Exploiting Domain Structure for Named Entity Recognition

Jing Jiang and **ChengXiang Zhai**
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{jiang4,czhai}@cs.uiuc.edu

Abstract

Named Entity Recognition (NER) is a fundamental task in text mining and natural language understanding. Current approaches to NER (mostly based on supervised learning) perform well on domains similar to the training domain, but they tend to adapt poorly to slightly different domains. We present several strategies for exploiting the domain structure in the training data to learn a more robust named entity recognizer that can perform well on a new domain. First, we propose a simple yet effective way to automatically rank features based on their generalizabilities across domains. We then train a classifier with strong emphasis on the most generalizable features. This emphasis is imposed by putting a rank-based prior on a logistic regression model. We further propose a domain-aware cross validation strategy to help choose an appropriate parameter for the rank-based prior. We evaluated the proposed method with a task of recognizing named entities (genes) in biology text involving three species. The experiment results show that the new domain-aware approach outperforms a state-of-the-art baseline method in adapting to new domains, especially when there is a great difference between the new domain and the training domain.

1 Introduction

Named Entity Recognition (NER) is the task of identifying and classifying phrases that denote certain types of *named entities* (NEs), such as persons, organizations and locations in news articles, and genes, proteins and chemicals in biomedical literature. NER is a fundamental task in many natural language processing applications, such as question answering, machine translation, text mining, and information retrieval (Srihari and Li, 1999; Huang and Vogel, 2002).

Existing approaches to NER are mostly based on supervised learning. They can often achieve high accuracy provided that a large annotated training set *similar* to the test data is available (Borthwick, 1999; Zhou and Su, 2002; Florian et al., 2003; Klein et al., 2003; Finkel et al., 2005). Unfortunately, when the test data has some difference from the training data, these approaches tend to not perform well. For example, Ciaramita and Altun (2005) reported a performance degradation of a named entity recognizer trained on CoNLL 2003 Reuters corpus, where the F1 measure dropped from 0.908 when tested on a similar Reuters set to 0.643 when tested on a Wall Street Journal set. The degradation can be expected to be worse if the training data and the test data are more different.

The performance degradation indicates that existing approaches adapt poorly to new domains. We believe one reason for this poor adaptability is that these approaches have not considered the fact that, depending on the genre or domain of the text, the entities to be recognized may have different mor-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 74–81,
New York, June 2006.

phological properties or occur in different contexts. Indeed, since most existing learning-based NER approaches explore a large feature space, without regularization, a learned NE recognizer can easily overfit the training domain.

Domain overfitting is a serious problem in NER because we often need to tag entities in completely new domains. Given any new test domain, it is generally quite expensive to obtain a large amount of labeled entity examples in that domain. As a result, in many real applications, we must train on data that do not fully resemble the test data.

This problem is especially serious in recognizing entities, in particular gene names, from biomedical literature. Gene names of one species can be quite different from those of another species syntactically due to their different naming conventions. For example, some biological species such as yeast use symbolic gene names like *tL(CAA)G3*, while some other species such as fly use descriptive gene names like *wingless*.

In this paper, we present several strategies for exploiting the domain structure in the training data to learn a more robust named entity recognizer that can perform well on a new domain. Our work is motivated by the fact that in many real applications, the training data available to us naturally falls into several domains that are similar in some aspects but different in others. For example, in biomedical literature, the training data can be naturally grouped by the biological species being discussed, while for news articles, the training data can be divided by the genre, the time, or the news agency of the articles. Our main idea is to exploit such domain structure in the training data to identify generalizable features which, presumably, are more useful for recognizing named entities in a new domain. Indeed, named entities across different domains often share certain common features, and it is these common features that are suitable for adaptation to new domains; features that only work for a particular domain would not be as useful as those working for multiple domains. In biomedical literature, for example, surrounding words such as *expression* and *encode* are strong indicators of gene mentions, regardless of the specific biological species being discussed, whereas species-specific name characteristics (e.g., prefix = "*-less*") would clearly not generalize well, and may even hurt the performance on a new domain. Similarly, in news articles, the part-of-speeches of surrounding words such as *"followed by a verb"* are more generalizable indicators of name mentions than capitalization, which might be misleading if the genre of the new domain is different; an extreme case is when every letter in the new domain is capitalized.

