performance on Brown data by 56.9% relative.

Figure 2 shows the result of the impact of the adaptation data's size. X axis represents the percentage of the adaptation data in BROWN corpus(B-ada). Y axis represents the accuracy. Two lines represent the results of test data set on BROWN (B-tst) and WSJ (wsj-tst) corpus. The result in 0% is got by the background model. The result in 10% is got by the model adapted by 10 percents B-ada data. We observe from the result that the larger adaptation data are used the higher accuracy in this domain could be get. When the size of the adaptation data is very small, this technique can also achieve good result. We use two sentences extracted from B-ada data to adapt the background model. The adapted model also achieves 30.2% relative improvement.

Then we evaluate the impact of σ_m^2 to the performance. The result is shown in Figure 3. X axis represents σ_m^2. Y axis represents the accuracy. As expected low values of

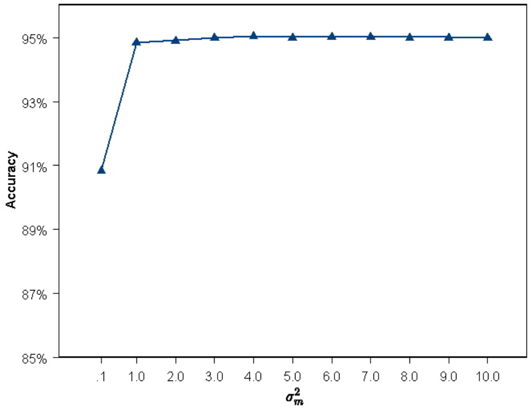

Fig. 3. The impact of the σ_m^2

σ_m^2 result little adaptation. When the σ_m^2 is between 1 and 10, the accuracy does have significant changes. Therefore this parameter can be easily set in the real system.

5.2 Experiments on Capitalizing

Capitalization can be converted to sequence tagging problem. Each low case words receive a tag which represents its capitalization form. It's also domain dependent. For example, in bioinformatics domain "gene" is almost low case form. It represents a concept. While in some domains, "gene" is usually capitalized, which represents a human name. Therefor we did some experiments to show the impact of model adaptation technique on this task.

The TIPSTER copra are used to generate both background and adaptation data for the capitalizer. The background data is WSJ data from 1987 - files from WSJ7_001 to WSJ7_127 in TIPSTER Phrase I. The in-domain test data is WSJ_0402 and WSJ_0403, which belong to WSJ 1990 in TIPSTER Phrase II. WSJ_0404 and WSJ_0405 are in-domain development data. The out-of-domain adaptation data is the combination of AP880212 and AP880213, which belong Associated Press 1988 in TIPSTER Phrase II. Files AP880214 and AP880215 are out-of-domain test data.

We use the same tag set with the set used in [11]. Each word in a sentence is labeled with one of the tags:

- LOC lowercase
- CAP capitalized
- MXC mixed case; no further guess is made as to the capitalization of such words.
- AUC all upper case
- PNC punctuation;

The feature templates we used are shown in table 3.

Table 4 shows results of in-domain and out-of-domain data. The σ^2 in background model we use in this experiment is 5, which is selected by development data. The σ_a^2

Table 3. Feature templates used by Capitalizer

type	template
Base features	w_{-1}, w_0,w_1
Bi-gram features	$w_{-1}w_0$, w_0w_1

used in adaptation part is set to 5. The σ_m^2 is 10. We can get the same trend with chunking's results. The adapted model gives 29.6% relative improvement. The size of adaptation data is less than 1% of the background WSJ data's size.

Table 4. Capitalizing Results on in-domain and out-of-domain data

Background data	Adaptation data	Evaluation data	Accuracy
WSJ	NONE	AP-tst	94.6%
WSJ	AP-ada	AP-tst	**96.2%**
WSJ	NONE	wsj-tst	**96.8%**
WSJ	AP-ada	wsj-tst	96.4%
WSJ+AP-ada	NONE	AP-tst	94.7%
WSJ+AP-ada	NONE	wsj-tst	96.8%

where “wsj-tst” represents the test part of WSJ,“AP-ada” represents the adaptation data, “AP-tst” represents the test part of Associated Press.

