MULTI-CRITERIA-BASED ACTIVE LEARNING FOR NAMED ENTITY RECOGNITION

SHEN DAN

(B.Eng., SJTU, PRC)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Name:	Shen Dan
Degree:	M.Sc.
Dept:	Computer Science, School of Computing
Thesis Title:	Multi-Criteria-based Active Learning for Named Entity Recognition

ABSTRACT

In this thesis, we propose a multi-criteria-based active learning approach and effectively apply it to the task of named entity recognition. Active learning targets to minimize the human annotation efforts to learn a model with the same performance level as supervised learning by selecting the most useful examples for labeling. To maximize the contribution of the selected examples, we consider the multiple criteria including *informativeness*, *representativeness* and *diversity* and propose some measurements to quantify them respectively in the SVM-based named entity recognition. More comprehensively, we effectively incorporate all the criteria using two active learning strategies, both of which result in less labeling cost than the single-criterion-based method. The best results show that the labeling cost can be reduced by 95% in the newswire domain and 86% in the biomedical domain without degrading the performance of the named entity recognizer. To our best knowledge, this is not only the first work to incorporate the multiple criteria in active learning but also the first work to study active learning for named entity recognition. Furthermore, since the above measurements and active learning strategies are quite general, they can also be easily adapted to other natural language processing tasks.

Keywords: active learning, named entity recognition, multiple criteria, informativeness, representativeness, diversity

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Su Jian, who has the largest immediate influence on this thesis, for her invaluable motivation, advice, comments throughout my research and my co-supervisor, Prof. Tan Chew Lim for his endless support and encouragement. I would also like to thank Dr. Zhou Guo Dong for his suggestion and comments regarding this thesis.

I gratefully acknowledge the financial support of National University of Singapore in the form of a research scholarship. I would also like to express my gratitude to Institute for Infocomm Research which provides me an excellent environment and facilities to study and research.

Special gratitude goes to Mr. Zhang Jie. Without his encouragement and support on the experiment, my research could not have been so smooth. It has been great pleasure working with him. I would also like to thank all my friends, Mr. Yang Xiao Feng, Mr. Hong Hua Qing, Ms. Xiao Juan and Mr. Niu Zheng Yu in the natural language synergy lab for their help, which make these 18 months a wonderful experience.

Last but not least, I would like to express my sincerest thanks to my parents. Their love and understanding are my impetus to do the research during my graduate studies.

TABLE OF CONTENTS

SUMMARY

Named entity recognition (NER) is a fundamental step to many natural language processing tasks. In recent years, more and more NER systems are developed using machine learning methods. In order to achieve the best performance, the systems are generally trained on a large human annotated corpus. However, since annotating such a corpus is very expensive and time-consuming, it is difficult to adapt the existing NER systems to a new application or domain. In order to overcome the difficulty, we try to develop automated methods to reduce the training cost without degrading the performance by using active learning.

Active learning is based on the assumption that a small number of annotated examples and a large number of unannotated examples are available. It selects examples actively and trains a model progressively to avoid redundantly labeling the examples which make little contribution to the model. For efficiency, a batch of examples is often selected at a time, which is called batch-based active learning. Different from some simple tasks, such as text classification, we define an example as a word sequence (named entity) in NER. In order to minimize the human annotation efforts, we propose a new multi-criteria-based active learning method based on the comprehensive criteria including informativeness, representativeness and diversity to select the most useful examples in the training process. Firstly, the informativeness criterion concerns the examples for which the current model are most uncertain. We propose three scoring functions to quantify the informativeness of a named entity. Secondly, the representativeness criterion concerns the similarities among the examples and prefers to select the examples with the most number of similar examples. Thus, we can avoid selecting outliers. We use the cosine- similarity measurement to quantify the similarity between two words and implement a dynamic time warping algorithm to calculate the similarity between two named entities. With similarity values among the named entities, the representativeness of a named entity can be quantified by its density. Thirdly, the diversity criterion tries to maximize the training utility of a batch of examples. It can avoid selecting repetitious examples in a batch. We propose two methods, a global and a local consideration, to incorporate the diversity criterion into active learning. Last but not least, we develop two active learning strategies to combine the three criteria all together in the training process. To our best knowledge, we are not only the first work that considers the informativeness, the representativeness and the diversity criteria all together, but also the first work that studies active learning for NER.

The experiments on NER show that the labeling cost can be significantly reduced by 95% in the newswire domain and 86% in the biomedical domain comparing with supervised learning. We also find that, in addition to the informativeness criterion, the representativeness and diversity criteria are also useful for active learning. The two active learning strategies, which we propose to combine the three criteria, outperform the single-criterion-based active learning methods.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

Named Entity Recognition (NER) is a fundamental step to many natural language processing tasks, such as Information Extraction, Information Retrieval and Question Answering. Traditional NER is defined by the Message Understanding Conferences (MUC), which recognizes names of entities such as PERSON, LOCATION and ORGANIZATION in the newswire domain. In recent years, the NER technique has been widely used in the biomedical domain, which recognizes names of entities, such as protein, DNA, RNA and cell line. In order to achieve the best performance, named entity recognizers are generally trained on a large annotated corpus, such as MUC-6 corpus and GENIA corpus. However, since annotating such a corpus is very expensive and time-consuming, it difficult to adapt the existing named entity recognizers to a new application or domain. In order to overcome the difficulty, we are to develop automated methods to reduce the training cost without degrading the performance within the framework of active learning. Active learning selects the most useful examples for labeling, so it can avoid redundantly labeling the examples which make little contribution to the model. Being the first piece of work on active learning for NER, we target to minimize the human annotation effort to learn a named entity recognizer with the same performance level as supervised learning. Furthermore, since the measurements and the strategies we propose in the active learning for NER are general, they can be easily adapted to other natural language processing tasks, such as text chunking, POS tagging and statistical parsing.

1.2 Background

Active learning is based on the assumption that a small number of annotated examples and a large number of unannotated examples are available. This assumption is valid in most natural language processing tasks. Different from supervised learning in which an entire corpus are labeled manually, active learning is to select the most useful example for labeling and add the labeled example to a training set to retrain a model. This procedure is repeated until the model achieves a certain performance level. In an ideal situation, one best example is selected at a time. However, since it is time consuming to retrain the model if only one new example is added to the training set, a batch *B* of the examples (batch size $k > 1$) are often selected at a time, which is called batched-based active learning [Lewis and Gale 1994]. Figure 1.1 presents the pseudo-code for a general batch-based active learning algorithm.

```
Given:
    $U$: an unlabeled data set
    $L$: an labeled training data set
    $B$: a batch of the examples selected (the maximum size of B is $k$)
    $M$: current model
Loop until certain level of performance is achieved:
    $M \leftarrow$ Train($L$)
    $B \leftarrow$ Select($U$, $M$, $k$)
    $L \leftarrow L \cup$ Label($B$)
```

Figure 1.1: A general batch-based active learning algorithm

Active learning has been applied in more and more natural language processing tasks such as POS tagging [Dagan and Engelson 1995; Engelson and Dagan 1999], information extraction [Thompson et al. 1999; Finn and Kushmerick 2003], text classification [Lewis and Gale 1994; Lewis and Catlett 1994; McCallum and Nigam 1998; Schohn and Cohn 2000; Tong and Koller 2000; Brinker 2003], statistical parsing [Thompson et al. 1999; Hwa 2000; Tang et al. 2002; Steedman et al. 2003], noun phrase chunking [Ngai and Yarowsky 2000] and word segmentation [Sassano 2002]. However, currently, there are no works exploring active learning for NER.

In these various tasks above, active learning are mainly based on two kinds of models: statistical model, such as Hidden Markov Model and Naïve Bayes [Dagan and Engelson 1995; Engelson and Dagan 1999; McCallum and Nigam 1998; Hwa 2000; Tang et al. 2002] and discriminative model, such as Support Vector Machines [Schohn and Cohn 2000; Tong and Koller 2000; Sassano 2002; Brinker 2003]. Following the general active learning framework (Figure 1.1), various model/task-specific measurements are proposed to evaluate the usefulness of the examples in the unlabeled data set *U*. In the next section, we will briefly introduce the related active learning methods in these natural language processing tasks.

1.3 Related Work

Although many supervised machine learning methods have achieved promising performances in the natural language processing tasks, they strongly depend on the availability of a large amount of annotated corpus. Nowadays, more and more researchers are interested in studying how to reduce the human annotation cost without degrading the performance by incorporating an active learning process into the existing model. From the selection strategy point of view, all of the previous active learning methods can be grouped into two types: committee-based and certainty-based.

1.3.1 Committee-based Active Learning

Committee-based active learning has been widely applied in statistical models for various natural language processing tasks. The representative research efforts include [Dagan and Engelson 1995; Engelson and Dagan 1999], [McCallum and Nigam 1998] and [Ngai and Yarowsky 2000].

[Dagan and Engelson 1995; Engelson and Dagan 1999] propose a committee-based active learning method to efficiently learn a Hidden Markov Model (HMM) for Part of Speech (POS) tagging by selecting only the most informative examples for labeling in a stream of unlabeled data set. The informativeness of an example is evaluated based on the disagreement level between several model variants (committee members). The disagreement level is quantified by using the entropy of the distribution of the tags assigned by the committee members, called vote-entropy. Given the statistics acquired from the training set selected so far, the committee members are generated according to the posterior probability distribution of the possible classifiers (Monte-Carlo sampling). Finally, the examples with the highest disagreement level among the committee members are selected for labeling. In the POS tagging, each sentence is considered as an example. The learning efficiency of the committee-based active learning method is compared to that of random selection in their experiments. The results show that the committee-based method requires less than one-fourth the amount of training data that the random selection does to reach 90.5% accuracy. In addition, Engelson and Dagan also investigate several different selection methods in depth.

[McCallum and Nigam 1998] combine active learning and Expectation Maximization (EM) on a pool of unlabeled data for text classification. In the part of active learning, a committee-based active learning method is proposed to select most informative documents for labeling. Compared with [Dagan and Engelson 1995], they present a better measurement to evaluate the committee members' disagreement, called Kullback-Leibler (KL) divergence to the mean. Unlike the vote entropy measurement, which compares only the committee members' top ranked class, KL divergence further consider the differences in the committee members' class distributions. More importantly, they further study the representativeness of a document in addition to its informativeness. They model the document density explicitly by measuring two documents' distance based on the word co-occurrence probabilities. A document with large density is considered strongly prototypical for a certain class. Finally, the overall contribution of an unlabeled document is measured by the committee members' disagreement (KL divergence) and its density, called Density-weighted KL Metric. This metric tend to select a both informative and representative document. The experimental results show that the method of combining EM and active learning requires only half as many training data to achieve the same accuracy as either EM or active learning.