Based on these intuitions, we regard a feature as *generalizable* if it is useful for NER in *all* training domains, and propose a generalizability-based feature ranking method, in which we first rank the features within each training domain, and then combine the rankings to promote the features that are ranked high in all domains. We further propose a rank-based prior on logistic regression models, which puts more emphasis on the more generalizable features during the learning stage in a principled way. Finally, we present a domain-aware validation strategy for setting an appropriate parameter value for the rank-based prior. We evaluated our method on a biomedical literature data set with annotated gene names from three species, fly, mouse, and yeast, by treating one species as the new domain and the other two as the training domains. The experiment results show that the proposed method outperforms a baseline method that represents the state-of-the-art NER techniques.

The rest of the paper is organized as follows: In Section 2, we introduce a feature ranking method based on the generalizability of features across domains. In Section 3, we briefly introduce the logistic regression models for NER. We then propose a rank-based prior on logistic regression models and describe the domain-aware validation strategy in Section 4. The experiment results are presented in Section 5. Finally we discuss related work in Section 6 and conclude our work in Section 7.

2 Generalizability-Based Feature Ranking

We take a commonly used approach and treat NER as a sequential tagging problem (Borthwick, 1999; Zhou and Su, 2002; Finkel et al., 2005). Each token is assigned the tag I if it is part of an NE and the tag O otherwise. Let $\boldsymbol{x}$ denote the feature vector for a token, and let y denote the tag for $\boldsymbol{x}$. We first compute the probability $p(y|\boldsymbol{x})$ for each token, using a

learned classifier. We then apply Viterbi algorithm to assign the most likely tag sequence to a sequence of tokens, i.e., a sentence. The features we use follow the common practice in NER, including surface word features, orthographic features, POS tags, substrings, and contextual features in a local window of size 5 around the target token (Finkel et al., 2005).

As in any learning problem, feature selection may affect the NER performance significantly. Indeed, a very likely cause of the domain overfitting problem may be that the learned NE recognizer has picked up some non-generalizable features, which are not useful for a new domain. Below, we present a generalizability-based feature ranking method, which favors more generalizable features.

Formally, we assume that the training examples are divided into m subsets $T_1, T_2, \ldots, T_m$, corresponding to m different domains $D_1, D_2, \ldots, D_m$. We further assume that the test set T_{m+1} is from a new domain D_{m+1}, and this new domain shares some common features of the m training domains. Note that these are reasonable assumptions that reflect the situation in real problems.

We use *generalizability* to denote the amount of contribution a feature can make to the classification accuracy on any domain. Thus, a feature with high generalizability should be useful for classification on any domain. To identify the highly generalizable features, we must then compare their contributions to classification among different domains.

Suppose in each individual domain, the features can be ranked by their contributions to the classification accuracy. There are different feature ranking methods based on different criteria. Without loss of generality, let us use $r_T : F \rightarrow \{1, 2, \ldots, |F|\}$ to denote a ranking function that maps a feature $f \in F$ to a rank $r_T(f)$ based on a set of training examples T, where F is the set of all features, and the rank denotes the position of the feature in the final ranked list. The smaller the rank $r_T(f)$ is, the more important the feature f is in the training set T. For the m training domains, we thus have m ranking functions $r_{T_1}, r_{T_2}, \ldots, r_{T_m}$.

To identify the generalizable features across the m different domains, we propose to combine the m individual domain ranking functions in the following way. The idea is to give high ranks to features that are useful in all training domains . To achieve this goal, we first define a scoring function $s : F \rightarrow \mathbb{R}$ as follows:

$$s(f) = \min_{i=1}^{m} \frac{1}{r_{T_i}(f)}. \quad (1)$$

We then rank the features in decreasing order of their scores using the above scoring function. This is essentially to rank features according to their maximum rank $\max_i r_{T_i}(f)$ among the m domains. Let function r_{gen} return the rank of a feature in this combined, generalizability-based ranked list.

