

Maximum Margin Active Learning for Sequence Labeling With Different Length ^{*}

Haibin Cheng¹, Ruofei Zhang², Yefei Peng², Jianchang Mao², Pang-Ning Tan¹

¹ CSE Department, Michigan State University
East Lansing, MI 48824
{chenghai,ptan}@msu.edu

² Yahoo, Inc.
2821 Mission College Blvd, Santa Clara, CA 95054
{rzhang,ypeng,jmao}@yahoo-inc.com

Abstract. Sequence labeling problem is commonly encountered in many natural language and query processing tasks. SVM^{struct} is a supervised learning algorithm that provides a flexible and effective way to solve this problem. However, a large amount of training examples is often required to train SVM^{struct} , which can be costly for many applications that generate long and complex sequence data. This paper proposes an active learning technique to select the most informative subset of unlabeled sequences for annotation by choosing sequences that have largest uncertainty in their prediction. A unique aspect of active learning for sequence labeling is that it should take into consideration the effort spent on labeling sequences, which depends on the sequence length. A new active learning technique is proposed to use dynamic programming to identify the best subset of sequences to be annotated, taking into account both the uncertainty and labeling effort. Experiment results show that our SVM^{struct} active learning technique can significantly reduce the number of sequences to be labeled while outperforming other existing techniques.

Key words: Active Learning, Struct Support Vector Machine, Uncertainty, Sequence Labeling, Natural Language Processing, Subphrase Generation

1 Introduction

Sequence labeling is the task of mapping an ordered list of inputs to a sequence of output tags. It has many practical applications in natural language processing such as named entity recognition, part-of-speech tagging, shallow parsing, and text chunking. Another potential application, which is investigated in this study, is the subphrase generation problem. The goal of subphrase generation in query processing is to find subphrases in a query that maximally preserve the user's intent. Unlike the classification of record-based data, sequence labeling depends not only on the features extracted from the input sequence but also on its previous output tags. Many algorithms have been proposed in the literature to address this problem, including Conditional Random Field [8], Hidden

^{*} The work was performed when the first author worked as a summer intern at Yahoo, Inc.

Markov Model [12] and Maximum Entropy Markov Model [9]. More recently, a maximum margin method known as Struct Support Vector Machine (SVM^{struct}) [19] was proposed to solve this problem. SVM^{struct} generalizes multi-class Support Vector Machine learning to complex data with features extracted from both inputs and outputs. An empirical study in [11] has demonstrated that SVM^{struct} outperforms other existing methods in the sequence labeling task.

A known problem in supervised learning tasks such as sequence labeling is the difficulty of acquiring labeled examples. The size of training data available is often limited because labeling examples can be very expensive. Labeling a sequence is also more challenging because the output tag depends on both the input and previous output tags. As a result, the tags of a sequence must be determined as a whole, rather than individually for each input element. Active learning may help to address this problem by selecting a small subset of examples for labeling from the large pool of unlabeled sequences available. By selecting the most informative examples, active learning can significantly reduce the required size of training data while maintaining comparable level of performance. However, the definition of “informative” varies for different algorithms and applications. One commonly used method is to select examples with largest uncertainties. In this paper, we treat each sequence as a whole for labeling and propose two strategies to measure the uncertainty of sequences under the SVM^{struct} framework, referred as simple uncertainty (SU) and most-possible-constraint-violation method ($MPSV$).

Another challenge of active learning for sequences is that we must consider the effort needed to annotate different sequences. Clearly, labeling a long sequence takes more effort than labeling a short sequence. For example, in named entity recognition problem there may be a large variation in the length of sentences from one to another, which will make the effort required for labeling different sequences different. Besides, the effort for labeling sequences from different domain knowledge may also be different. For example, in subphrase generation problem, queries submitted by users coming from different specialty such as biology, which will make the labeling difficult for labeler with different background such as computer science. However, such kind of complexity in sequence is hard to quantify, in this paper we only use sequence length as a measure of effort. Furthermore, previous active learning system in sequence labeling assumes the input to be the number of sequences we want to select. However, this may not be suitable for the active learning in sequence labeling since the number of sequences did not really reflect the effort we need to put on labeling because of the sequence length problem. This paper defines the effort that can be spent in labeling sequence as the total length of all selected sequences instead of the total number of of all selected sequences. We propose a dynamic programming approach that can select the best combination of sequences for labeling which maximize the total uncertainty while restricting the effort that can be spent.