[Ngai and Yarowsky 2000] apply a committee-based active learning method to base noun phrase chunking. They construct the committee members by dividing a training corpus into different subsets using bagging or n-fold partitioning. Furthermore, they propose a novel disagreement measurement between the committee members using a f-measure metric, which is called f-complement. They also state that the f-complement is more applicable and slightly outperforms the vote entropy measurement used in [Dagan and Engelson 1995; Engelson and Dagan 1999]. More importantly, the f-complement can be used in the applications where the implementation of the vote entropy is difficult, such as parsing. The comparison between the f-complement-based method and random selection shows that the method reduces the amount of data needed to reach a given performance level by approximately 50%.

1.3.2 Certainty-based Active Learning

Compared with the committee-based active learning above, there are also some groups studying the certainty-based active learning, such as [Thompson et al. 1999], [Hwa 2000], [Schohn and Cohn 2000], [Tong and Koller 2000], [Sassano 2002], [Tang et al. 2002], [Brinker 2003] and [Finn and Kushmerick 2003].

[Thompson et al. 1999] first apply active learning to two non-classification natural language processing tasks: semantic parsing and information extraction. They develop two rule-learning systems CHILL and RAPIER for the semantic parsing task and the information extraction task respectively. Then, they apply a certainty-based active learning method to both of these systems. The certainty of an example in rule-based decision is evaluated by the number of the positive and negative training examples which are used to induce the specific rules to make the decision for the example. An example with most uncertainty level is considered most informative for the learner and is selected for labeling. The results show that the active learning method can significantly reduce the number of the annotated examples required to achieve a given performance level in these two tasks.

[Hwa 2000] apply a certainty-based active learning method to statistical grammar induction. They also target to select the most informative examples for which the model are most uncertain. The grammar's certainty for assigning a parse tree to a sentence is quantified by two functions they proposed. The first function is a simple heuristic that approximates the certainty in terms of the length of the sentence. The intuition behind this function is based on the observation that longer sentences tend to have more complex structures and ambiguous parses. The second function computes the certainty in terms of the tree entropy of the sentence. The tree entropy of a sentence is computed by the distribution of the probabilities of all parses for the sentence which is produced by the current model. The best experimental result shows that the active learning method can reduce the human efforts for parsing the sentences by 36%.

[Schohn and Cohn 2000] describe an active learning method to enhance the generalization behavior of SVM for text classification. In their work, the active learning in SVM is explored based on two observations. The first is that the examples that are orthogonal to the space spanned by the current training set will be informative for the model, since they can give the information about the dimensions which the model has not yet explored. The second is that labeling the examples which lies on or close to the separating hyperplane will have a large effect on the model. Furthermore, a stopping criterion for the active learning in SVM is proposed. If the distance of the best example selected to the separating hyperplane is no closer than that of any support vectors to the hyperplane, the active learning process will be stopped and the peak of performance will be achieved. The experiment shows that SVM trained on a well-chosen data subset frequently outperforms that trained on all available data. Compared to supervised learning, the active learning method can offer better performance with fewer data.

[Tong and Koller 2000] introduce a new active learning method in the inductive and transductive setting of SVM for text classification. They provide a theoretical motivation for the active learning in SVM using the notion of version space. Based on the motivation that the examples which split the current version space into two equal parts as much as possible are most informative for the model, they present three selection methods: Simple Margin, MaxMin Margin and Ratio Margin. The experiments on Reuters-21578 data set show that the three selection methods perform similarly and each of them appreciably outperforms random selection. In this task, random selection on average requires over six times as much data as the active learning method do to achieve the same performance level.

[Sassano 2002] is the first paper on applying active learning in SVM to a more complex task, Japanese word segmentation. In particular, they discuss how the size of a pool affects the learning curve. To our understanding, the pool is the unlabeled data set from which the most useful examples are selected. It is found that the performance on a larger pool is worse than that on a smaller pool in the early stage of training. The reason may be that in the case of a larger pool, the examples iteratively selected are more likely to be similar to each other. Therefore, they propose a two-pool algorithm which gradually moves examples from a large unlabeled data set (a secondary pool) to a small unlabeled data set (a primary pool) and then selects examples directly from the primary pool. The algorithm implicitly decreases the probability of selecting similar examples into a batch. The experiments show that the two pool algorithm only needs 59.3% of the labeled data which are required in the general active learning algorithm and only 17.4% of the labeled data which are required in random selection.

[Tang et al. 2002] propose an active learning method based on more comprehensive considerations including informativeness and representativeness for statistical parsing. In the consideration of the informativeness, they use an uncertainty-based selection method. They take advantage of the availability of parsing scores from the existing statistical parser and propose three entropy-based uncertainty scores. The first score is computing the entropy of the most probable parse tree of a sentence, which can be represented by a sequence of events. The second score is computing the entropy of the distribution over all candidate parses of a sentence. The third score is computing the per word entropy of a sentence by normalizing the sentence entropy (the second score above) by the length of the sentence. In the consideration of the representativeness, a model-specific distance is proposed to measure the difference between the most likely parse trees of two sentences. Based on the distances, the density of a sentence is computed to quantify its representativeness. Finally, the examples are selected and weighted based on its uncertainty and density value respectively. The best result shows that for the same accuracy, only a third of the examples are needed to annotate as compared to random selection.

[Brinker 2003] especially design an active learning method for batch-based sample selection and apply it to text classification. Compared with [Sassano 2002], the active learning method explicitly avoids selecting similar examples into a batch by incorporating a diversity measurement. The diversity degree between two examples is measured by the angles of the feature vectors of the examples in the sample space. Furthermore, they propose a batch-based active learning strategy which combines the certainty measurement and the diversity measurement by using linear interpolation. To our knowledge, this is the only work exploring the diversity criterion in active learning. The experiment indicates that the combination strategy outperforms both the general active learning methods and random selection in SVM for text classification.

[Finn and Kushmerick 2003] investigate several active learning approaches that are particularly relevant to information extraction. Through the active learning approaches, users are required to label the most informative documents only. They propose two main approaches to estimate the informativeness of a document: confidence-based and distance-based. In the confidence-based approach, the confidence of the existing model for a document is the same as the certainty of the model for the document, so this approach can be regarded as a certainty-based active learning approach, which has been explored in many previous works. In the distance-based approach, they assume that the training data set which can optimize the performance of the learner should have the maximum pair-wise distance between its members. Based on the assumption, they select the documents that are most different to those already in the training data set. The difference between two documents is evaluated by using a distance metric which is specific to the information extraction task. Furthermore, they also use a simple method, called ENSEMBLE, to combine the two approaches. In the ENSEMBLE, half of the documents are selected using the confidence-based approach and half of the documents are selected using the distance-based approach. The experiments show that the confidence-based approach is biased toward improving precision, while the distance-based approach is biased toward improving recall. But neither of them can achieve both high recall and precision. In addition, the experiments also show that the ENSEMBLE performs slightly better than either of the approaches.

From the review of the recent literatures on active learning, we find that most of the existing works in the area are only based on the informativeness consideration although various active learning methods, such as certainty-based methods and committee-based methods are proposed for various tasks. [McCallum and Nigam 1998] and [Tang et al. 2002] are the only two works considering the representativeness in active learning. However, the measurements they propose to quantify the representativeness are very specific to their tasks (text classification and semantic parsing) and are difficult to be adapted to other tasks. On the other hand, [Brinker 2003] first consider the diversity in batch-based active learning in addition to the informativeness. However, he didn't further explore how to avoid selecting outliers to a batch. So far, we haven't found any previous works integrating the informativeness, representativeness and diversity all together.

1.4 Contribution

Our contribution to the research of active learning for named entity recognition can be concluded as follows:

Firstly, we present a novel active learning method, called multi-criteria-based active learning, based on more comprehensive criteria including informativeness, representativeness and diversity. We develop various measurements to quantify the criteria respectively and propose two active learning strategies to effectively combine them. These combination strategies are to maximize not only the contribution of individual examples but also the contribution of a batch. Although the individual criterion has been explored in few research works respectively (refer to Section 1.3), this is the first work to incorporate them all together to select the most useful examples. The experiment also indicates that active learning based on the multi-criteria outperforms that based on the single criterion, such as the traditional certainty-based active learning.

Secondly, this is the first time to study how to effectively incorporate active learning to named entity recognition. Firstly, we propose three scoring functions to evaluate the informativeness of a named entity. Secondly, we employ an algorithm to compute the similarity between named entities and propose a measurement to compute the representativeness of a named entity based on the similarities. Thirdly, we make a global consideration by using K-Means algorithm and a local consideration by making pair-wise comparisons for the diversity of a batch. The experiment shows that the active learning method achieves a promising result in NER. It is found that the amount of the labeled training data can be reduced by 95% in the newswire domain and 86% in the biomedical domain without degrading the performance of the named entity recognizer.

Thirdly, in the active learning framework, the measurements that we propose are more general than those in [McCallum and Nigam 1998; Tang et al. 2002] and may be easily adapted to other natural language processing tasks when the example to be selected is a sequence of words. Therefore, the multi-criteria-based active learning method can also contribute to other tasks, such as text chunking, POS tagging and parsing.

1.5 Organization of the Thesis

The thesis is organized as follows: Chapter 2 provides a brief introduction of the SVM-based NER system in both the newswire domain and the biomedical domain. Moreover, the general framework of active learning for NER is described in the last section of this chapter. In Chapter 3, we present the multiple criteria, viz. informativeness, representativeness and diversity, used in the active learning method for NER and propose some measurements to quantify them. In Chapter 4, we propose two active learning strategies to effectively combine the criteria and incorporate the strategies into the SVM-based named entity recognizer. In Chapter 5, we show our experimental configurations and various experimental results. Finally, in Chapter 6, we conclude this thesis with the future works.

Chapter 2

SVM AND NAMED ENTITY RECOGNITION

2.1 SVM

Support Vector Machines (SVM) [Vapnik 1995] is a powerful machine learning method, which has been applied successfully in named entity recognition, such as [Zhou et al. 2004b; Lee et al. 2003; Kazama et al. 2002; Takeuchi and Collier 2002].

SVM constructs a binary classifier that predict whether an instance, which is presented as a feature vector in a space $\boldsymbol{R}^n$ ($\boldsymbol{x} \in \boldsymbol{R}^n$), is positive ($f(\boldsymbol{x}) = 1$) or negative ($f(\boldsymbol{x}) = -1$). In the simplest form (linear SVM trained on separable data), the decision is based on a separating hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ as follows:

$f(\boldsymbol{x}) = sgn(\boldsymbol{w} \cdot \boldsymbol{x} + b)$ for $\boldsymbol{w} \in \boldsymbol{R}^n$ and $b \in \boldsymbol{R}$

All instances lying on one side of the hyperplane are classified to a positive class, while others are classified to a negative class.

Given a set of labeled training instances $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$, where $\boldsymbol{x}_i \in \boldsymbol{R}^n$ and $y_i = \{1, -1\}$, SVM is to find the optimal hyperplane that separates the positive and negative training instances with a maximum margin, as shown in Figure 2.1. The margin is defined as the shortest distance from the separating hyperplane to the closest positive (negative) training instances.