The original ranking function r_T used for individual domain feature ranking can use different criteria such as information gain or χ^2 statistic (Yang and Pedersen, 1997). In our experiments, we used a ranking function based on the model parameters of the classifier, which we will explain in Section 5.2.

Next, we need to incorporate this preference for generalizable features into the classifier. Note that because this generalizability-based feature ranking method is independent of the learning algorithm, it can be applied on top of any classifier. In this work, we choose the logistic regression classifier. One way to incorporate the feature ranking into the classifier is to select the top-k features, where k is chosen by cross validation. There are two potential problems with this *hard* feature selection approach. First, once k features are selected, they are treated equally during the learning stage, resulting in a loss of the preference among these k features. Second, this incremental feature selection approach does not consider the correlation among features. We propose an alternative way to incorporate the feature ranking into the classifier, where the preference for generalizable features is transformed into a non-uniform prior over the feature parameters in the model. This can be regarded as a *soft* feature selection approach.

3 Logistic Regression for NER

In binary logistic regression models, the probability of an observation $\boldsymbol{x}$ being classified as I is

$$p(I|\boldsymbol{x}, \boldsymbol{\beta}) = \frac{\exp(\beta_0 + \sum_{i=1}^{|F|} \beta_i x_i)}{1 + \exp(\beta_0 + \sum_{i=1}^{|F|} \beta_i x_i)} \quad (2)$$

$$= \frac{\exp(\boldsymbol{\beta} \cdot \boldsymbol{x'})}{1 + \exp(\boldsymbol{\beta} \cdot \boldsymbol{x'})}, \quad (3)$$

where β_0 is the bias weight, β_i ($1 \leq i \leq |F|$) are the weights for the features, and $\boldsymbol{x'}$ is the augmented feature vector with $x_0 = 1$. The weight vector $\boldsymbol{\beta}$ can be learned from the training examples by a maximum likelihood estimator. It is worth pointing out that logistic regression has a close relation with maximum entropy models. Indeed, when the features in a maximum entropy model are defined as conjunctions of a feature on observations only and a Kronecker delta of a class label, which is a common practice in NER, the maximum entropy model is equivalent to a logistic regression model (Finkel et al., 2005). Thus the logistic regression method we use for NER is essentially the same as the maximum entropy models used for NER in previous work.

To avoid overfitting, a zero mean Gaussian prior on the weights is usually used (Chen and Rosenfeld, 1999; Bender et al., 2003), and a maximum a posterior (MAP) estimator is used to maximize the posterior probability:

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} p(\boldsymbol{\beta}) \prod_{j=1}^{N} p(y_j|\boldsymbol{x_j}, \boldsymbol{\beta}), \tag{4}$$

where y_j is the true class label for $\boldsymbol{x_j}$, N is the number of training examples, and

$$p(\boldsymbol{\beta}) = \prod_{i=1}^{|F|} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{\beta_i^2}{2\sigma_i^2}). \tag{5}$$

In previous work, σ_i are set uniformly to the same value for all features, because there is in general no additional prior knowledge about the features.

4 Rank-Based Prior

Instead of using the same σ_i for all features, we propose a rank-based non-uniform Gaussian prior on the weights of the features so that more generalizable features get higher prior variances (i.e., low prior strength) and features on the bottom of the list get low prior variances (i.e., high prior strength). Since the prior has a zero mean, such a prior would force features on the bottom of the ranked list, which have the least generalizability, to have near-zero weights, but allow more generalizable features to be assigned higher weights during the training stage.

4.1 Transformation Function

We need to find a transformation function $h : \{1, 2, \ldots, |F|\} \rightarrow \mathbb{R}^+$ so that we can set $\sigma_i^2 = h(r_{\text{gen}}(f_i))$, where $r_{\text{gen}}(f_i)$ is the rank of feature f_i in the generalizability-based ranked feature list, as defined in Section 2. We choose the following h function because it has the desired properties as described above:

$$h(r) = \frac{a}{r^{1/b}}, \tag{6}$$

where a and b ($a, b > 0$) are parameters that control the degree of the confidence in the generalizability-based ranked feature list. Note that a corresponds to the prior variance assigned to the top-most feature in the ranked list. When b is small, the prior variance drops rapidly as the rank r increases, giving only a small number of top features high prior variances. When b is larger, there will be less discrimination among the features. When b approaches infinity, the prior becomes a uniform prior with the variance set to a for all features. If we set a small threshold τ on the variance, then we can derive that at least $m = \left(\frac{a}{\tau}\right)^b$ features have a prior variance greater than τ. Thus b is proportional to the logarithm of the number of features that are assigned a variance greater than the threshold τ when a is fixed. Figure 1 shows the h function when a is set to 20 and b is set to a set of different values.