The rest of the paper is organized as follows. In Section 2, we introduce the background of our work including the sequence labeling problem, active learning and SVM^{struct} algorithm for sequence labeling. In Section 2.2, we propose to use SVM^{struct} for sequence active learning. In Section 4, we present the problem of active learning in labeling sequences with different length and propose to solve it by dynamic programming. Section 5 shows some experiment results and we made our conclusion in Section 6.

2 Background

In this section, we will introduce some background on sequence labeling, active learning and Struct Support Vector Machine.

2.1 Sequence Labeling Problem

Joey, an employee in Yahoo, has been living in California for 10 years. [Person] [] [] [Organization] [] [] [] [] [Location] [] [] []	Query : where can I buy DVD player online? 0 0 0 1 1 1 0 Subphrase : buy DVD player
------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------

Fig. 1. Example of named entity recognition(left) and subphrase generation problem(right).

Sequence labeling is a common problem with many applications in many areas such as named entity recognition [15], POS tagging [16], text chunking [16], etc. In our work, we also investigate the problem of subphrase generation, as a sequence labeling problem. The left panel of Figure 1 shows an example of named entity recognition problem, which labels text elements as predefined categories such as the names of persons, organizations or locations. The right panel of Figure 1 shows another example of subphrase generation problem. Label “0” means that the query word is dropped from the original query and “1” means “keep”. As a result, the remaining part of the given query “where can I buy DVD player online?” becomes “buy DVD player”. **Definition 1** and **Definition 2** give the formal definition of sequence and sequence labeling problem.

Definition 1 [Sequence]: A sequence \mathbf{x} is an ordered list of elements $\mathbf{x} = (x^1, x^2, \dots, x^t)$.

Definition 2 [Sequence Labeling]: Given a sequence of inputs \mathbf{x} , the sequence labeling problem is trying to label it with a sequence of tags $\mathbf{y} = (y^1, y^2, \dots, y^t)$, where each tag y^i belongs to a tag set D with $|D|$ tags.

One simple way to solve the sequence labeling problem is to use traditional classification algorithm such as SVM [2], which treats each element in the sequence as one example. However, it requires the features extracted only depend on the inputs \mathbf{x} , which is not true in sequence labeling problem. The features extracted for sequence labeling not only depends on the inputs \mathbf{x} , but also depends on the outputs \mathbf{y} . The feature vector for a sequence (\mathbf{x}, \mathbf{y}) is represented as a joint feature mapping vector $\phi(\mathbf{x}, \mathbf{y})$. The definition of ϕ depends on the nature of different applications. One example feature for the subphrase matching problem would be “previous word is dropped \rightarrow current word is kept”, which represents the transition from previous tag “0” to current tag “1”.

Now assume that we have a training sequence set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ with its corresponding tag sequence set $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$. We are interested in learning a mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Instead of learning f directly, the strategy is to transform the problem into learning a discrimination function F over the joint mapping of input and output:

$$\mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$$

Given a test sequence \mathbf{x} , its prediction is achieved by maximizing F over the response variable. The generalized form of the hypotheses f becomes:

$$f(\mathbf{x}; \mathbf{w}) = \arg \max_{y \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \quad (1)$$

where \mathbf{w} is the parameters to be learned. Using the joint feature vector $\phi(\mathbf{x}, \mathbf{y})$, it can be further formulated as:

$$f(\mathbf{x}; \mathbf{w}) = \arg \max_{y \in \mathcal{Y}} F(\phi(\mathbf{x}, \mathbf{y}), \mathbf{w}) \quad (2)$$

Note that many existing methods for sequence labeling problem can be explained in the above framework. For example, the function form F that are maximized in the above prediction function represents the conditional probability $P(\mathbf{y}|\mathbf{x})$ in conditional random field [8], Hidden Markov Models [12] and Maximum Entropy Markov Models [9]. The detailed difference between these methods are illustrated in [11] and not discussed in our paper. In this work we will mainly focus on SVM^{struct} with prediction function:

$$f(\mathbf{x}; \mathbf{w}) = \arg \max_{y \in \mathcal{Y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) \quad (3)$$

SVM^{struct} has been proved to outperform all the other methods according to a recent empirical study in [11].