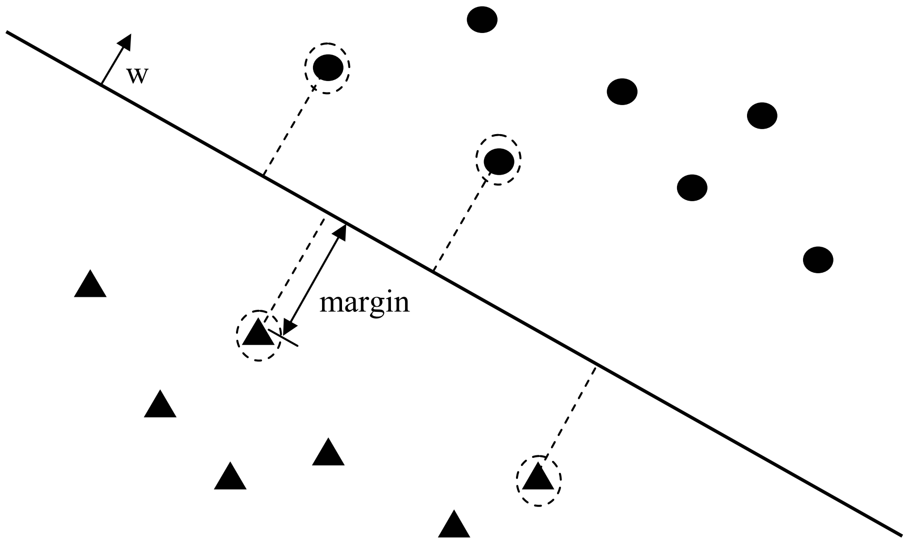


Figure 2.1: Linear separating hyperplane for the separable case in SVM

The positive (negative) training instances nearest to the separating hyperplane are called support vectors, for which $|(\boldsymbol{w} \cdot \boldsymbol{x} + b)| = 1$. In Figure 2.1, the support vectors are in dashed line. Support vectors are the critical elements of a training data set since they lie closest to the decision boundary (separating hyperplane). Even if all the other training instances are removed, the separating hyperplane will not be changed. Practically, training SVM is to find the support vectors and their weights from the training data set by solving a quadratic programming problem. Based on the weighted support vectors, the decision can be reformulated as follows:

$$f(\boldsymbol{x}) = sgn(\boldsymbol{w} \cdot \boldsymbol{x} + b) = sgn(\sum_{\boldsymbol{s}_i \in SVs} y_i \alpha_i \boldsymbol{x} \cdot \boldsymbol{s}_i + b)$$

where, $\boldsymbol{s}_i$ is one of the support vectors and α_i is the weight of $\boldsymbol{s}_i$.

In a more general form (nonlinear SVM), we use a function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, called kernel function, instead of the inner product in the above formula. The kernel function projects an instance in the original space $\boldsymbol{R}^{\boldsymbol{n}}$ to a higher dimensional space. Then, a separating hyperplane are constructed in the higher dimensional space. Corresponding to the original space $\boldsymbol{R}^{\boldsymbol{n}}$, a non-linear separating surface is found. By this means, we are still doing a linear separation, but in a different space. The kernel function has to be defined based on the Mercer's condition. Generally, the following kernel functions are widely used in natural language processing tasks.

Polynomial kernel function: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^p$

Gaussian radial basis kernel function: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma^2}$

Sigmoidal kernel function: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\kappa \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \delta)$

Since SVM is well described in the cited literatures, from now on, we will focus on the development of a named entity recognizer using SVM.

2.2 Named Entity Recognition

Named entity recognition is to recognize pre-defined names in texts, such as person, location, organization names in the newswire articles and protein, DNA, RNA names in the biomedical articles. Conceptually, it can be regarded as a combination of two procedures: identification, which finds the boundaries of a named entity in a text, and classification, which determines the semantic class of the identified named entity. We develop our named entity recognizer using the $\text{SVM}^{\text{Light}}$ software[1] [Joachims 1999] which is an combination of Vapnik's Support Vector Machine and an optimization algorithm [Joachims 2002].

2.2.1 Definition of Named Entity Recognition

Different from the traditional NER task, we develop a simple and effective named entity recognizer [Zhou et al. 2004b] which recognizes one class of named entities at a time, such as recognize protein names in the biomedical articles. Since there is only one class of named entities to recognize, we employ *IO* tags to represent the region information of the named entities in stead of the traditional *BIO* tags [Shen et al. 2003]. In the *IO* representation, *I* indicate the current word is a part of a named entity, which corresponds to the SVM output 1; *O* indicates the current word is not a part of a named entity, which corresponds to the SVM output -1. Here is an example of the *IO* representation for the protein named entity recognition.

Interleukin-5	signaling	in	human	eosinophils	involves	JAK2	tyrosine	kinase	and …
I	*O*	*O*	*O*	*O*	*O*	*I*	*I*	*I*	*O*

After the simplification, the task becomes a binary classification task, which classify each word to either the class *I* or the class *O*. The limitation of the *IO* representation is that it cannot provide enough information to differentiate consecutive named entities. However, it simplifies the NER task a lot since we can avoid the multi-class problem in SVM. We find it is a worth tradeoff.

[1] http://svmlight.joachims.org/.

Certainly, we can further study how to effectively combine several named entity recognizers which recognize different classes of named entities respectively, and build a combination system to recognize more than one class of name entities at a time in future work.

2.2.2 Features

We use a binary feature vector representation for a word with its contexts in SVM. Each dimension of the vectors indicates whether the word has a certain feature. In our task, we develop a named entity recognizer for two domains: recognizing the named entities of person, location, organization in the newswire domain and the named entities of protein in the biomedical domain. Since named entities in the two domains have different characteristics, which has been described in [Shen et al. 2003; Zhou et al. 2004a; Zhang et al. 2004] in detail, we design different features to cope with them.

Note that since the named entity recognizer will be used for active learning and there is only a few labeled training data initially, the features which are produced statistically from the training data set will not be incorporated into the model, which is different from the supervised named entity recognizer we develop previously [Zhou and Su 2002; Shen et al. 2003; Zhou et al. 2004a; Zhang et al. 2004; Zhou et al. 2004b]. Furthermore, no gazetteer or dictionaries are used in our model. Therefore, in active learning for the NER task, human experts are required to provide only some basic knowledge for the certain class of named entities, such as some semantic triggers, and to label the most useful examples iteratively.

- **Features in the Newswire Domain**

In the newswire domain, we use the same features including surface word, orthographic features and semantic trigger features as [Zhou and Su 2002].

1) **Surface Word**: if a word occurs in a vocabulary, one dimension in the feature vector of the word corresponding to its position in the vocabulary is set to 1. The vocabulary is constructed by taking all the words from all available documents.

2) **Orthographic Features**: the orthographic features are manually designed to capture the word formation information, such as capitalization and digitalization. In the newswire domain, they are helpful not only to identify the region information but also to distinguish the classes for named entities. For examples, *CapPeriod* often indicates a person name initial. Table 2.1 shows the sorted list of orthographic features we designed for this domain. Each orthographic feature corresponds to one dimension in the feature vector.

3) **Semantic Trigger Features**: the semantic trigger features consist of some special words for a class of named entities, as shown in Table 2.2. They are very useful for classifying named entities according to the semantic information. In our task, we use about 179 trigger words for person names, 36 trigger words for location names and 177 trigger words for organization names, which are provided by human experts. Each trigger word corresponds to one dimension in the feature vector.

Orthographic Feature	Example	Explanation
OneDigitNum	9	Digital Number
TwoDigitNum	90	Two-Digit year
FourDigitNum	1990	Four-Digit year
YearDecade	1990s	Year Decade
ContainsDigitAndAlpha	A8956-67	Product Code
ContainsDigitAndDash	09-99	Date
ContainsDigitAndOneSlash	3/4	Fraction or Date
ContainsDigitAndTwoSlashs	19/9/1999	DATE
ContainsDigitAndComma	19,000	Money
ContainsDigitAndPeriod	1.00	Money, Percentage
OtherContainsDigit	123124	Other Number
AllCaps	IBM	Organization
CapPeriod	M.	Person Name Initial
CapOtherPeriod	St.	Abbreviation
CapPeriods	N.Y.	Abbreviation
FirstWord	First word of sentence	No useful capitalization info.
InitialCap	Microsoft	Capitalized Word
LowerCase	will	Un-capitalized Word
Other	$	All other words

Table 2.1: The sorted list of orthographic features in the newswire domain

NE Class	Semantic Triggers	Example	Explanation
PERSON (179)	PrefixPERSON1 PrefixPERSON2 FirstNamePERSON …	Mr. President Michael …	Person Title Person Designation Person First Name …
LOC (36)	SuffixLOC …	River …	Location Suffix …
ORG (177)	SuffixORG …	Ltd …	Organization Suffix …

Table 2.2: Examples of semantic trigger features in the newswire domain

Note that Part of Speech (POS) features are not used here based on the observation of their effectiveness in the previous supervised NER model [Zhou and Su 2002]. The experiment in the supervised learning model shows that the incorporation of the POS features in the newswire domain even degrades the performance.

- **Features in the Biomedical Domain**

Previous research works [Shen et al. 2003; Zhou et al. 2004a; Zhang et al. 2004] state that named entity recognition in the biomedical domain is more difficult than that in the newswire domain. Based on the characteristics of biomedical named entities, we have to explore more effective features. In this domain, we use the same features as those in our system [Zhou et al. 2004b], which achieves the best performance in the closed test of BioCreAtIve Competition 2003[2]. The features are grouped as follows:

1) **Surface Word**: if a word occurs in a vocabulary, one dimension in the feature vector of the word corresponding to its position in the vocabulary is set to 1. The vocabulary is constructed by taking all the words from all available documents.

2) **Orthographic Features**: In the supervised NER model, we find orthographic features have weaker predictive power for named entity classification in the biomedical domain than in the newswire domain. However, the features still can indicate the occurrence of unknown words, such as abbreviations. Table 2.3 shows the list of orthographic features we used in the biomedical domain. Comparing Table 2.3 with Table 2.1, one can find that the features such as *GreekLetter*, *RomanDigit*, *ATCGseq* and the features dealing with mixed alphabetical letters and digits are specially designed for the biomedical domain.

   In SVM, since each orthographic feature corresponds to one dimension in the feature vectors, one word may have more than one orthographic feature by setting the values

[2] http://www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html

of the corresponding dimensions to 1. This is different from our previous HMM-based model in which one word has only one orthographic feature according to the priority. In our task, if a word contains hyphens, we will separate the word into several parts according to the positions of the hyphens and consider the orthographic features in all parts respectively. For example, the word *TGF-alpha* has the orthographic features *AllCap* and *GreekLetter*. By this way, we may capture more format information of the word.