Then we combine adaptation data(AP-ada) with the background data(WSJ) and train a capitalizer with it. The accuracy of capitalizing wsj-tst is 96.8%. In AP-tst data, the accuracy is 94.7%. Comparing with the results got by background model, the capitalizer trained by combined data couldn't significantly improve the performance.

6 Conclusions

In this paper we present a novel technique for maximum a posteriori (MAP) adaptation of Conditional Random Fields Model. Through experimental results,we observe that this technique can effectively adapt a background model to a new domain with a small amount of domain specific labeled data. We did several experiments in three different fields: chunking and capitalizing. The relative chunking's performance improvement of the adapted model over the background model is 56.9%. With two in-domain sentences, it also can achieve 30.2% relative improvement. The relative improvement of capitalizing experiment is 29.6%. The experimental results prove that the MAP adaptation of Conditional Random Fields Model technique can benefit the performances in different tasks.

Acknowledgements

This work was partially supported by Chinese NSF 60435020 and 60673038. The authors would like to thank Bingqing Wang and Lin Zhao for their carefully proofreading. Thanks the anonymous reviewers for their valuable comments and suggestions.

References

1. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning (2001)
2. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proceedings of Human Language Technology-NAACL 2003 (2003)
3. Carreras, X., Márquez, L., Padró, L.: Learning a Perceptron-Based Named Entity Chunker via Online Recognition Feedback. In: Association with HLT-NAACL 2003 (2003)
4. Okanohara, D., Miyao, Y., Tsuruoka, Y., Tsujii, J.: Improving the scalability of semi-markov conditional random fields for named entity recognition. In: Proceedings of COLING-ACL 2006 (2006)
5. Settles, B.: Biomedical named entity recognition using conditional random fields and rich feature sets. In: COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP 2004) (2004)
6. Peng, F., Feng, F., McCallum, A.: Chinese Segmentation and New Word Detection using Conditional Random Fields. In: Proceedings of COLING 2004 (2004)
7. Feng, Y., Sun, L., Lv, Y.: Chinese word segmentation and named entity recognition based on conditional random fields models. In: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (2006)
8. Peng, F., McCallum, A.: Accurate information extraction from research papers using conditional random fields. In: HLT-NAACL 2004: Main Proceedings (2004)
9. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics 19, 313–330 (1993)
10. Clarkson, P., Robinson, A.J.: Language model adaptation using mixtures and an exponentially decaying cache. In: Proc. ICASSP 1997 (1997)
11. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: Little data can help a lot. In: Proceedings of EMNLP 2004 (2004)
12. Leggetter, C., Woodland, P.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. Journal of Computer Speech and Language (1995)
13. McClosky, D., Charniak, E., Johnson, M.: Reranking and self-training for parser adaptation. In: Proceedings of COLING-ACL 2006 (2006)
14. Lease, M., Charniak, E.: Parsing biomedical literature. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, Springer, Heidelberg (2005)
15. Schapire, R.E., Rochery, M., Rahim, M.G., Gupta, N.: Incorporating prior knowledge into boosting. In: Proceedings of the ICML 2002 (2002)
16. Liu, B., Li, X., Lee, W.S., Yu, P.S.: Text classification by labeling words. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence (2005)
17. Wu, X., Srihari, R.: Incorporating prior knowledge with weighted margin support vector machines. In: Proceedings of the tenth ACM SIGKDD (2004)
18. Daumé III, H., Marcu, D.: Domain Adaptation for Statistical Classifiers. Journal of Artificial Intelligence Research (2006)
19. Chen, S.F., Rosenfeld, R.: A gaussian prior for smoothing. maximum entropy models. Technical Report CMU-CS-99-108 (1999)
20. Della-Pietra, S., Della-Pietra, V., Lafferty, J.: Inducing features of random fields. IEEE Transactions on PAMI (1997)

21. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical Programming
22. Harman, D., Liberman, M.: Tipster complete. In: Linguistic Data Consortium catalog number LDC93T3A and ISBN: 1-58563-020-9 (1993), http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T3A
23. Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In: Proceedings of the Third Workshop on Very Large Corpora, Somerset, New Jersey, pp. 82–94 (1995)
24. Sang, E.F.T.K., Veenstra, J.: Representing Text Chunks