2.2 Active Learning in Sequence Labeling

Active learning is a process to actively select a subset of unlabeled data to query for labeling. There are many different frameworks of active learning. Among those, stream based active learning [4] selectively queries the examples from a stream for labels. The advantage of stream based active learning methods is that it can make fast and instant decision. However, stream based active learning only considers one example at a time and fails to take the underlying distribution of the whole unlabeled data into consideration. In our work, we are mostly concerned with pool based active learning methods, which selects the best examples from the entire pool of unlabeled data. Pool based active learning methods have been used to reduce the number of training data required to obtain an certain level of performance [1] [15] or to improve the overall performance [3].

The most challenging problem in active learning is to choose an appropriate measure as criteria for selecting the best examples from a pool of data. Many criteria have been developed for this purpose. Divergence based methods such as query by committee method [4] select examples with the largest disagreement between different models and aims to minimize the classification variance. Error-reduction-based active learning attempts to select the examples with minimal expected error for querying to minimize classification error over the test data [13]. Another big family of active learning is uncertainty based methods, which use model confidence as a criterion for selecting best examples and thus differ for different models. For example, Jing et al. use entropy based methods [7] to select unlabeled examples for the application of image retrieval.

[18] propose three margin based methods in Support Vector Machine to select examples for querying which reduce the version space as much as possible. The underlying distribution of the unlabeled data is also investigated to choose the most representative examples [10].

Active learning on simple data has been well studied, however, there is not much work for more complex data set such as sequences. Active learning for sequence labeling is even more important because it is very expensive to label a long sequence. One challenging problem in sequence active learning is that we must select the whole sequence to query for labeling since the labeling of its elements depends on the context information. Previous methods summarized above can only be used to select one element in the sequence which can not be labeled without context information. [15] proposes a multi-Criteria-based active learning for the problem of named entity recognition using Support Vector Machine. However, they assume that the features depend only on the input sequence and are independent of the output tag sequence. One work that considers the input-output joint feature map is by [17], which utilizes conditional random field as the underlying model for sequence active learning. To the best of our knowledge, non work has been conducted using Struct Support Vector Machine, which has shown its potential improvement over CRF and other sequence learning algorithm such as HMMs [12] and Maximum Entropy [11]. Another common weakness of previous work on active learning for sequence labeling is that they did not take into account the difference in labeling effort for sequences different length. Labeling long sequence usually requires much more efforts than short sequence in terms of the time spent by the labeler. In this paper, we propose some simple but effective measurement for sequence active learning using Struct Support Vector Machine [19] as well as some dynamic programming algorithm to solve the length difference problem.

In this section, we reviewed some previous work on active learning for simple data and sequence data. In the next section, we will present previous work on SVM^{Struct} algorithm for structure prediction.

2.3 Struct Support Vector Machine for Sequence Labeling

The sequence labeling problem can be solved by multi-class Support Vector Machine [20] by treating each tag sequence as a class. For a tag sequence $\mathbf{y} = (y^1, y^2, \dots, y^t)$ with t elements and $|D|$ possible tags for each element $y^i, i = 1, \dots, t$, the possible number of classes is $|D|^t$. When the sequence length t is large, the huge number of classes makes the multi-class Support Vector Machine infeasible. Given a set of training sequences $(\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, Struct Support Vector Machine solves this problem by exploring the underlying structure with \mathcal{Y} . In Struct Support Vector Machine, the margin of each training example $(\mathbf{x}_i, \mathbf{y}_i)$ is defined as:

$$\forall i : r_i = \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i} \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) \quad (4)$$

By maximizing the $\min_i r_i$ and fixing the functional margin ($\min_i r_i \geq 1$), we find a unique solution of \mathbf{w} . Thus the hard margin of SVM^{struct} learns the parameter vector

w in the training phase by solving the following optimization function:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s. t.} \quad & \forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \\ & \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i} \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) \geq 1, \end{aligned} \quad (5)$$

The nonlinear constraint in the above equation is equivalent to a set of linear constraints:

$$\forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) \geq 1 \quad (6)$$

which makes the objective function into the form:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s. t.} \quad & \forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \\ & \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) \geq 1 \end{aligned} \quad (7)$$