Orthographic Feature	**Example**	**Explanation**
Comma	,	comma
Dot	.	dot
LRB	(	left round bracket
RRB	)	right round bracket
LSB	[	left squared bracket
RSB	]	right squared bracket
RomanDigit	II, IV	Roman digit
GreekLetter	beta	Greek letter
StopWord	in, at	stop word
ATCGseq	AACAAAG	nucleotide sequence
OneDigit	5	one digit
AllDigits	60	all digits
DigitCommaDigit	1,25	digits + comma + digits
DigitDotDigit	0.5	digits + dot + digits
OneCap	T	single capital letter
AllCaps	CSF	all capital letters
CapLowAlpha	All	capital letter followed by lowercase letters
CapMixAlpha	IgM	capital letter followed by mixture of cases
LowMixAlpha	kDa	lowercase letter followed by mixture of cases
AlphaDigitAlpha	H2A	letters + digits + letters
AlphaDigit	T4	letters + digits
DigitAlphaDigit	6C2	digits + letters + digits
DigitAlpha	19D	digits + letters

Table 2.3: The list of orthographic features in the biomedical domain

3) **POS Features:** Since many biomedical named entities are descriptive and long, identifying the boundaries of named entities in the biomedical domain is not a trivial task. POS tags provide the evidence of noun phrase region based on the syntactic information of words, therefore, they can help to solve the problem of the boundary identification. In previous work [Shen et al. 2003; Zhou et al. 2004a; Zhang et al. 2004], we have found that the POS tagger trained on the biomedical documents perform much better on the biomedical test set than that trained on the WSJ documents. So, in our task, we train a HMM-based POS tagger on the GENIA corpus V3.02p [Ohta et al. 2002] in stead of the PENN TreeBank corpus to effectively adapt the POS tagger to the biomedical domain. Then, we use the POS tagger to assign the POS feature to each word. One POS tag corresponds to one dimension in the feature vector. In the supervised learning model, POS features are proved very beneficial and make a significant improvement of performance.

4) **Morphological Features:** The morphological information, such as prefix and suffix, is considered as an important cue for terminology identification. According to the basic knowledge of protein names, some suffixes, such as ~ase, ~zyme, ~ome, ~gen, are used in the model. Certainly, some common words of these suffixes will be filtered out, such as, *disease*, *base*, case and *come.* Each suffix corresponds to one dimension in the feature vector.

5) **Semantic Trigger Features**: the semantic trigger features, which are supplied by users, consist of some special head nouns and some context words for a class of named entities. Head noun means a noun or noun phrase of some compound words,

which describes the function or property of these words, e.g. *motif* is the head noun for the protein name <PROTEIN>*AP-4 HLH motif*</PROTEIN>. Compared with the other words in a biomedical named entity, the head noun is a decisive factor for classifying the named entity. Table 2.4 shows some examples of the unigram and bi-gram head nouns for the named entities of protein. Furthermore, we also use some context words of named entities as the semantic triggers. These context words can help to identify and classify named entities, but for themselves, are always excluded from the named entities. For example, the word *activation* is mostly following a protein name such as <PROTEIN>*ERK*</PROTEIN> *activation* and *activation of* <PROTEIN>*MKP-3*</PROTEIN>, that is, one noun phrase adjacent to the word *activation* is more likely to be a protein name. Some examples of the context words are also shown in Table 2.4. Each semantic trigger corresponds to one dimension in the feature vector and totally 99 semantic triggers including 65 head nouns and 34 context words are used to recognize the named entities of protein.

NE Class	**Semantic Triggers**	**Unigram**	**Bi-grams**
Protein (99)	head noun (65)	protein promoter motif enhancer …	transcription factor binding site NF kappaB binding factor …
	context words (34)	activation transcription stimulation mutation …	

Table 2.4: Examples of semantic trigger features in the biomedical domain

Up to now, we have introduced all of the features of a word used in the SVM-based named entity recognizer in both the newswire domain and the biomedical domain. Note that a window of a target word is also used to make a decision on the word. In this task, we set the window size as 7, that is, the features of the previous and next 3 words will also be included into the feature vector of the target word.

2.3 Active Learning for Named Entity Recognition

In this section, we will discuss how to incorporate an active learning process into the named entity recognizer. Being the first piece of work on active learning for NER, we target to minimize the human annotation efforts to learn a model which can still reach the same performance level as supervised learning. We select the examples with the maximum contribution to the model for labeling iteratively instead of blindly labeling a whole corpus. Before we propose the active learning strategies, let's discuss how to define an example unit to be selected in the NER task. There are three ways to define an example unit:

The simplest one is a word-based example definition, like word segmentation [Sassano 2002], which iteratively selects the most useful word as an example unit and require human experts to classify it into the class *I* or the class *O*, just like the binary classification in SVM. However, in the NER task, it is not reasonable to select a single word without contexts for labeling manually. Even if we require human experts to label a single word, they have to make an additional effort to refer to the contexts of the word. Therefore, although this method can minimize the training data size, it can not minimize the actual human annotation efforts.

The next one is a sentence-based example definition, which iteratively select the most useful sentence as an example unit and require human experts to read and annotate the sentence. Many previous active leaning methods are based on the definition, such as POS tagging [Dagan and Engelson 1995; Engelson and Dagan 1999] and statistical parsing [Thompson et al. 1999; Hwa 2000; Tang et al. 2002]. Certainly, it is more reasonable than the word-based example definition in our NER task. However, we find that only few words in a sentence are considered to be a named entity while most of the words in the sentence are not useful. Therefore, human experts may not need to read the whole sentence to annotate one named entity, such as a person name and a location name, in the sentence.

Therefore, in active learning for the NER task, we use a named entity-based example definition. We select a word sequence, which consists of a named entity and its context, as an example unit rather than a single word or a full sentence, and require human experts to label/correct it.

Based on this example definition, all of the measurements we propose in active learning should be applied to named entities. Since only word-based scores are available from the existing SVM model, we have to further study how to extend the measurements for words to those for named entities. Thus, active learning for the SVM-based NER will be more complex than that for the simple classification tasks, such as text classification on which most research works on active learning in SVM [Schohn and Cohn 2000; Tong and Koller 2000; Brinker 2003] are conducted.

In the next chapter, we will consider the multiple criteria, viz. informativeness, representativeness and diversity, used in active learning for NER and propose some measurements to quantify the criteria.

Chapter 3

MULTIPLE CRITERIA FOR ACTIVE LEARNING

In this Chapter, we explore the multiple criteria, viz. informativeness, representativeness and diversity, used in active learning for NER and propose some measurements to quantify the criteria.

3.1 Informativeness

The basic idea of the informativeness criterion is that we would like to select the examples on which the existing model cannot do well. Many previous active learning methods, such as the committee-based methods [Dagan and Engelson 1995; Engelson and Dagan 1999; McCallum and Nigam 1998; Ngai and Yarowsky 2000] and the certainty-based methods [Thompson et al. 1999; Hwa 2000; Tang et al. 2002; Schohn and Cohn 2000; Tong and Koller 2000; Brinker 2003], are based on this criterion. In our task, we use a certainty-based method to select the most informative examples for which the existing model are most uncertain. We propose a distance-based measurement to quantify the certainty of a word in SVM and extend it to a measurement to quantify the certainty of a named entity using three scoring functions.

3.1.1 Informativeness Measurement for Word

In a linear and separable form, SVM is to find a hyperplane that separates the positive and negative examples in a training data set with the maximum margin. The margin is defined by the distance of the separating hyperplane to the nearest positive and negative examples. The training examples which are closest to the hyperplane are called support vectors. In

SVM, only support vectors are useful for the classification, which is different from statistical models. SVM training is to get the support vectors and their weights from the training data set by solving a quadratic programming problem. The weighted support vectors are later used to classify the test data, as described in Chapter 2.

We consider the informativeness of an example as how it will make effect on the support vectors by adding it to the training data set. Intuitively, if the distance of an example's feature vector to the separating hyperplane is less than the margin (the distance of the support vectors to the hyperplane), adding the example to the training data set may induce the new support vectors and change the separating hyperplane. So, the example is regarded informative for the learner. This intuition is also justified by [Schohn and Cohn 2000; Tong and Koller 2000] based on a theoretical version space analysis. They state that labeling an example that lies on or close to the hyperplane is guaranteed to have an effect on the solution. In our task, we use the distance of an example's feature vector to the hyperplane to measure the informativeness degree of the example.

The distance of an example's feature vector $\boldsymbol{x}$ to the separating hyperplane is computed as follows:

$$Dist(\boldsymbol{x}) = \left| \sum_{i=1}^{M} \alpha_i y_i k(\boldsymbol{s}_i, \boldsymbol{x}) + b \right|$$

where $\boldsymbol{x}$ is the feature vector of the example, α_i, y_i, $\boldsymbol{s_i}$ corresponds to the weight, the class and the feature vector of the i^{th} support vector respectively. M is the number of the support vectors in the current model. We would like to select the example with the minimal *Dist*, which indicates that it comes closest to the hyperplane in the feature space and is most informative for the current model.

3.1.2 Informativeness Measurement for Named Entity

Based on the above informativeness measurement for a word, we compute the overall informativeness degree of a named entity *NE*, which is considered as a sequence of words. Let $NE = \mathbf{w}_1\mathbf{w}_2\ldots\mathbf{w}_N$, in which $\mathbf{w}_i$ is the i^{th} word of *NE*. $\mathbf{w}_i$ is represented as a feature vector in SVM; *N* is the number of words in *NE*. Note that the words outside the margin are not useful for the model, so we ignore such words by setting their distance to 1.

$$Dist^*(\mathbf{w}_i) = \begin{cases} Dist(\mathbf{w}_i) & \text{if } Dist(\mathbf{w}_i) \le 1 \\ 1 & \text{otherwise} \end{cases}$$

Then, we propose three scoring functions to quantify the informativeness of a named entity as follows.

- **Info_Avg**: The informativeness of *NE* is scored by the average distance of the words within *NE* to the hyperplane.

$$Info(NE) = 1 - \frac{\sum_{\mathbf{w}_i \in NE} Dist^*(\mathbf{w}_i)}{N}$$

where, $\mathbf{w}_i$ is the feature vector of the i^{th} word in *NE*.

- **Info_Min**: The informativeness of *NE* is scored by the minimal distance of the words within *NE* to the hyperplane.

$$Info(NE) = 1 - \underset{\boldsymbol{w}_i \in NE}{Min}\{Dist^*(\boldsymbol{w}_i)\}$$

- **Info_InclRate**: If the distance of a word to the hyperplane is less than a threshold α (= 1 in our task), the word is assumed to have short distance to the hyperplane. Then, we compute the proportion of the number of words with short distance to the total number of words within *NE* and use the inclusion rate to quantify the informativeness of *NE*.

$$Info(NE) = \frac{\underset{\boldsymbol{w}_i \in NE}{NUM\,(Dist^*}(\boldsymbol{w}_i) < \alpha)}{N}$$

In Chapter 5, we will evaluate the effectiveness of these three scoring functions.

3.2 Representativeness

In addition to the most informative example, we also prefer the most representative example. The representativeness of an example can be evaluated based on how many examples are similar or near to it. Thus, the examples with high representative degree are less likely to be outliers. Adding them to the training data set will have effect on a large number of unlabeled examples. To our knowledge, there are only few works considering this selection criterion [McCallum and Nigam 1998; Tang et al. 2002] and both of them are specific to their tasks, viz. text classification and statistical parsing. In this section, we propose a more general representativeness measurement. Firstly, we compute the similarity between words using a general vector-based measurement, viz. cosine similarity measurement. Secondly, we extend the measurement to compute the similarity between named entities using a dynamic time warping algorithm. Thirdly, we quantify the representativeness of a named entity by its density.

