Chapter 1

Introduction

Hindi is the fourth most spoken language only after English, Chinese and Spanish. Bengali is the sixth language in the list with 211 million speakers. Four other Indian languages take place in the list of top 20 most spoken languages, these are, Punjabi, Telugu, Marathi and Tamil. In spite of such a large number of speakers in these languages, the usage of Indian languages in the web and in the computer world is very low. English is the primary medium for computer interaction still today and for this the majority of the Indian population are away from the advantage of the computer technology. To spread the advantage of the computer technology to a larger population, many more languages should be incorporated in the computer system. For boosting the use of Indian languages in the computer world, a large amount of natural language processing research is required so that a variety of language tools in Indian languages become available to the common users. As the named entity recognition systems have huge applications in such researches, the development of good named entity recognition system is a big step towards that.

1.1 Named Entity Recognition

The term 'named entity' (NE) is coined by Grishman and Sundheim (1995) in the sixth message understanding conference (MUC6). A named entity (NE) denotes a noun or noun phrase referring to a name belonging to a predefined category

like person, location and organization. The task of identifying and categorizing named entities from text is known as named entity recognition (NER). As example here we consider a sentence chosen from the CoNLL-2002 shared task (Sang, 2002): "Wolff, currently a journalist in Argentina, played with Del Bosque in the final years of the seventies in Real Madrid". The sentence contains a few names, which are, *Wolff* and *Del Bosque* - person name, *Argentina* - location name and *Real Madrid* - organization name. A NER system identifies these names with their category.

NEs 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. Although a lot of work was done in the early 90's to develop NER systems, it is still an active research area. The need for NER systems in various languages and domains, and the language and domain specific difficulties, keep the task alive and interesting.

Before moving into the development issues and techniques, we first want to mention the major applications of NER systems.

1.2 Applications of NER System

As the names are the pivotal element of text, NER system is used in a number of text processing tasks. The major applications of NER system are mentioned below.

Information extraction: Information extraction (IE) is the task of extracting structured information from unstructured natural language texts. NER is one of the major subtasks of IE. NEs should be identified for further extraction of terminology, relationship and other required information from a text.

Text mining: Apart from IE, NER plays an important role in various text mining tasks. As the NEs are the most information bearing content of text, these can help in a number of subtasks of text mining or intelligent text processing.

For example, the NEs help to understand the content of a document. Only identifying the NEs from a text, one can get an overall idea about it. Hence a NER system can help in the *document understanding* and *text summarization* tasks. Also a NER system can improve the performance of a *question answering* system. NER system has applications in the *topic tracking* and *topic detection* tasks too.

Machine translation: For building a machine translation system it is required to detect the NEs as a preprocessing task. Because, during the translation of a text from one language to another, the NEs are not translated, these are transliterated.

Intelligent web mining: NEs can play a major role in mining from the web. The result of a search engine can be improved by using the NEs occurring in the web pages. Also the NER systems help in *cross lingual information retrieval*.

Apart from these major applications, NER system has other usage too. For example, a NER system can help to improve the accuracy of a parts-of-speech tagging system by resolving the ambiguities between the names and other words, NER system can be used for *mining opinion* from a large blog data and *time varying entity analysis*.

1.3 NER Approaches

Now we summarize the approaches that have been used for developing NER systems (these are discussed in detail later in Chapter 2). The techniques have been used for NER system development can be classified broadly into two categories, namely, rule based and machine learning (ML) based.

The rule based approach for NER uses a set of linguistic rules to identify the names from text (e.g., Grishman, 1995; Jacobs and Rau, 1990; McDonald, 1996). The rules for NER are primarily defined using the context information of a target word, that is, the surrounding words decide whether the target word is a NE or not. Hence the identification of the rules requires deep knowledge about the language and domain. Identification of a set of rules which can recognize most

of the names from an arbitrary text requires huge time and deep study of the domain. Also the rules defined for a particular NER task (in a specific language and domain) can not be transferred to another language or domain. Although the earlier NER systems used this approach for NE identification, the rule based approach is not widely used in the past few years due to these limitations.