The above solution assumes the training set is separable. Similar to standard SVM, a slack variable ξ_i is introduced for each sequence \mathbf{x}_i to allow errors. Another weakness of the above solution is the assumption of zero-one classification loss, which is infeasible for sequence labeling problem where $|\mathcal{Y}|$ is large. To allow arbitrary loss function $\Delta(\mathbf{y}_i, \mathbf{y})$, one way is to rescale the margin. By taking error relaxation and loss function into consideration, the final optimization problem is formulated as:

$$\begin{aligned} \min_{\mathbf{w}, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^n \xi_i \\ \text{s. t.} \quad & \forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \\ & \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}) \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \quad (8)$$

where $\Delta(\mathbf{y}_i, \mathbf{y})$ is the loss function which is calculated as the number of different tags between \mathbf{y}_i and \mathbf{y} in our paper. Since the number of constraints is $n|\mathcal{Y}|$, which is large for sequence labeling problem. *SVM^{struct}* [19] solves this problem in polynomial time by keeping a small working set of constraints and in each iteration adding the most violated constraint as following:

$$\max_{\mathbf{y} \in \mathcal{Y}} \Delta(\mathbf{y}_i, \mathbf{y}) - (\mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y})) \quad (9)$$

After learning the parameter w , the tag sequence $\hat{\mathbf{y}}$ for a test sequence \mathbf{x} is predicted by solving the following argmax function using Viterbi search algorithm [5]:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) \quad (10)$$

SVM^{struct} is a flexible and effective solution for the sequence labeling problem and has been proved empirically to outperform other sequence labeling algorithm such as CRF [8], HMM [12] and Maximum Entropy [9]. In this paper, we will investigate the usage of *SVM^{struct}* in active learning setting.

3 Active Learning By SVM^{Struct}

In the last section, we have introduced the background of SVM^{Struct} for sequence labeling problem and some previous work on sequence labeling and active learning. In this section, we will propose to use SVM^{Struct} for active learning. From the previous

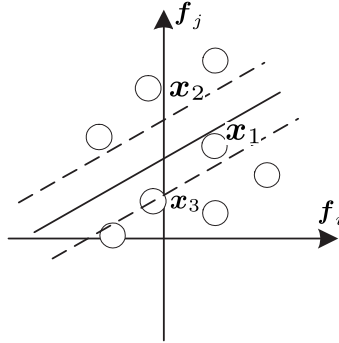


Fig. 2. Simple margin method will select unlabeled data x_1 for querying, which lies closest to the hyperplane.

work on active learning [7] [18], measurement of uncertainty has played an important role in selecting the most valuable examples from a pool of unlabeled data. In the framework of Support Vector Machine [18], three methods have been proposed to measure the uncertainty of simple data, which are referred as simple margin, MaxMin margin and ratio margin. Simple margin measures the uncertainty of an simple example x by its distance to the hyperplane w calculated as:

$$|w \bullet \varphi(x)| \quad (11)$$

As illustrated in Figure 2, examples lying closer to the hyperplane are assigned with larger uncertainty score. This is consistent with the intuition that examples close to the hyperplane are classified with lower confidence. These examples are considered as valuable examples since they have higher probability to be misclassified and thus more informative to be selected for further training.

However, labeling an element in a sequence by itself is almost infeasible in most sequence labeling applications because of the requirement for context information. In most situations we have to consider a whole sequence as an unit for uncertainty measurement and active selection. Given a pool of unlabeled sequences, $\mathcal{U} = \{s_1, s_2, \dots, s_m\}$, the goal of active learning in sequence labeling is to select the most valuable sequences from the pool. Similar to regular Support Vector Machine, a straightforward way to measure the uncertainty of a sequence s is by its prediction score. The prediction score $w^T \phi(s, y)$ measures the certainty of labeling test sequence s using the tag sequence y .

The simple uncertainty for sequence s is then calculated in SVM^{struct} as:

$$UC(s) = \exp(-\max_{y \in \mathcal{Y}} \mathbf{w}^T \phi(s, \mathbf{y})) \quad (12)$$

which is based on the negative value of the prediction score given by formula (10). Note that the features in sequence labeling not only depend on the input sequence s , but also depends on the output \mathbf{y} . As a result, we must run Viterbi algorithm to get the uncertainty score for each sequence in the pool of unlabeled sequences U . Finally, the sequences with larger uncertainty are selected as valuable examples to add to the training set for further learning. We refer this method as simple uncertainty(SU) in this paper.

