Pattern Recognition Letters 31 (2010) 1591–1597

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A composite kernel for named entity recognition

Sujan Kumar Saha *, Shashi Narayan, Sudeshna Sarkar, Pabitra Mitra

Computer Science and Engineering Department, Indian Institute of Technology Kharagpur, Kharagpur 721302, India

ARTICLE INFO

Article history:
Received 21 August 2009
Available online 8 May 2010
Communicated by T. Vasilakos

Keywords:
Named entity recognition
Support vector machine
Kernel methods
String kernel
Machine learning

ABSTRACT

In this paper, we propose a novel kernel function for support vector machines (SVM) that can be used for sequential labeling tasks like named entity recognition (NER). Machine learning methods like support vector machines, maximum entropy, hidden Markov model and conditional random fields are the most widely used methods for implementing NER systems. The features used in machine learning algorithms for NER are mostly string based features. The proposed kernel is based on calculating a novel distance function between the string based features. In tasks like NER, the similarity between the contexts as well as the semantic similarity between the words play an important role. The goal is to capture the context and semantic information in NER like tasks. The proposed distance function makes use of certain statistics primarily derived from the training data and hierarchical clustering information. The kernel function is applied to the Hindi and biomedical NER tasks and the results are quite promising.

1. Introduction

A named entity (NE) denotes a noun or noun phrase referring to a name belonging to a predefined category like person, location and organization. Named entity recognition (NER) is the task of identifying and categorizing the named entities from text. Named entities are often the pivotal as well as the most information-bearing elements of a text, and NER systems find application in a number of tasks like information extraction, text mining and machine translation. Due to its immense importance, a substantial amount of work has been carried out for NER system development in various languages and domains.

The use of support vector machines (SVM) is quite common in NER and other natural language processing (NLP) tasks. SVM (Vapnik, 1995) is a margin based approach where the similarity between the instances, which are composed of the data along with the corresponding feature values, is used to define the classifier. The computation of the similarity (or distance) between the instances plays a very important role in the performance of a SVM classifier.

In NER task the most commonly used features are the surrounding words, suffix and prefix information. One approach of using such string features in SVM classifier is to use binary feature vectors (Kudo and Matsumoto, 2001; Isozaki and Kazawa, 2002; Takeuchi and Collier, 2002), where a particular feature is converted into several binary values. For example, the feature 'previous word' is converted into N binary features where N is the total number of unique words in the lexicon. Another approach has been to define a kernel function that is directly applicable to the string features. Several types of 'string kernels' (Leslie et al., 2002; Lodhi et al., 2002; Tian et al., 2007) have been defined based on the fact that the similarity between two particular strings depends on the number of common sub-strings they have. String kernels have been used successfully in various tasks like text classification, protein classification, and entity and relation extraction.

But we feel that the binary representation and sub-string similarity based string kernels are not able to capture well the semantic similarity between the instances in context sensitive tagging tasks like NER. For example, the two words 'Prof.' and 'Chairman' have some similarity in the context of the NER task as both of these occur at the preceding position of the person named entities. Such similarity between the words cannot be captured by binary or sub-string kernel based approaches. In order to characterize these NER task specific similarity between the features we propose a class association kernel and a hierarchical word clustering based kernel. We then form a composite kernel by combining these two kernels.

For the class association kernel, the feature space is divided into a number of sub-groups where each feature group consists of a set of similar features. The individual features are then transformed into a $c+1$ dimensional vector where c is the number of named entity classes. These vectors are based on class association based statistics derived from the training data. The similarity between the vectors in a feature group is computed by making use of an appropriate distance function. This similarity is the sub-kernel value corresponding to the feature group. Finally, the sub-kernels

* Corresponding author. Tel.: +91 9732655684; fax: +91 3222 278985.
E-mail addresses: sujan.kr.saha@gmail.com (S.K. Saha), shashi.narayan@gmail.com (S. Narayan), shudeshna@gmail.com (S. Sarkar), pabitra@gmail.com (P. Mitra).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.05.004

are combined in a weighted fashion to obtain the final class association kernel.

In the second approach of computing the distance between the strings we use hierarchical clustering information. We have used the Brown clustering algorithm (Brown et al., 1992) to cluster the string feature values (e.g., words) based on their contextual similarity in a corpus.

The proposed kernel has been tested in NER task on two different domains. The first task we attempt is the Hindi NER task. The proposed kernel is compared with a binary feature based linear SVM classifier, substring similarity based string kernel and also with other statistical classifiers like maximum entropy (MaxEnt) and conditional random fields (CRF). We also test the proposed kernel in another domain, namely, the biomedical NER. In both the task the proposed kernel performs quite well.

2. Previous work

In this section, we present an overview of the research that have been carried out for developing NER systems in Hindi and biomedical domain. We also present a brief overview of the kernel based approaches in NER and other text processing tasks.