3.2.1 Similarity Measurement between Words

In a general vector space model, the similarity between two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be measured by computing the cosine value of the angle θ between them. This measurement, called cosine-similarity measurement, has been widely used in Information Retrieval [Baeza-Yates and Ribeiro-Neto 1999] to compute the similarity between two documents, or between a document and a query. They indicate that the smaller the angle, the more similar between the vectors. The measurement is as follows:

$$\cos\theta = \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}{\left\|\boldsymbol{x}_i\right\| \cdot \left\|\boldsymbol{x}_j\right\|}$$

In our task, we also use the cosine similarity measurement to quantify the similarity between two words represented as feature vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ in SVM. Particularly, the calculation in the SVM framework need to be projected to a higher dimensional space by using a certain kernel function. Therefore, we use a kernel function $k(\boldsymbol{w}_i, \boldsymbol{w}_j)$ to replace the inner product $\boldsymbol{w}_i \cdot \boldsymbol{w}_j$ and adapt the cosine-similarity measurement to SVM as follows:

$$Sim(\boldsymbol{w}_i, \boldsymbol{w}_j) = \frac{\left|k(\boldsymbol{w}_i, \boldsymbol{w}_j)\right|}{\sqrt{k(\boldsymbol{w}_i, \boldsymbol{w}_i)k(\boldsymbol{w}_j, \boldsymbol{w}_j)}}$$

where, $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are the feature vectors of the i^{th} and j^{th} word respectively. The similarity measurement is similar to [Brinker 2003], but in their work, it is generated based on a version space analysis.

Furthermore, if we use a linear kernel $k(\boldsymbol{w}_i, \boldsymbol{w}_j) = \boldsymbol{w}_i \cdot \boldsymbol{w}_j$, the measurement is the same as the traditional cosine-similarity measurement $\cos\theta = \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \cdot \|\boldsymbol{w}_j\|}$ and may be regarded as a general vector-based similarity measurement.

3.2.2 Similarity Measurement between Named Entities

In this section, we compute the similarity between two named entities given the similarities between words. Regarding a named entity as a word sequence, this work is analogous to the alignment of two sequences. We employ a dynamic time warping (DTW) algorithm [Sakoe and Chiba 1971; Itakura 1975], which is widely applied in the area of speech recognition [Rabiner et al. 1978]. Given a point-by-point distance measurement between a reference contour *R(n)* and a test contour *T(m)*, the DTW algorithm is to find an optimum alignment between *R(n)* and *T(m)* which minimizes the accumulated distance along the path.

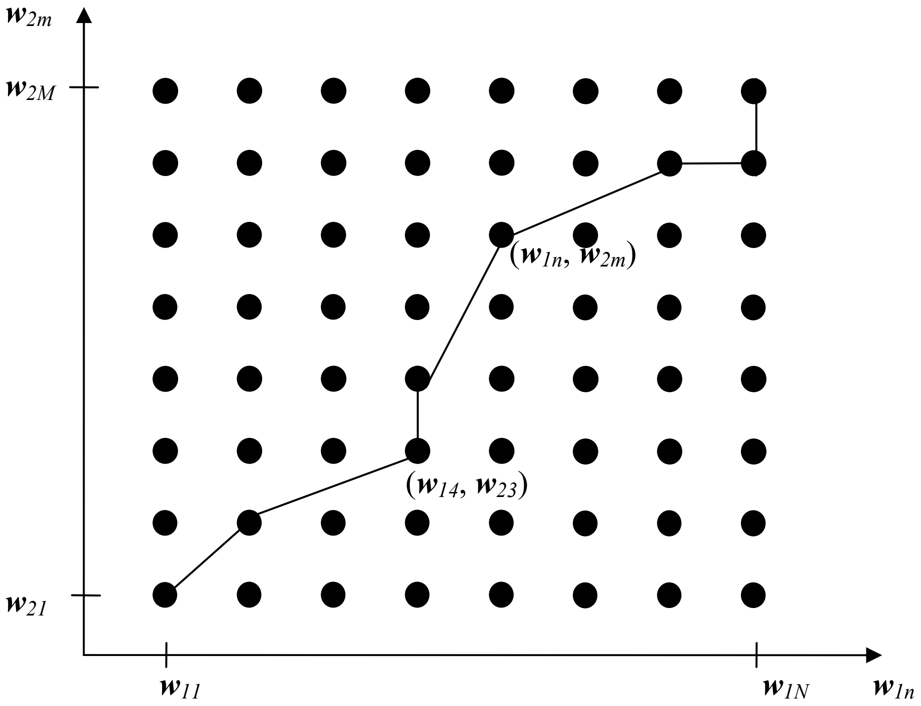


Figure 3.1: Word alignment of two sequences NE_1 and NE_2

Here, we adapt the DTW to the word sequence alignment based on the similarity measurement between words. The greater the similarity between two words, the smaller the distance between them. So, we first convert the similarity measurement to the distance measurement using the function

$Dist(\boldsymbol{w}_i, \boldsymbol{w}_j) = 1 - Sim(\boldsymbol{w}_i, \boldsymbol{w}_j)$, where $Sim(\boldsymbol{w}_i, \boldsymbol{w}_j) \in [0,1]$

Then, the optimal alignment between the words in two sequences NE_1 and NE_2 is chosen to minimize the accumulated distance (or maximize the accumulated similarity) between NE_1 and NE_2, as shown in Figure 3.1. A sketch of the algorithm is as follows.

Let $NE_1 = \boldsymbol{w}_{11}\boldsymbol{w}_{12}\ldots\boldsymbol{w}_{1n}\ldots\boldsymbol{w}_{1N}$, $(n = 1,\ldots, N)$ and $NE_2 = \boldsymbol{w}_{21}\boldsymbol{w}_{22}\ldots\boldsymbol{w}_{2m}\ldots\boldsymbol{w}_{2M}$, $(m = 1,\ldots, M)$ denote two word sequences to be matched. NE_1 and NE_2 consist of M and N words respectively. $NE_1(n) = \boldsymbol{w}_{1n}$ and $NE_2(m) = \boldsymbol{w}_{2m}$. A distance value $Dist(\boldsymbol{w}_{1n}, \boldsymbol{w}_{2m})$ has been given for every pair of words $(\boldsymbol{w}_{1n}, \boldsymbol{w}_{2m})$ within NE_1 and NE_2. An optimum dynamic path $m = map(n)$, which map n onto the corresponding m, is to minimize the accumulated distance $Dist^*$ along the path.

$$Dist^* = \underset{\{map(n)\}}{Min}\{\sum_{n=1}^{N} Dist(NE_1(n), NE_2(map(n))\}$$

An especially powerful technique for determining the optimum path $m = map(n)$ is the method of dynamic programming. Using this technique, the accumulated distance $Dist_A$ to any grid point (n, m) can be recursively calculated as

$$\begin{aligned} Dist_A(n,m) = Dist(\boldsymbol{w}_{1n}, \boldsymbol{w}_{2m}) + Min\{&Dist_A(n-1,m), \\ &Dist_A(n-1,m-1), Dist_A(n,m-1)\} \end{aligned}$$

where, $Dist_A(n,m)$ is the maximum accumulated distance to the grid point (n,m).

We find the solution $Dist^* = Dist_A(N,M)$

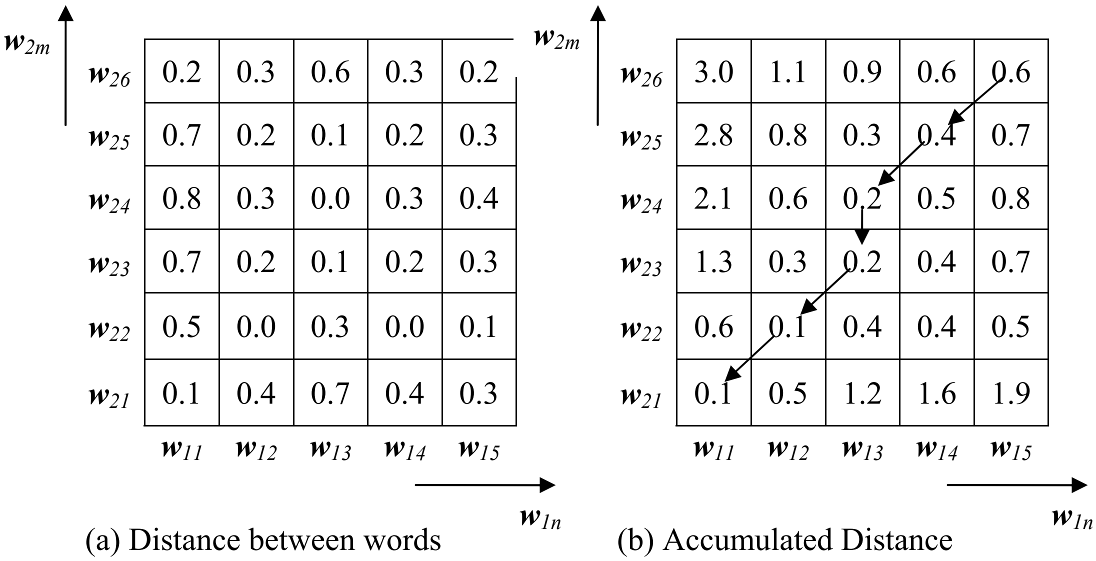
Certainly, the overall distance measurement $Dist^*$ has to be normalized as longer sequences normally give higher score (poorer matching). So, the distance between two sequences NE_1 and NE_2 is calculated as

$$Dist(NE_1, NE_2) = \frac{Dist^*}{Min(N,M)}$$

Finally, we convert the distance measurement to the similarity measurement using $Sim(NE_1, NE_2) = 1 - Dist(NE_1, NE_2)$

Figure 3.2 shows an example of the dynamic time warping algorithm. Figure 3.2 (a) shows the distances between the words $\boldsymbol{w_{1n}}$ in NE_1 and $\boldsymbol{w_{2m}}$ in NE_2. Figure 3.2 (b) shows the accumulated distance to any grid point *(n, m)*. We find the optimal path, shown in Figure 3.2 (c) with the following steps:

1. The accumulated distance from the starting position to any position in the first column is computed.
2. The minimum accumulated distance from the starting position to any position in second column is computed.
3. Repeat the step 2 and obtain the minimum accumulated distance from the starting position to every position in every column.
4. The overall distance *Dist** between the word sequences NE_1 and NE_2 is the value in the right-top grid.
5. Normalize the overall distance *Dist**.



w_{26}	0.2	0.3	0.6	0.3	0.2
w_{25}	0.7	0.2	0.1	0.2	0.3
w_{24}	0.8	0.3	0.0	0.3	0.4
w_{23}	0.7	0.2	0.1	0.2	0.3
w_{22}	0.5	0.0	0.3	0.0	0.1
w_{21}	0.1	0.4	0.7	0.4	0.3
	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}

(a) Distance between words

w_{26}	3.0	1.1	0.9	0.6	0.6
w_{25}	2.8	0.8	0.3	0.4	0.7
w_{24}	2.1	0.6	0.2	0.5	0.8
w_{23}	1.3	0.3	0.2	0.4	0.7
w_{22}	0.6	0.1	0.4	0.4	0.5
w_{21}	0.1	0.5	1.2	1.6	1.9
	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}

(b) Accumulated Distance

n	*m = map(n)*
1	1
2	2
3	3, 4
4	5
5	6

(c) Alignment mapping

$$\begin{aligned} Dist^* &= Dist_A(N, M) \\ &= 0.6 \end{aligned}$$

$$\begin{aligned} Dist(NE_1, NE_2) &= \frac{Dist^*}{Min(N, M)} \\ &= 0.12 \end{aligned}$$

$$\begin{aligned} Sim(NE_1, NE_2) &= 1 - Dist(NE_1, NE_2) \\ &= 0.88 \end{aligned}$$

Figure 3.2: An example of the dynamic time warping algorithm

Furthermore, Figure 3.3 shows an actual example of the similarity measurement between two named entities taken from the experimental data, such as *NF kappa B binding protein* and *Oct 1 binding protein*, using the dynamic time warping algorithm.