One drawback of the simple prediction score is its ignorance of the underlying score distribution among different classes and only use the maximum score as a measure of certainty. Here we propose another method which defines the uncertainty of a sequence x as:

$$UC(s) = \exp(-\max_{\substack{\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y} \\ \mathbf{y}_1 \neq \mathbf{y}_2}} (\mathbf{w}^T \phi(s, \mathbf{y}_1) - \mathbf{w}^T \phi(s, \mathbf{y}_2))) \quad (13)$$

which can be further formulated as:

$$UC(s) = \exp(\min_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^T \phi(s, \mathbf{y}) - \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^T \phi(s, \mathbf{y})) \quad (14)$$

We measure the uncertainty of an sequence s as the difference between the minimum prediction score and the maximum prediction score, which is actually the most possible violated constraint for a sequence s that can be added into the optimization problem. We refer this method as the most-possible-constraint-violation method (MPSV) in this paper.

The two methods SU and MPSV proposed here are used to calculate the uncertainty for each test sequence s . The test sequences with maximum uncertainty score are selected as the most informative sequences. These sequences are submitted to the labeler to query for labeling and further added into the training set. The detailed algorithm is presented in Figure 3. However, here we treat each sequence the same disregard with their length. Since labeling sequences with different length requires different effort, we will propose a dynamic programming algorithm to solve it in the next section.

4 Active Learning for Sequence Labeling with Different Length

In previous sections, we have introduced some strategies to select the most valuable sequences from a pool of unlabeled sequences for querying in SVM^{struct} based on the measurement of uncertainty for a sequence. Long sequences tend to have larger value in terms of prediction score as in formula (10) and thus smaller score as in formula (12) than short sequences. One simple way to solve this problem of comparing sequences with different length is by normalization. However, we still did not consider the different effort needed in labeling sequences with different length. Actually, some other

factors should also be concerned in measuring labeling effort such as context knowledge difficulty. However, we ignore those factors due to the difficulty in quantizing context knowledge. For example, given two queries “hotel in LA” and “car loan for people who have filed bankruptcy” in subphrase generation problem, the second query with length of 8 takes longer time for the labeler to label than the first query with length of only 3. This problem is even more severe in named entity recognition problem since there is huge difference in the length of sentences. Furthermore, existing sequence active learning [15] framework usually selects a predefined number of sequences, which is not appropriate since the efforts to be spent in labeling is restricted by the total length of selected sequences. To address these problems, we make the following two assumptions:

- The effort for a labeler that can be spent for labeling a set of sequences is defined as the total number of elements in the sequence set instead of the total number of sequences.
- The effort needed to label a sequence s is related and only related to the length of the sequence.

The first assumption changes the output for the previous sequence active learning system and is able to measure the effort in labeling effectively. The second assumption gives a simple definition to measure the efforts needed in labeling sequences with different length. These two assumptions brings out another concern about the choice between one longer sequence and several shorter sequences.

Here we formulate the problem as follows. Given a pool of unlabeled sequences $\mathcal{U} = \{s_1, s_2, \dots, s_m\}$ with uncertainty score $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$, we define the effort needed for corresponding sequences as:

$$\mathcal{L} = \{l_1, l_2, \dots, l_m\}$$

where $l_i = |s_i|, i = 1, 2, \dots, m$. We also define the effort that can be spent in labeling sequences by the labeler as L . The goal is to utilize as much effort that can be spent as possible while maximizing the total uncertainty, which leads to the following objective function:

$$\begin{aligned} & \max\left(\sum_{i=1}^m e_i f_i\right) & (15) \\ & s.t. \sum_{i=1}^m e_i l_i \leq L \end{aligned}$$

where $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ is the indication vector with:

$$e_i = \begin{cases} 0 & \text{sequence } i \text{ is not selected.} \\ 1 & \text{sequence } i \text{ is selected.} \end{cases} \quad (16)$$

This is a NP-hard problem[6], which is approximately solved by dynamic programming algorithm in this paper. We define $K(i, j)$ as the maximum total uncertainty that can be

SVM^{struct} Active Learning Algorithm for Sequences with Different Length:
<p>Input: A small set of training sequences $(\mathcal{X}, \mathcal{Y}) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, a large pool of unlabeled sequences $\mathcal{U} = \{s_1, s_2, \dots, s_m\}$ and a predefined number of words that can be labeled L</p> <p>Output: A vector $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ with $e_i = 1$ indicating the selection of sequence i.</p> <p>Method:</p> <ol style="list-style-type: none"> Learn the parameter vector w by the standard SVM^{struct} algorithm with training data $(\mathcal{X}, \mathcal{Y})$. Calculate the uncertainty scores $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$ for each sequence in the pool of unlabeled sequences \mathcal{U} by $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ Viterbi Search according to: ${}^1 f_i = \exp(-\max_{y \in \mathcal{Y}} w^T \phi(s_i, y))$ ${}^2 f_i = \exp(\min_{y \in \mathcal{Y}} w^T \phi(s_i, y) - \max_{y \in \mathcal{Y}} w^T \phi(s_i, y))$ Solve formula (16) by dynamic programming to learn the indication vector $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ and send sequence s_i to query for labeling if $e_i = 1$.