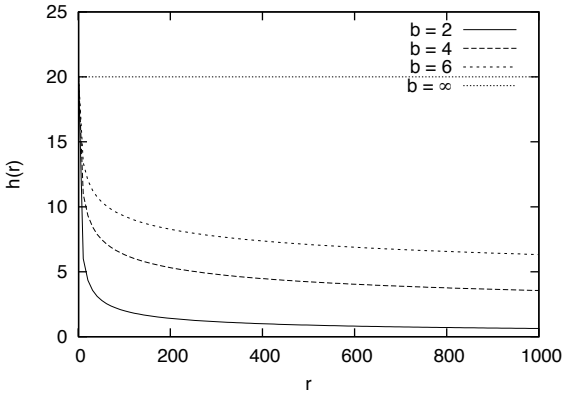

Figure 1: Transformation Function $h(r) = \frac{20}{r^{1/b}}$

4.2 Parameter Setting using Domain-Aware Validation

We need to set the appropriate values for the parameters a and b. For parameter a, we use the following

simple strategy to obtain an estimation. We first train a logistic regression model on all the training data using a Gaussian prior with a fixed variance (set to 1 in our experiments). We then find the maximum weight

$$\beta_{\max} = \max_{i=1}^{|F|} |\beta_i| \quad (7)$$

in this trained model. Finally we set $a = \beta_{\max}^2$. Our reasoning is that since a is the variance of the prior for the best feature, a is related to the "permissible range" of β for the best feature, and $\beta_{\max}$ gives us a way for adjusting a according to the empirical range of β_i's.

As we pointed out in Section 4.1, when a is fixed, parameter b controls the number of top features that are given a relatively high prior variance, and hence implicitly controls the number of top features to choose for the classifier to put the most weights on. To select an appropriate value of b, we can use a held-out validation set to tune the parameter value b. Here we present a validation strategy that exploits the domain structure in the training data to set the parameter b for a new domain. Note that in regular validation, both the training set and the validation set contain examples from all training domains. As a result, the average performance on the validation set may be dominated by domains in which the NEs are easy to classify. Since our goal is to build a classifier that performs well on new domains, we should pay more attention to hard domains that have lower classification accuracy. We should therefore examine the performance of the classifier on each training domain individually in the validation stage to gain an insight into the appropriate value of b for a new domain, which has an equal chance of being similar to any of the training domains.

Our domain-aware validation strategy first finds the optimal value of b for each training domain. For each subset T_i of the training data belonging to domain D_i, we divide it into a training set T_i^t and a validation set T_i^v. Then for each domain D_i, we train a classifier on the training sets of all domains, that is, we train on $\bigcup_{j=1}^m T_j^t$. We then test the classifier on T_i^v. We try a set of different values of b with a fixed value of a, and choose the optimal b that gives the best performance on T_i^v. Let this optimal value of b for domain D_i be b_i.

Given b_i $(1 \leq i \leq m)$, we can choose an appropriate value of b_{m+1} for an unknown test domain D_{m+1} based on the assumption that D_{m+1} is a mixture of all the training domains. b_{m+1} is then chosen to be a weighted average of b_i, $(1 \leq i \leq m)$:

$$b_{m+1} = \sum_{i=1}^{m} \lambda_i b_i, \quad (8)$$

where λ_i indicates how similar D_{m+1} is to D_i. In many cases, the test domain D_{m+1} is completely unknown. In this case, the best we can do is to set $\lambda_i = 1/m$ for all i, that is, to assume that D_{m+1} is an even mixture of all training domains.