(a) Distance between words

protein	0.5	0.5	0.71	0.25	0
binding	0.5	0.5	0.71	0	0.25
1	1	1	0.67	1	1
Oct	0.5	0.5	0.71	0.25	0.25
	NF	kappa	B	binding	protein

(b) Accumulated Distance

protein	2.5	2.5	2.71	1.92	**1.67**
binding	2	2	2.21	**1.67**	1.92
1	1.5	1.5	**1.67**	2.67	2.96
Oct	**0.5**	**1**	1.71	1.96	2.21
	NF	kappa	B	binding	protein

$$Dist^* = Dist_A(N,M) = 1.67$$

$$Dist(NE_1, NE_2) = \frac{Dist^*}{Min(N,M)} = 0.4175$$

$$Sim(NE_1, NE_2) = 1 - Dist(NE_1, NE_2) = 0.4825$$

Figure 3.3: An example of the dynamic time warping algorithm for calculating the similarity between the named entities "*NF kappa B binding protein*" and "*Oct 1 binding protein*"

3.2.3 Representativeness Measurement for Named Entity

Given a set of named entities *NESet* = {*NE*$_1$, …, *NE*$_N$}, the representativeness of a named entity *NE*$_i$ in *NESet* is quantified by its density, which is defined as the average similarity between *NE*$_i$ and all the other named entities *NE*$_j$ ($j \neq i$) in *NESet* as follows.

$$Rep(NE_i) = \frac{\sum_{j \neq i} Sim(NE_i, NE_j)}{N-1}$$

where, *N* is the number of named entities in *NESet*; $NE_i \in NESet$ and $NE_j \in NESet$.

If a named entity has the largest density among all of the name entities in *NESet*, it can be regarded as the centroid of *NESet* and also the most representative example in *NESet*.

3.3 Diversity

The diversity criterion is to maximize the training utility of a batch. We prefer the batch of which the members have high variance to each other. To our knowledge, there is only one work [Brinker 2003] exploring the criterion. In our task, we propose two methods: local and global, to make the examples diverse enough in a batch.

3.3.1 Global Consideration

For the global consideration, we cluster all named entities in *NESet* based on the similarity measurement proposed in Section 3.2.2. Since the named entities in one cluster may be considered quite similar to each other, we select the named entities from different clusters at one time. A K-Means clustering algorithm [Jelinek 1997] is implemented to cluster the named entities in *NESet*, as shown in Figure 3.4.

In each round, we need compute the pair-wise similarities among the named entities within each cluster to get the centroids of the clusters. And then, we need to compute the similarities between each example and the centroids of all clusters to repartition the examples. So, the algorithm is very time-consuming. Based on the assumption that N examples are uniformly distributed among the K clusters, the time complexity of the algorithm is about $O(N^2/K+NK)$ [Tang et al. 2002]. In one of our experiments, the size of the *NESet* (N) is around 17000 and K is equal to 50, so the time complexity is about

$O(10^6)$. For efficiency, we may filter out some of the named entities from *NESet* based on other criteria before clustering them. This will be further discussed in Chapter 4.

```
Given:
  $NESet = \{NE_1, \ldots, NE_N\}$
Suppose:
  The number of clusters is $K$
Initialization:
  Randomly equally partition $\{NE_1, \ldots, NE_N\}$ into $K$ initial clusters $C_j$ ($j = 1, \ldots, K$).
Loop until the number of changes for the centroids of all clusters is less than a threshold
  • Find the centroid of each cluster $C_j$ ($j = 1, \ldots, K$).
    $NECent_j = \arg\max_{NE \in C_j} ( \sum_{NE_i \in C_j} Sim(NE_i, NE))$
  • Repartition $\{NE_1, \ldots, NE_N\}$ into $K$ clusters. $NE_i$ will be assigned to Cluster $C_j$ if
    $Sim(NE_i, NECent_j) \geq Sim(NE_i, NECent_w), w \neq j, w = 1, \ldots, K$
```

Figure 3.4: Global consideration for diversity using K-Means clustering algorithm

3.3.2 Local Consideration

Another consideration, called local consideration, is based on the analysis of the examples in a batch rather than in the whole sample space. For an example candidate, we compare it with all of the previously selected examples in the current batch *BatchSet* one by one. If the similarity between any of them is above a threshold β, the example cannot be added into *BatchSet*. The order of the selection of the example candidates is based on the certain measurement, such as the informativeness measurement, the representativeness measurement and their combination. This local selection method is shown in Figure 3.5. In this way, we avoid selecting too similar examples (similarity value ≥ β) in a batch. The threshold β may be set to the average of the pair-wise similarities among the examples in *NESet*.

```
Given:
  $NESet = \{NE_1, \ldots, NE_N\}$
  BatchSet with the maximal size $K$.
Initialization:
  BatchSet = empty
Loop until BatchSet is full
  • Select $NE_i$ based on certain measure from NESet.
  • RepeatFlag = false;
  • Loop from j = 1 to CurrentSize(BatchSet)
      If $Sim(NE_i, NE_j) \geq \beta$ Then
          RepeatFlag = true;
          Stop the Loop;
  • If RepeatFlag == false Then
      add $NE_i$ into BatchSet
  • remove $NE_i$ from NESet
```

Figure 3.5: Local consideration for diversity

Compared with the K-means algorithm for the global consideration, we find the algorithm for the local consideration only requires $\mathbf{O}(NK+K^2)$ computational time. In one of our experiments (N ≈ 17000 and K = 50), the time complexity is about $\mathbf{O}(10^5)$. So, it is more efficient than the global consideration described in Section 3.3.1.

Chapter 4

ACTIVE LEARNING STRATEGIES

In this chapter, we will study how to combine and strike a proper balance among the multiple criteria, viz. informativeness, representativeness and diversity, to reach the maximum effectiveness of active learning for NER. We propose two active learning strategies to combine the informativeness, representativeness measurements and the diversity consideration. These strategies are based on the different priorities of the criteria and the different degrees to satisfy the criteria.

4.1 Strategy 1

In the Strategy 1, we first consider the informativeness criterion. We choose *M* examples with the most informativeness score from *NESet* to an intermediate set called *INTERSet* (size is *M*). By this pre-selection, we make the selection process faster in the later steps since the size of *INTERSet* is much smaller than that of *NESet*. Then we cluster the examples in *INTERSet* and choose the centroid of each cluster into a batch called *BatchSet*. The centroid of a cluster is the most representative example in the cluster since it has the largest density. Furthermore, the examples in the different clusters may be considered diverse to each other. By this means, we consider the representativeness and diversity criteria at the same time. This strategy is shown in Figure 4.1. One limitation of this strategy is that the clustering result may not reflect the distribution of the whole sample space since we only cluster the examples in *INTERSet* for efficiency. Another limitation is that the representativeness of an example is only evaluated in the clusters rather than *NESet*. If a cluster size is too small, the most representative example in the cluster may not be representative in the whole sample space. Thus, we can not completely avoid selecting outliers. However, this problem can be solved by exploring a more effective clustering algorithms in future work, such as the X-Means clustering algorithm, in which the size of the clusters can be controlled automatically.

```
Given:
  $NESet = \{NE_1, \ldots, NE_N\}$
  BatchSet with the maximal size K.
  INTERSet with the maximal size M
Steps:
  • $BatchSet = \varnothing$
  • $INTERSet = \varnothing$
  • Select M entities with most Info score from NESet to INTERSet.
  • Cluster the entities in INTERSet into K clusters
  • Add the centroid entity of each cluster to BatchSet
```

Figure 4.1: Active Learning Strategy 1

4.2 Strategy 2

Another strategy is shown in Figure 4.2. We combine the informativeness and representativeness criteria using a linear interpolation function $\lambda Info(NE_i) + (1-\lambda) Rep(NE_i)$, in which the *Info* and *Rep* value of NE_i are normalized first. The individual importance of each criterion in the function is adjusted by the trade-off parameter λ ($0 \le \lambda \le 1$) (set to 0.6 in the experiment). Firstly, we select an example candidate NE_i with the maximum value of the function from *NESet*. Secondly, we consider the diversity criterion using the local method described in Section 3.5. We add NE_i to *BatchSet* only if NE_i is different enough from any previously selected examples in the batch. The threshold β is set to the average of the pair-wise similarities among the named entities in *NESet*.

```
Given:
  $NESet = \{NE_1, \ldots, NE_N\}$
  BatchSet with the maximal size $K$.
Initialization:
  $BatchSet = \varnothing$
Loop until BatchSet is full
  • Select $NE_i$ which have the maximum value for the combination function between
    Info score and Density socre from NESet.
  $NE_i = \arg\max_{NE_i \in NESet} Max(\lambda Info(NE_i) + (1-\lambda) Density(NE_i))$
  • RepeatFlag = false;
  • Loop from j = 1 to CurrentSize(BatchSet)
      If $Sim(NE_i, NE_j) \geq \beta$ Then
          RepeatFlag = true;
          Stop the Loop;
  • If RepeatFlag == false Then
      add $NE_i$ into BatchSet
  • remove $NE_i$ from NESet
```

Figure 4.2: Active Learning Strategy 2

Chapter 5

EXPERIMENTATION

5.1 Data set

In this section, we will evaluate the effectiveness of the multi-criteria-based active learning in the SVM-based NER model which recognizes person, location and organization names in the newswire domain using the MUC-6 corpus and protein names in the biomedical domain using the GENIA corpus.