Fig. 3. The Active Learning Algorithm for Sequences with Different Length

achieved with total number of elements less than or equal to j using sequences up to i . The recursive function is defined as:

$$\begin{aligned}
K(0, j) &= 0 \\
K(i, 0) &= 0 \\
K(i, j) &= K(i-1, j), \text{ if } l_i > j \\
K(i, j) &= \max(K(i-1, j), f_i + K(i-1, j-l_i)), \text{ if } l_i \leq j
\end{aligned} \tag{17}$$

$K(m, L)$ is final uncertainty we get for m input sequences and desired effort L . This algorithm is a generalized algorithm that can be applied into any sequence active learning framework with different algorithms or different definitions of uncertainty scores. In this paper, we apply this algorithm into the SVM^{struct} active learning framework, which is described in Figure 3

In this section, we give a clear definition about the effort needed to label a sequence. We also redefine the effort that can be spent for labeling as the total number of elements instead of the number of sequences. Our sequence active learning algorithm utilizes the SVM^{struct} framework and takes the effort needed in sequence labeling into account to select a subset of sequences with maximum uncertainty from a pool of unlabeled sequences. In the next section, we will conduct some experiments to evaluate our methods.

5 Experimental Result

In this section, we will conduct some experiments on real data sets to prove the effectiveness of our active learning algorithm.

5.1 Experiment Setup

We applied our algorithm to three data sets in our experiment. The first two data sets come from named entity recognition shared task of CoNLL-2002[14]. One is Spanish data (**ESP**), which is a collection of news wire articles made available by the Spanish EFE News Agency. Another is Dutch data **NED**, which consist of four editions of the Belgian newspaper "De Morgen" of 2000. The task is to label each word in the sentence using some predefined entity tags such as person names (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC) with a B ahead of them denoting the first item of a phrase and an I any non-initial word. The third data we are using is collected from the query subphrase matching project (**QSPM**) of Yahoo Sponsor Search. Given a query by a typical search engine user, the goal is to generate subphrases that preserve the user intent as well as match the bidded terms submitted by advertisers. There are two tags: "KEEP"("1") and "DROP"("0") for each position.

For each position in the sequence, we extract its context features such as "current word is", "previous word is", "next word is" and so on. We also used tag transition features such as "previous tag to current tag". Some word features such as prefix and suffix are also used based on the language of the data such as "th" for English data. We did not employ any feature selection methods in our experiments. For the **DER** data, the Part-Of-Speech tags are also utilized as grammatical features. All our experiments were conducted on a Linux server with 7.2GHz CPU and 15GB RAM.

5.2 Overall Performance

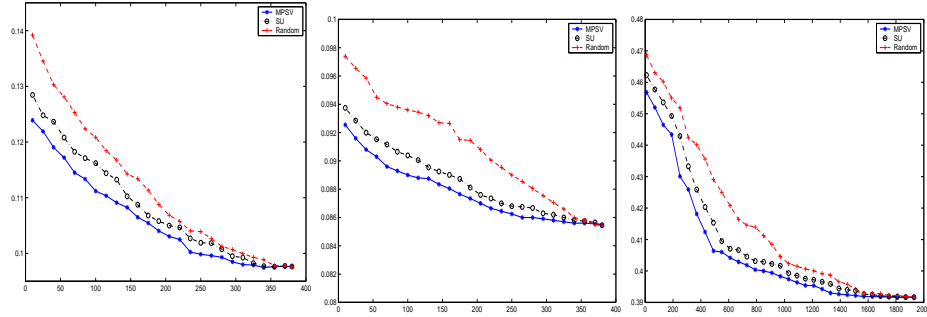


Fig. 4. The average error for ESP data set by three active learning uncertainty measurements