2.1. Hindi NER task

In the last few years a substantial amount of work has been carried out for developing NER systems in different languages and domains. For developing NER systems, machine learning (ML) based approaches have been mostly used. Several machine learning algorithms have been used for NER system development in various languages and domains. Hidden Markov model (HMM) (Collier et al., 2000; Shen et al., 2003; Zhou and Su, 2004; Ponomareva et al., 2007), MaxEnt (Borthwick, 1999; Lin et al., 2004; Kim and Yoon, 2007; Saha et al., 2008), CRF (Li and McCallum, 2004; Settles, 2004; Tsai et al., 2006; Leaman and Gonzalez, 2008), SVM (Kazama et al., 2002; Takeuchi and Collier, 2002; Lee et al., 2004), etc. are the most commonly used techniques.

Machine learning based methods are mostly used for Hindi NER system development too. Due to several language specific issues like, absence of capitalization, free word order, high ambiguity in Indian names and unavailability of sufficient resources, the Hindi NER task is quite difficult.

A pioneering work on Hindi NER is by Li and McCallum (2004) where they used CRF and feature induction. In their study the training corpus size was 340K words with 15,063 NEs belonging to three types, namely person, location and organization. They achieved a *f*-value of 71.5. Saha et al. (2008) used a training corpus containing 243K words to develop a Hindi NER system using MaxEnt classifier. They explored different NER features in Hindi language and studied their effectiveness. Gazetteer lists as well as identified context patterns were integrated in the system. Integrating all these approaches a *f*-value of 81.52 was achieved considering four NE classes (person, location, organization and date). In IJCNLP 2008 (International Joint Conference on Natural Language Processing, Hyderabad, India) a shared task[1] was organized on identification of NEs from texts in south and south-east Asian languages. Five languages were considered in the task which are Bengali, Hindi, Oriya, Telugu and Urdu. The task also included the identification of the nested NEs. The best result in the shared task was a *f*-value of 65.13 for Hindi where MaxEnt classifier and context rules were combined to prepare a hybrid system (Singh, 2008).

[1] More information on the shared task is available at: *http://ltrc.iiit.net/ner-ssea-08/index.cgi*.

2.2. Biomedical NER task

Biomedical NER task refers to the identification of biomedical named entities (like, protein, DNA, RNA) from (biomedical) text. Due to the presence of several difficulties (Shen et al., 2003), the performance of the biomedical NER systems is quite low compared to the general domain English NER systems. As in general domain, machine learning techniques are mostly used in biomedical NER. Several of the systems use a rule based postprocessing step, even though the core system is primarily built using machine learning algorithms.

In our experiments we have used the JNLPBA 2004 corpus (Kim et al., 2004). Here we mention a few systems developed using this corpus. A number of systems participated in JNLPBA 2004 shared task. Among these the highest accuracy was achieved by the system developed by Zhou and Su (2004) which achieved a *f*-value of 72.55. This system used HMM and SVM with some deep domain knowledge like in domain POS, name alias resolution, cascaded NE resolution, abbreviation detection, external name dictionaries. Without these domain knowledge the reported accuracy of the system was a *f*-value of 60.3. The second highest accuracy in the task was achieved by the maximum entropy Markov model (MEMM) based system developed by Finkel et al. (2004). This system used external resources (e.g., British National Corpus, large gazetteer lists, web), deeper syntactic features etc. to achieve a *f*-value of 70.06. Some other systems (Settles, 2004; Song et al., 2004) in the shared task that achieved good accuracy also used some amount of domain knowledge or external resources.

2.3. SVM and kernel in text processing tasks

SVM based classifiers have been used in various text processing tasks in the last few years. As most of the text processing tasks are required to use string features, several techniques have been adapted for handling the strings. Binary representation of the features is a common approach. Kudo and Matsumoto (2000, 2001) used SVM with binary feature representation in the English base phrase identification task. Takeuchi and Collier (2002) used SVM in the NER task with binary feature representation. They used the SVM classifier in the general domain NER task using MUC6 data and also in the molecular biology domain. For the NER task SVM was also used by Isozaki and Kazawa (2002). They proposed a few techniques like removal of unnecessary features to make the classifier efficient in terms of training time and performance.

The use of kernel functions applicable on string features is another popular approach for handling string features in SVM. A function that calculates the inner product between mapped instances in a feature space is a kernel function. A set of such functions have been proposed for handling string features, which are commonly named as 'string kernel'. String kernels calculate the similarity between the strings. One idea for getting the similarity between two strings is to find the amount of common substrings they contain – more substrings in common might refer to more similarity. A few kernels have been proposed based on this idea. Leslie et al. (2002) proposed the spectrum kernel and used it in the protein classification task. Leslie et al. (2004) proposed another string kernel for the protein classification task and named this as mismatch string kernel. Lodhi et al. (2002) proposed the string subsequence kernel which has been successfully used in the text classification task. Tian et al. (2007) proposed a light-weight string kernel based on matching subsequences with all possible lengths shared by two strings and used the kernel in the sequence data classification problem. The computation of such substring based kernels is complex, and some work has been done (e.g., Leslie and Kuang, 2004; Teo and Vishwanathan, 2006) in order to reduce the computation cost of the string kernels.