5.1.1 MUC-6 corpus

The Message Understanding Conferences (MUC), sponsored by DARPA in the U.S., defined the task of named entity recognition and sponsored an evaluation to assess the state of the art in the research area. The MUC-6 corpus, including a training data set, a development data set and a test data set, are prepared by human annotators for the MUC-6 evaluation. The corpus contains 378 annotated Wall Street Journal articles (the training data set: 318 articles; the development data set: 30 articles; the test data set: 30 articles). There are three types of annotations in the corpus: named entity, such as *PERSON* names, *LOCATION* names and *ORGANIZATION* names; temporal expression, such as *DATE* and *TIME* expressions; and number expression, such as quantity expressions of *MONETARY* value and *PERCENTAGE*. In our task, the model is to recognize the named entities of *PERSON (PER)*, *LOCATION (LOC)* and *ORGANIZATION (ORG)*.

5.1.2 GENIA corpus

Currently, the GENIA corpus is the largest annotated corpus in the molecular biology domain available to public [Ohta et al. 2002]. In our experiment, two versions of the corpus, viz. GENIA V1.1 and GENIA V2.1 are used.

GENIA V1.1 contains 670 MEDLINE abstracts. The annotations of the biomedical named entities are based on the GENIA ontology, which defines 22 distinct classes of named entities including *MULTI-CELL*, *MONO-CELL*, *VIRUS*, *BODYPART*, *TISSUE*, *CELL-TYPE*, *CELL-COMPONENT*, *CELL-LINE*, *PROTEIN*, *DNA*, *RNA*, etc. In our task, the model is to recognize the named entities of *PROTEIN (PRT)*, therefore, we remove the annotations of other classes of named entities from the corpus first.

GENIA V2.1 contains the same 670 abstracts as V1.1 with the additional part-of-speech tagging. We use the version to train the POS tagger in the biomedical domain (described in Section 2.2.2). Then the tagger is used to assign the POS features in the biomedical NER.

5.2 Experiment Setting

We conduct the experiment of the active learning strategies on NER in both the newswire domain and the biomedical domain. Firstly, we randomly split the whole corpus into three parts: an initial training data set to build an initial model, a test dataset to evaluate the performance of the existing model and an unlabeled data set to select examples. The size of each data set is shown in Table 5.1. Then, we iteratively select a batch of examples following the active learning strategies, require human experts to label them and add them into the training data set. Since previous research works [Kazama et al. 2002; Shen et al. 2003; Zhang et al. 2004] state that NER in the biomedical domain is much more difficult than that in the newswire domain, we assume that NER in the biomedical domain need more training data than that in the newswire domain. Therefore, considering the efficiency of the active learning process, we set the batch size *K* to 50 in the biomedical domain and 10 in the newswire domain. Each example is defined as a named entity and its context words including the previous 3 words and the next 3 words, as described in Section 2.3.

Domain	Class	Corpus	Initial Training Set	Test Set	Unlabeled Set
Biomedical	PRT	GENIA1.1	10 sentences (277 words)	900 sentences (26K words)	8004 sentences (223K words)
Newswire	PER	MUC-6	5 sentences (131 words)	602 sentences (14K words)	7809 sentences (157K words)
	LOC		5 sentences (130 words)		7809 sentences (157K words)
	ORG		5 sentences (113 words)		7809 sentences (157K words)

Table 5.1: Experiment setting of active learning using GENIA V1.1 (PRT) and MUC-6 (PER, LOC, ORG)

5.3 Experiment Result

The goal of our work is to minimize the human annotation efforts to learn a named entity recognizer with the same performance level as a supervised learning model. The supervised learning model is trained on the entire annotated corpus. The performance of the model is evaluated using “precision/recall/F-measure”, in which “precision” is calculated as the ratio of the number of correctly found named entities to the total number of named entities found by our model; "recall" is calculated as the ratio of the number of correctly found named entities to the number of true named entities; and "F-measure" is defined by the formula:

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$

5.3.1 Overall Experiment Results

In this section, we evaluate the active learning strategies by comparing them with a random selection method, in which a batch of examples is randomly selected iteratively, on the GENIA and MUC-6 corpus. Table 5.2 shows the amount of training data needed to achieve the performance of supervised learning using various selection methods, viz. *Random*, *Strategy1* and *Strategy2*.

Domain	Class	Supervised	Random	Strategy1	Strategy2
Biomedical	PRT	223K (F=63.3)	83K	40K	31K
Newswire	PER	157K (F=90.4)	11.5K	4.2K	3.5K
	LOC	157K (F=73.5)	13.6K	3.5K	2.1K
	ORG	157K (F=86.0)	20.2K	9.5K	7.8K

Table 5.2: Overall results of active learning for named entity recognition in the newswire domain and the biomedical domain

From the experiment in the biomedical domain (GENIA corpus), we find:

- The model achieves 63.3 F-measure using 223K words in *Supervised learning*.
- The best performer is *Strategy2* (31K words), requiring less than 40% of the training data that *Random Selection* (83K words) does and 14% of the training data that *Supervised learning* does.
- *Strategy1* (40K words) performs slightly worse than *Strategy2*, requiring 9K more words. It is probably because *Strategy1* cannot avoid selecting outliers if a cluster is too

small.

- *Random Selection* (83K words) requires about 37% of the training data that *Supervised learning* does. It indicates that only the words in and around a named entity are useful for classification and the words far from the named entity may not be helpful.

When we apply the model to the newswire domain (MUC-6 corpus) to recognize person, location and organization names, *Strategy1* and *Strategy2* show a more promising result by comparing with *Supervised learning* and *Random Selection*, as shown in Table 5.2. On average, only about 5% of the training data are needed to achieve the same performance level with *Supervised learning* in the MUC-6 corpus. It is probably because that the named entities are distributed much sparser in the newswire texts than in the biomedical texts.

Furthermore, we find that *Strategy2* always outperforms *Strategy1*. The reason may be that the K-Means clustering algorithm used in *Strategy1* is not so robust, which may result in too small size of a cluster. In this case, we can not avoid selecting outliers. In future work, we may explore a more effective clustering algorithm, which can prevent too small size of the cluster automatically, to overcome the limitation of *Strategy1*.

5.3.2 Effectiveness of Single-Criterion-based Active Learning

In this section, we investigate the effectiveness of the informativeness-based active learning methods in NER. Figure 5.1 shows a plot of training data size versus F-measure achieved by the various informativeness-based measurements proposed in Section 3.1.2:

Info_Avg, *Info_Min* and *Info_InclRate* as well as *Random Selection* in the biomedical NER. In Figure 5.1, the horizontal dashed line is the performance level (63.3 F-measure) achieved by *Supervised learning* (223K words). We find that the three informativeness-based measurements perform similarly and each of them outperforms *Random Selection*. Table 5.3 highlights the various data sizes to achieve the peak performance using these selection methods. We find that *Random Selection* (83K words) on average requires over 1.5 times as much data as the informativeness-based active learning methods (52K words).

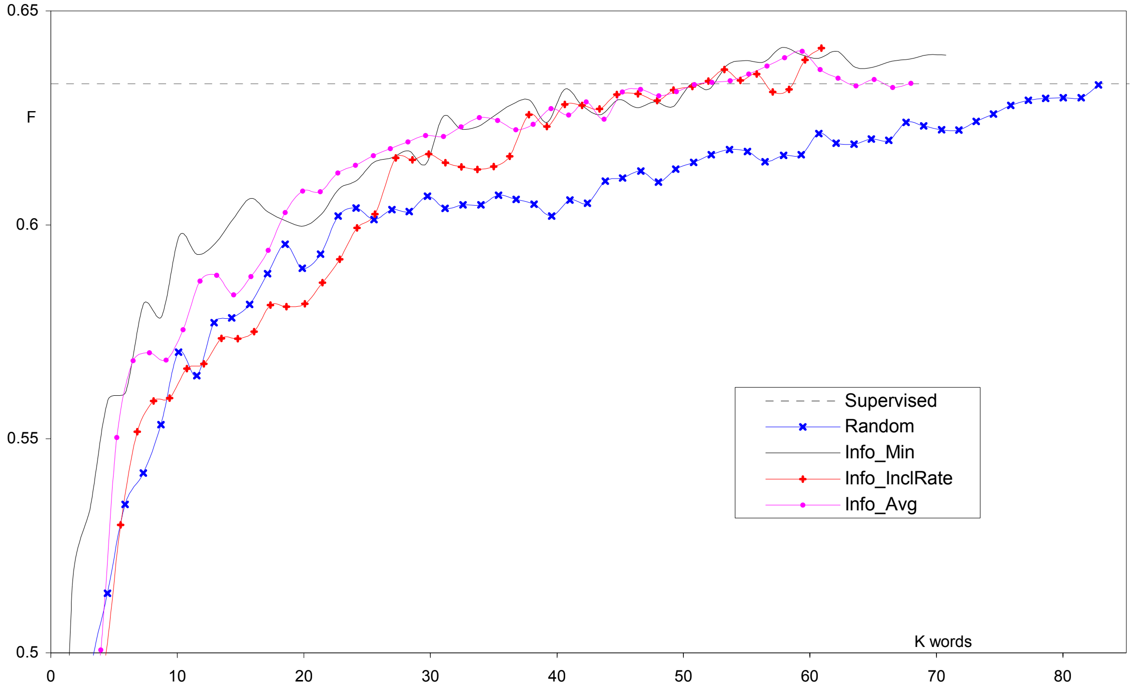


Figure 5.1: Active learning curves: effectiveness of the three informativeness-based active learning methods comparing with random selection in the biomedical named entity recognition

Supervised	Random	Info_Avg	Info_Min	Info_ InclRate
223K	83K	52.0K	51.9K	52.3K

Table 5.3: Comparison of training data sizes for the three informativeness-based active learning methods to achieve the same performance level as supervised learning in the biomedical named entity recognition

5.3.3 Effectiveness of Multi-Criteria-based Active Learning

In addition to the informativeness criterion, we further explore the representativeness and diversity criteria in active learning and incorporate them using two active learning strategies described in Chapter 4. Comparing the active learning strategies with the best result of the single-criterion-based active learning methods *Info_Min*, we are to justify that the representativeness and diversity criteria are also important for active learning.