5 Empirical Evaluation

5.1 Experimental Setup

We evaluated our domain-aware approach to NER on the problem of gene recognition in biomedical literature. The data we used is from BioCreAtIvE Task 1B (Hirschman et al., 2005). We chose this data set because it contains three subsets of MEDLINE abstracts with gene names from three species (fly, mouse, and yeast), while no other existing annotated NER data set has such explicit domain structure. The original BioCreAtIvE 1B data was not provided with every gene annotated, but for each abstract, a list of genes that were mentioned in the abstract was given. A gene synonym list was also given for each species. We used a simple string matching method with slight relaxation to tag the gene mentions in the abstracts. We took 7500 sentences from each species for our experiments, where half of the sentences contain gene mentions. We further split the 7500 sentences of each species into two sets, 5000 for training and 2500 for testing.

We conducted three sets of experiments, each combining the 5000-sentence training data of two species as training data, and the 2500-sentence test data of the third species as test data. The 2500-sentence test data of the training species was used for validation. We call these three sets of experiments *F+M⇒Y*, *F+Y⇒M*, and *M+Y⇒F*.

we use FEX[1] for feature extraction and BBR[2] for logistic regression in our experiments.

[1] http://l2r.cs.uiuc.edu/ cogcomp/asoftware.php?skey=FEX
[2] http://www.stat.rutgers.edu/ madigan/BBR/

5.2 Comparison with Baseline Method

Because the data set was generated by our automatic tagging procedure using the given gene lists, there is no previously reported performance on this data set for us to compare with. Therefore, to see whether using the domain structure in the training data can really help the adaptation to new domains, we compared our method with a state-of-the-art baseline method based on logistic regression. It uses a Gaussian prior with zero mean and uniform variance on all model parameters. It also employs 5-fold regular cross validation to pick the optimal variance for the prior. Regular feature selection is also considered in the baseline method, where the features are first ranked according to some criterion, and then cross validation is used to select the top-k features. We tested three popular regular feature ranking methods: *feature frequency* (F), *information gain* (IG), and χ^2 *statistic* (CHI). These methods were discussed in (Yang and Pedersen, 1997). However, with any of the three feature ranking criteria, cross validation showed that selecting all features gave the best average validation performance. Therefore, the best baseline method which we compare our method with uses all features. We call the baseline method *BL*.

In our method, the generalizability-based feature ranking requires a first step of feature ranking within each training domain. While we could also use F, IG or CHI to rank features in each domain, to make our method self-contained, we used the following strategy. We first train a logistic regression model on each domain using a zero-mean Gaussian prior with variance set to 1. Then, features are ranked in decreasing order of the absolute values of their weights. The rationale is that, in general, features with higher weights in the logistic regression model are more important. With this ranking within each training domain, we then use the generalizability-based feature ranking method to combine the m domain-specific rankings. The obtained ranked feature list is used to construct the rank-based prior, where the parameters a and b are set in the way as discussed in Section 4.2. We call our method *DOM*.

In Table 1, we show the precision, recall, and F1 measures of our domain-aware method (*DOM*) and the baseline method (*BL*) in all three sets of experiments. We see that the domain-aware method outperforms the baseline method in all three cases when F1 is used as the primary performance measure. In *F+Y⇒M* and *M+Y⇒F*, both precision and recall are also improved over the baseline method.

Exp	Method	P	R	F1
F+M⇒Y	BL	0.557	0.466	0.508
	DOM	0.575	0.516	**0.544**
F+Y⇒M	BL	0.571	0.335	0.422
	DOM	0.582	0.381	**0.461**
M+Y⇒F	BL	0.583	0.097	0.166
	DOM	0.591	0.139	**0.225**

Table 1: Comparison of the domain-aware method and the baseline method, where in the domain-aware method, $b = 0.5b_1 + 0.5b_2$

Note that the absolute performance shown in Table 1 is lower than the state-of-the-art performance of gene recognition (Finkel et al., 2005).[3] One reason is that we explicitly excluded the test domain from the training data, while most previous work on gene recognition was conducted on a test set drawn from the same collection as the training data. Another reason is that we used simple string matching to generate the data set, which introduced noise to the data because gene names often have irregular lexical variants.