In this experiment, we compare the most-possible-constraint-violations method (**MPSV**) and simple uncertainty(**SU**) method with the random method. To alleviate the length problem in sequence active learning, we select a subset of sequences from the training data, which has the same length. For each data set, we run four experiments, each on a different length selected from the training data. For **NED** data, we select all the sequences with length 12,13,14 and 15 in each experiment. For **ESP** data, we select all

the sequences with length 42, 43, 44, 45 in each experiment. For the **QSPM** data, we select all the sequences with length 3, 4, 5, 6. For the **NED** and **MPSV** data set, we select 400 sequences at each length. The first 10 are used for initial training. The pool of the remaining 390 sequences is for active selection. Each time we select 15 sequences and the result is reported as the average error rate of different length. For the **QSPM** data, we select 1930 sequences at each length. The first 10 sequences are used for initial training. The pool of the remaining 1920 sequences is for active selection. Each time we select 60 sequences and the result is reported as the average error rate of different length on the test set.

Figure 4 shows the results for the three methods in the three data sets **ESP**, **NED** and **QSPM**. The x-axis denotes the number of unlabeled sequences selected to query for labeling. The y-axis represents the average error rate, which is calculated in the word level as follows:

$$Error\ Rate_{\{Word\ Level\}} = \frac{Total\ number\ of\ correctly\ tagged\ words}{Total\ number\ of\ words} \quad (18)$$

We observe from the Figure 4 that both **MPSV** and **SU** methods outperform random approach on all three data sets. Also **MPSV** performs better than **SU**, which means that **MPSV** is a better way to measure uncertainty for SVM^{struct} . Furthermore, the gap between the **MPSV** and other two methods seems very large when the number of selected sequences is small. It means that **MPSV** serves as a good criteria that only a small number of sequences are needed to get good performance. In this experiment, all the sequences are of the same length to compare three methods and we are aiming to select a predefined number of sequences. In the next section, we conduct experiments on sequences with efferent length utilizing the dynamic programming algorithm.

5.3 Selecting Sequences with Different Length

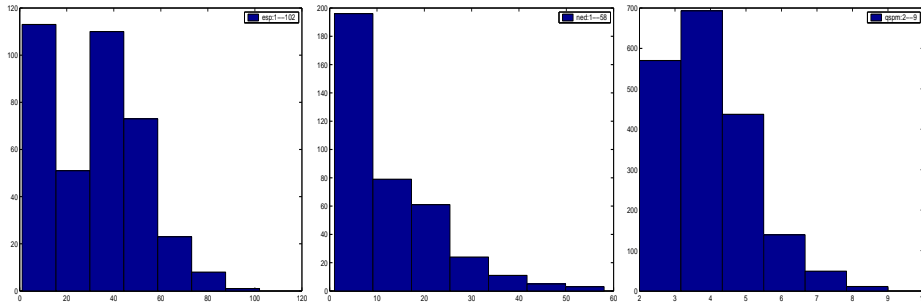


Fig. 5. The length distribution for the three data sets: ESP (left), NED (middle) and QSPM (right).

In the last section, we have compared two uncertainty measure (**MPSV**) and (**SU**) in SVM^{struct} with random method to select sequences with the same length. In this

section, we will conduct experiments on our new active learning system which takes the effort in labeling into consideration. We select 400 sequences randomly from the original data set for **NED** and **ESP** separately with different length. We select 1930 sequences randomly from original **QSPM** data set with different length. Figure 5 shows the histograms of length distribution for the three sample data sets. As we can see, the length of sequences varies from 1 to 102 for **ESP** data set, from 1 to 58 for the **NED** data set and from 2 to 9 for the **QSPM** data. The wide spread of the length distribution elaborates our concern on different effort spent in labeling sequences with different length.

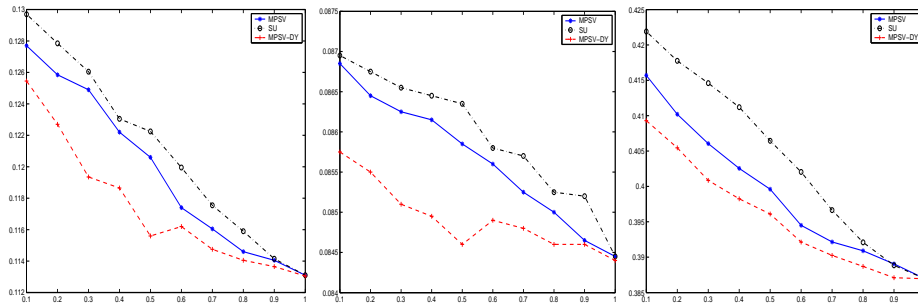


Fig. 6. The average error for the three data sets (EPS,NED,QSPM from left to right) by the dynamic active learning system using **MPSV-DY** as uncertainty measurement and existing active learning methods using **MPSV** and **SU** as uncertainty score.