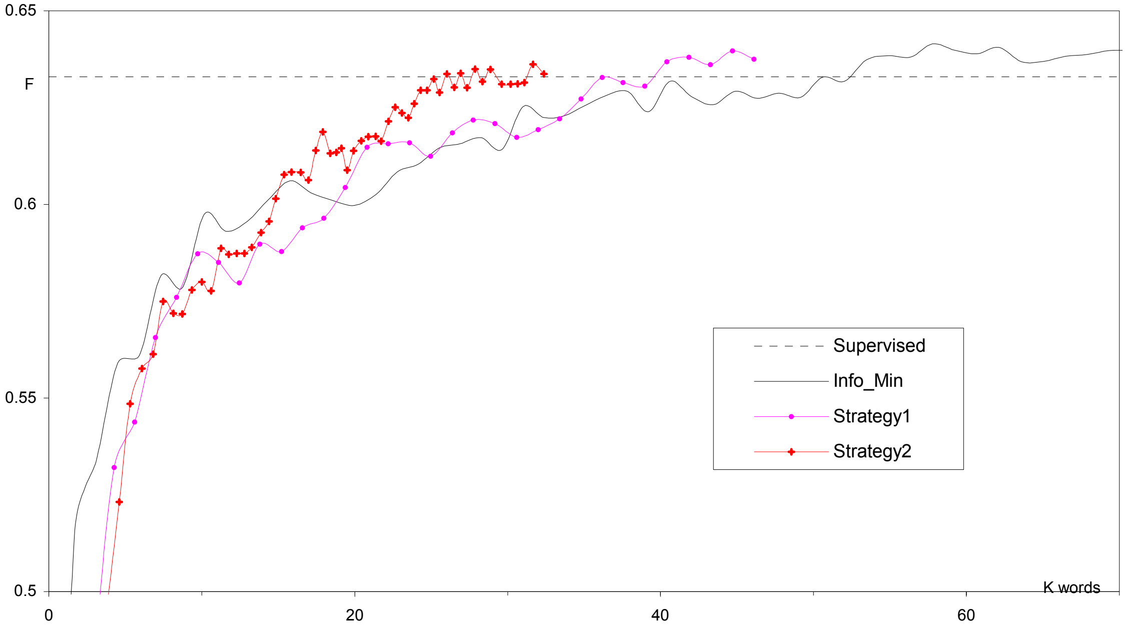


Figure 5.2: Active learning curves: effectiveness of the two multi-criteria-based active learning strategies comparing with the best informativeness-based active learning method (*Info_Min*) in the biomedical named entity recognition

Info_Min	Strategy1	Strategy2
51.9K	40K	31K

Table 5.4: Comparisons of training data sizes for the multi-criteria-based active learning strategies and the best informativeness-based active learning method (*Info_Min*) to achieve the same performance level as supervised learning in the biomedical named entity recognition

Figure 5.2 shows the learning curves for the various methods: *Strategy1*, *Strategy2* and *Info_Min* in the biomedical NER. In the beginning iterations (F-measure < 60), the three methods performed similarly. But with the larger training data set, the efficiencies of *Stratety1* and *Strategy2* begin to be evident. Table 5.4 highlights the final result of the three methods. In order to reach the performance of supervised learning, *Strategy1* (40K words) and *Strategy2* (31K words) require about 80% and 60% of the data that *Info_Min* (51.9K) does. Therefore, we believe that the effective combinations of the informativeness, representativeness and diversity criteria will help to learn the model more quickly and cost less in annotation.

Chapter 6

CONCLUSION

6.1 Conclusions

In this thesis, we study active learning in a more complex natural language processing task, named entity recognition. We propose a multi-criteria-based active learning method to select the most useful examples based on their informativeness, representativeness and diversity in the SVM-NER model. Considering these criteria, we make efforts in four aspects: firstly, we propose three scoring functions to quantify to the informativeness of a named entity. Secondly, we compute the similarity between two named entities and propose a density measurement to evaluate the representativeness of a named entity. Thirdly, we present two considerations (global and local) to satisfy the diversity requirement. Last but not least, we study how to effectively combine these criteria. We propose two combination strategies depending on the different priorities of the criteria. To our best knowledge, this is not only the first work to incorporate the multiple criteria in active learning but also the first work to study active learning for named entity recognition.

The experiments show that the active learning strategies for NER achieve a promising result. Compared with supervised learning, the labeling cost can be significantly reduced by 95% in the newswire domain and 86% in the biomedical domain. Furthermore, we find, in addition to the informativeness criterion, the representativeness and diversity criteria are also useful for active learning. The two active learning strategies, which we propose to combine the criteria, outperform the single-criterion-based active learning method.

6.2 Future Work

Although the current experiment results are very encouraging, some parameters in the experiment, such as the batch size K and λ in the linear interpolation function of the active learning strategy 2, are decided by our experience in the domain. In practical applications, the optimal value of these parameters should be decided automatically in the training process.

Another interesting work is to study when to stop the active learning process. Especially for SVM, the stop criterion may depend on the change of the support vectors.

6.3 Dissemination of Results

This thesis presents a work on the exploration of how to reduce the human annotation cost to learn a named entity recognizer by using active learning. The work on developing a SVM-based named entity recognition system is covered in our paper [Zhou et al. 2004b] accepted by *the EMBO Workshop 2004 on a critical assessment of text mining methods in molecular biology*. In the BioCreAtIve Competition 2003[3], this system achieves the best performance for the task of protein/gene name recognition in on among 15 groups around the world. The detailed information of the features and the evaluation of their effectiveness are covered in the paper [Shen et al. 2003] published in the *Proceedings of*

[3] http://www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html

the ACL 2003 Workshop on Natural Language Processing in Biomedicine, and the paper [Zhou et al. 2004a] accepted by the *Bioinformatics*. The paper [Shen et al. 2004] about some initial exploration on this topic has been published in the *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP) 2004*. Moreover, the paper about the study on the multi-criteria-based active learning has been submitted to the *Conference of the Association of Computational Linguistics (ACL), 2004*.

REFERENCES

[Baeza-Yates and Ribeiro-Neto 1999] R. Baeza-Yates and B. Ribeiro-Neto. 1999. Modern Information Retrieval. *ISBN 0-201-39829-X.*

[Brinker 2003] K. Brinker. 2003. Incorporating diversity in active learning with support vector machines. In *Proceedings of the International Conference on Machine Learning.*

[Dagan and Engelson 1995] I. Dagan and S. A. Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the International Conference on Machine Learning.*

[Engelson and Dagan 1999] S. A. Engelson and I. Dagan. 1999. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research.*

[Finn and Kushmerick 2003] A. Finn and N. Kushmerick. 2003. Active learning selection strategies for information extraction. In *Proceedings of International Workshop on Adaptive Text Extraction and Mining.*

[Freund et al. 1997] Y. Freund, H. S. Seung, E. Shamir and N. Tishby. 1997. Selective sampling using the Query By Committee algorithm. *Machine Learning, 28, 133-168.*

[Hwa 2000] R. Hwa. 2000. Sample selection for statistical grammar induction. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP).*

[Itakura 1975] F. I. Itakura. 1975. Minimum prediction residual principle applied to speech recognition. In *Proceedings of IEEE Transactions on acoustics speech and signal processing. Vol. ASSP-23, pp. 67-72.*

[Jelinek 1997] F. Jelinek. 1997. Statistical Methods for Speech Recognition. *MIT Press.*

[Joachims 1999] T. Joachims. 1999. Making large-scale SVM learning practical. In B. Scholkopf, C. Burges and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning,* MIT Press.

[Joachims, 2002] T. Joachims. 2002. Learning to Classify Text Using Support Vector Machines. *Dissertation, Kluwer.*

[Kazama et al. 2002] J. Kazama, T. Makino, Y. Ohta and J. Tsujii. 2002. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL 2002 Workshop on Natural Language Processing in the Biomedical Domain.*

[Lee et al. 2003] K. J. Lee, Y. S. Hwang and H. C. Rim. 2003. Two-phase biomedical NE recognition based on SVMs. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine.*

[Lewis and Catlett 1994] D. D. Lewis and J. Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the International Conference on Machine Learning.*

[Lewis and Gale 1994] D. D. Lewis and W. A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference.*

[McCallum and Nigam 1998] A. K. McCallum and K. Nigam. 1998. Employing EM and pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning.*

[Ngai and Yarowsky 2000] G. Ngai and D. Yarowsky. 2000. Rule writing or annotation: cost-efficient resource usage for base noun phrase chunking. In *Proceedings of the Association of Computational Linguistics(ACL).*

[Rabiner et al. 1978] L. R. Rabiner, A. E. Rosenberg and S. E. Levinson. 1978. Considerations in Dynamic Time Warping Algorithms for Discrete Word

Recognition. In *Proceedings of IEEE Transactions on acoustics speech and signal processing. Vol. ASSP-26. No. 6.*

[Sakoe and Chiba] H. Sakoe and S. Chiba. 1971. A dynamic programming approach to continuous speech recognition. In *Proceedings of Int. Cong. Acoustics, Budapest, Hungary, paper 20 C 13.*

[Sassano 2002] M. Sassano. 2002. An empirical study of active learning with SVM for Japanese word segmentation. In *Proceedings of the Association of Computational Linguistics (ACL)*.

[Schohn and Cohn 2000] D. Schohn and D. Cohn. 2000. Less is more: active learning with support vector machines. In *Proceedings of International Workshop on Adaptive Text Extraction and Mining.*

[Seung et al. 1992] H. S. Seung, M. Opper and H. Sompolinsky. 1992. Query By Committee. In *Proceedings of the ACM Workshop on Computational Learning Theory.*

[Shen et al. 2003] D. Shen, J. Zhang, G. D. Zhou, J. Su and C. L. Tan. 2003. Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine.*

[Shen et al. 2004] D. Shen, J. Zhang, J. Su, G. D. Zhou and C. L. Tan. 2004. A Collaborative Ability Measurement for Co-Training. To *appear in the 1st International Joint Conference on Natural Language Processing (IJCNLP) 2004.*

[Steedman et al. 2003] M. Steedman, R. Hwa, S. Clark, M. Osborne, A. Sarkar, J. Hockenmaier, P. Ruhlen, S. Baker and J. Crim. 2003. Example selection for bootstrapping statistical parsers. In *Proceedings of Human Language Technology* *conference - North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL).*

[Takeuchi and Collier 2002] K. Takeuchi and N. Collier. 2002. Use of support vector machines in extended named entity recognition. In *Proceedings of the Sixth Conference on Natural Language Learning (CONLL 2002).*

[Tang et al. 2002] M. Tang, X. Luo and S. Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of the Association of Computational Linguistics (ACL).*

[Thompson et al. 1999] C. A. Thompson, M. E. Califf and R. J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of the International Conference on Machine Learning.*

[Tong and Koller 2000] S. Tong and D. Koller. 2000. Support vector machine active learning with application to text classification. *Journal of Machine Learning Research.*

[Vapnik et al. 1995] V. N. Vapnik. 1995. The nature of statistical learning theory. Springer-Verlag, New York.

[Zhang et al. 2004] J. Zhang, D. Shen, G. D. Zhou, J. Su and C. L. Tan. 2004 Enhancing HMM-based Biomedical Named Entity Recognition by Studying Special Phenomena. To *appear in the Journal of Biomedical Informatics, Special Issue on Natural Language Processing in Biomedicine: Aims, Achievements and Challenge.*

[Zhou and Su 2002] G. D. Zhou and J. Su. 2002. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of the Association of Computational Linguistics (ACL).*

[Zhou et al. 2004a] G. D. Zhou, J. Zhang, J. Su, D. Shen and C. L. Tan. 2004 Recognizing Names in Biomedical Texts: A Machine Learning Approach. To *appear in the Bioinformatics.*

[Zhou et al. 2004b] G. D. Zhou, D. Shen, J. Zhang, J. Su and C. L. Tan. 2004. To *appear in the EMBO Workshop 2004 on a critical assessment of text mining methods in molecular biology.*