5.3 Comparison with Regular Feature Ranking Methods

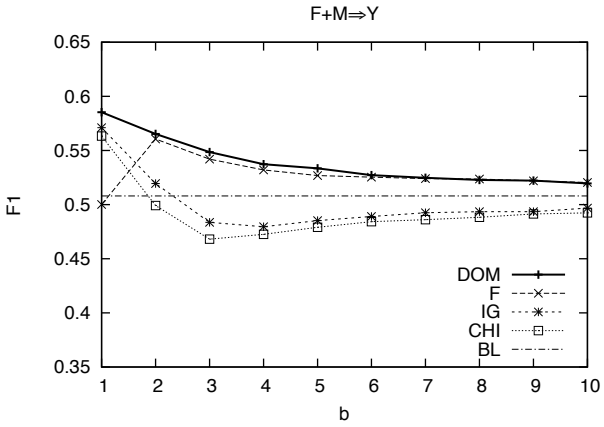

Figure 2: Comparison between regular feature ranking and generalizability-based feature ranking on *F+M⇒Y*

[3] Our baseline method performed comparably to the state-of-the-art systems on the standard BioCreAtIvE 1A data.

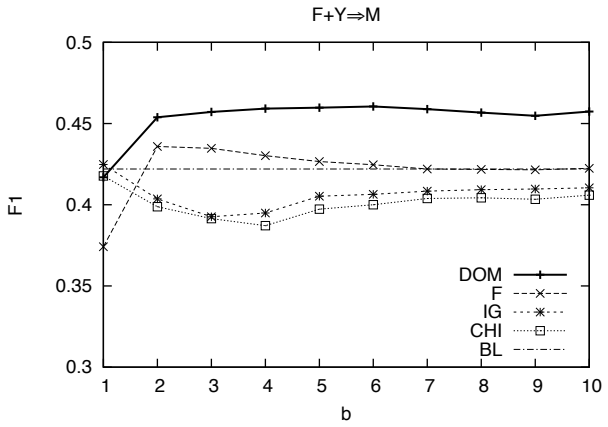

Figure 3: Comparison between regular feature ranking and generalizability-based feature ranking on $F+Y \Rightarrow M$

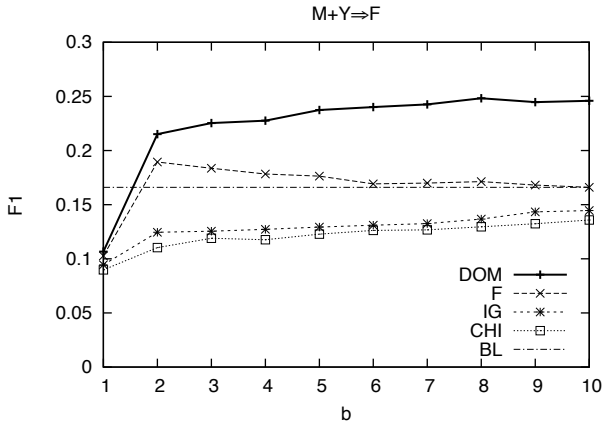

Figure 4: Comparison between regular feature ranking and generalizability-based feature ranking on $M+Y \Rightarrow F$

To further understand how our method improved the performance, we compared the generalizability-based feature ranking method with the three regular feature ranking methods, F, IG, and CHI, that were used in the baseline method. To make fair comparison, for the regular feature ranking methods, we also used the rank-based prior transformation as described in Section 4 to incorporate the preference for top-ranked features. Figure 2, Figure 3 and Figure 4 show the performance of different feature ranking methods in the three sets of experiments as the parameter b for the rank-based prior changes. As we pointed out in Section 4, b is proportional to the logarithm of the number of "effective features".

From the figures, we clearly see that the curve for the generalizability-based ranking method DOM is always above the curves of the other methods, indicating that when the same amount of top features are being emphasized by the prior, the features selected by DOM give better performance on a new domain than the features selected by the other methods. This suggests that the top-ranked features in DOM are indeed more suitable for adaptation to new domains than the top features ranked by the other methods.