The input here is the percentage of words we want to select from the pool of sequences instead of the number of sequences. The percentage of words to be selected is the effort that can be spent in labeling sequences by the labeler. The baseline here is the previous active learning system, which ranks the sequences in the pool based on the normalized uncertainty score and selects the sequences with highest scores. We compare our dynamic active learning methods with previous active learning methods on the three data sets. Since **MPSV** has shown its improvement over random method and **SU** methods. We use **MPSV** as the uncertainty score measurement for our dynamic active learning algorithm.

Figure 6 reports the error rate on the ESP, NED and QSPM data sets from left to right comparing our new active learning system with previous active learning system. X-axis is the percentage of words we want to select for labeling which is used to illustrate the effort that can be spent in labeling. Y-axis represents the error rate after querying the selected sequences for labeling and retrain the model with the new labeled sequences added. The result on the esp data shows that our dynamic active learning algorithm with **MPSV-DY** as underlying uncertainty score outperforms previous active learning methods using **MPSV** and **SU** as uncertainty score significantly with lower average error rate. It shows that our active learning system is able to select the most informative subset of unlabeled sequences to query for labeling.

6 Conclusion and Future Work

In this paper, we have proposed two measurements of uncertainty in SVM^{struct} for selecting the most informative sequences to query from labeling from a pool of unlabeled sequences. One is the most-possible-constraint-violation method (**MPSV**) and another is simple uncertainty(**SU**) method. We compare our proposed methods with random selection on three real data set from named entity recognition task and subphrase generation task for queries. Our first experiment result on selecting sequences with same length shows that the most-possible-constraint-violation method (**MPSV**) and simple uncertainty(**SU**) outperform the random method significantly. Also **MPSV** outperforms **SU** by considering the underlying class distribution. We also propose a new active learning for sequence labeling using dynamic programming to select the best combination of sequences that maximizes the total uncertainty and restricts labeling effort, which is defined as the total number of elements of the selected sequences. Experiment result shows that it performs better than previous active learning system for sequence labeling. In the future, we will explore the possibility of considering representativeness of a sequence in a pool and selecting a sequence with both high uncertainty and good representativeness.

References

1. F. N. Aidan. Active learning selection strategies for information extraction.
2. C. J. C. Burges. A tutorial on support vector machines for pattern recognition. In *Knowledge Discovery and Data Mining*, page 2(2), 1998.
3. D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
4. I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *International Conference on Machine Learning*, pages 150–157, 1995.
5. G. D. Forney. The viterbi algorithm. In *Proceedings of the IEEE 61(3)*, pages 268–278, 1973.
6. M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
7. F. Jing, M. J. Li, H. J. Zhang, and B. Zhang. Entropy-based active learning with support vector machines for content-based image retrieval. In *IEEE International Conference on Multimedia and Expo*, pages Volume 1, Issue , 27–30 June 2004 Page(s): 85 – 88 Vol.1. Digital Object Identifier, 2004.
8. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
9. A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA, 2000.
10. A. K. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In J. W. Shavlik, editor, *Proceedings of 15th International Conference on Machine Learning*, pages 350–358, Madison, US, 1998. Morgan Kaufmann Publishers, San Francisco, US.

11. N. Nguyen and Y. S. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, pages 681–688, New York, NY, USA, 2007. ACM Press.
12. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
13. N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann, San Francisco, CA, 2001.
14. E. F. T. K. Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan, 2002.
15. D. Shen, J. Zhang, J. Su, G. D. Zhou, and C. L. Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*, pages 589–596, Barcelona, Spain, July 2004.
16. L. Stegeman. Part-of-speech tagging and chunk parsing of spoken dutch using support vector machines. 2006.
17. C. T. Symons, N. F. Samatova, R. Krishnamurthy, B. H. Park, T. Umar, D. Buttler, T. Critchlow, and D. Hysom. Multi-criterion active learning in conditional random fields. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 323–331, Washington, DC, USA, 2006. IEEE Computer Society.
18. S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In P. Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 999–1006, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.
19. I. Tsochantaris, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.
20. J. Weston and C. Watkins. Multi-class support vector machines. *Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK.*, 1998.