The figures also show that the ranking method DOM achieved better performance than the baseline over a wide range of b values, especially in $F+Y \Rightarrow M$ and $M+Y \Rightarrow F$, whereas for methods F, IG and CHI, the performance quickly converged to the baseline performance as b increased.

It is interesting to note the comparison between F and IG (or CHI). In general, when the test data is similar to the training data, IG (or CHI) is advantageous over F (Yang and Pedersen, 1997). However, in this case when the test domain is different from the training domains, F shows advantages for adaptation. A possible explanation is that frequent features are in general less likely to be domain-specific, and therefore feature frequency can also be used as a criterion to select generalizable features and to filter out domain-specific features, although it is still not as effective as the method we proposed.

6 Related Work

The NER problem has been extensively studied in the NLP community. Most existing work has focused on supervised learning approaches, employing models such as HMMs (Zhou and Su, 2002), MEMMs (Bender et al., 2003; Finkel et al., 2005), and CRFs (McCallum and Li, 2003). Collins and Singer (1999) proposed an unsupervised method for named entity classification based on the idea of co-training. Ando and Zhang (2005) proposed a semi-supervised learning method to exploit unlabeled data for building more robust NER systems. In all these studies, the evaluation is conducted on unlabeled data similar to the labeled data.

Recently there have been some studies on adapting NER systems to new domains employing techniques such as active learning and semi-supervised learning (Shen et al., 2004; Mohit and Hwa, 2005),

or incorporating external lexical knowledge (Ciaramita and Altun, 2005). However, there has not been any study on exploiting the domain structure contained in the training examples themselves to build generalizable NER systems. We focus on the domain structure in the training data to build a classifier that relies more on features generalizable across different domains to avoid overfitting the training domains. As our method is orthogonal to most of the aforementioned work, they can be combined to further improve the performance.

7 Conclusion and Future Work

Named entity recognition is an important problem that can help many text mining and natural language processing tasks such as information extraction and question answering. Currently NER faces a poor domain adaptability problem when the test data is not from the same domain as the training data. We present several strategies to exploit the domain structure in the training data to improve the performance of the learned NER classifier on a new domain. Our results show that the domain-aware strategies we proposed improved the performance over a baseline method that represents the state-of-the-art NER techniques.

Acknowledgments

This work was in part supported by the National Science Foundation under award numbers 0425852, 0347933, and 0428472. We would like to thank Bruce Schatz, Xin He, Qiaozhu Mei, Xu Ling, and some other BeeSpace project members for useful discussions. We would like to thank Mark Sammons for his help with FEX. We would also like to thank the anonymous reviewers for their comments.

References

Rie Kubota Ando and Tong Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *Proceedings of ACL-2005*.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of CoNLL-2003*.

Andrew Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, New York University.

Stanley F. Chen and Ronald Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, School of Computer Science, Carnegie Mellon University.

Massimiliano Ciaramita and Yasemin Altun. 2005. Named-entity recognition in novel domains with external lexical knowledge. In *Workshop on Advances in Structured Learning for Text and Speech Processing (NIPS-2005)*.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of EMNLP/VLC-1999*.

Jenny Finkel, Shipra Dingare, Christopher D. Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. 2005. Exploring the boundaries: Gene and protein identification in biomedical text. *BMC Bioinformatics*, 6(Suppl 1):S5.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003*.

Lynette Hirschman, Marc Colosimo, Alexander Morgan, and Alexander Yeh. 2005. Overview of BioCreAtIvE task 1B: normailized gene lists. *BMC Bioinformatics*, 6(Suppl 1):S11.

Fei Huang and Stephan Vogel. 2002. Improved named entity translation and bilingual named entity extraction. In *ICMI-2002*.

Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. 2003. Named entity recognition with character-level models. In *Proceedings of CoNLL-2003*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL-2003*.

Behrang Mohit and Rebecca Hwa. 2005. Syntax-based semi-supervised named entity tagging. In *Proceedings of ACL-2005*.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of ACL-2004*.

Rohini Srihari and Wei Li. 1999. Information extraction supported question answering. In *TREC-8*.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of ICML-1997*.

Guodong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of ACL-2002*.