Machine learning (ML) based approaches have been mostly used for NER system development in the past few years (e.g., Borthwick, 1999; McCallum and Li, 2003; Nadeau, 2007; Olsson, 2008; Srihari et al., 2000; Takeuchi and Collier, 2002; Zhou and Su, 2004). The core module of a machine learning based system is a machine learning classifier which is trained using training data and a set of features. Several of machine learning classifiers have been used in the NER task. Hidden Markov model (HMM), maximum entropy (MaxEnt), conditional random fields (CRF), artificial neural network (ANN), support vector machines (SVM) etc. are the most widely used techniques. The primary requirement for preparing a machine learning classifier is a training data, where the NEs are pre-annotated. The performance of a classifier largely depends on the amount and quality of the training data.

Gazetteer list (a collection of names of a particular category) look up based name identification technique is also widely used for NER system development. Here a target word is searched in the gazetteer lists and occurrence of the word in a name list decides the name category. But as the preparation of a name list that contains all the possible names is impossible and a particular name may belong to more than one category (ambiguity), a good NER system can not be built using only the gazetteer look up based approach. This technique has been often used as an additional module to improve the performance of a rule based or machine learning based NER system.

1.3.1 NER Task in Indian Languages

In the last two decades, a lot of works has been carried out for NER system development in English and several other languages like Spanish, German, Chinese and Japanese. In the past few years several people have worked for NER systems in various Indian languages. Like English, in Indian languages also machine

learning based approaches are mostly used to develop NER systems.

A few of the representative works are, Li and McCallum (2003), Kumar and Bhattacharyya (2006), Ekbal and Bandyopadhyay (2008), Sobha and Ram (2007), Shishtla et al. (2008). In IJCNLP 2008 (International Joint Conference on Natural Language Processing, Hyderabad, India), a shared task was organized on NER for south and south-east Asian languages (Singh, 2008). The five languages considered in the task were Bengali, Hindi, Oriya, Telugu and Urdu. A number of systems participated in the task. More detailed report on NER task in general and in Indian language is presented later in Chapter 2.

1.4 NER Difficulties: General and Indian Language Specific

In general the major difficulties of the NER task are, constantly created new names, name variants, name abbreviations and above all, the ambiguity.

Ambiguity: Ambiguity is one of the major challenges in NER system development. A particular word may belong to different classes in its different occurrences. Due to the ambiguity the NE classification is not straightforward because NE categories are not so clear. Ambiguity occurs between person and location (e.g., Jordan, Bangalore etc. are person as well as location entities), location and organization (e.g., in "India has own the cricket cup" - 'India' refers to a team i.e., an organization), person and organization (e.g., Ford) and similarly between other name classes. Even in some languages high ambiguity is observed between the names and the common words.

Embedded NE: A NE belonging to a particular category may combine with other words to form a NE of another category. For example, person to location (Subhas Chandra Bose *road*, Vivekanada *Vihar*), person to organization (Tata *Steel*, Vidyasagar *University*), location to organization (Delhi *University*, *IIT* Kharagpur). Such embedded or nested NEs are difficult to recognize properly.

New names: It is not possible to create a name dictionary that contains all

the names. Names are being created constantly and the amount of total names available in the world is increasing day by day. The NER systems sometimes find difficulty to identify the unknown names.

Abbreviation: Names are often abbreviated. For example, 'Microsoft Research India' is abbreviated as MSRI, 'IBM India Research Lab' is often written as as IRL. These abbreviations increase the difficulty. Sometimes the person names are also abbreviated; for example, 'Kevin Peterson' is abbreviated as KP, 'Sachin Ramesh Tendulkar' might occur in text as SRT, 'Mahendra Singh Dhoni' is abbreviated as MSD, again Ministry of Social Development is also abbreviated as MSD and there are other organizations like MSD Soft and MSD Pharma. These abbreviations are very difficult to recognize.

Name variants: A particular name might occur with several variations in a text. For example, the name 'Rabindranath Tegore' might also occur as, Rabithakur, Rabi-babu, Tegore-ji ('ji' is an honorary term used often in Hindi) and so on. These variations add difficulty in the name recognition task.

In addition, the NER task in Indian language have other challenges too. These are mentioned below.

Resource scarcity: The major challenge in developing NER system in Hindi we face is the resource scarcity. When we started developing a machine learning based NER system in Hindi, we could not find any publicly available NE annotated corpus. Not only the NE annotated corpus, other relevant resources are also not available in Indian languages. We have observed that name dictionaries or gazetteer lists have been used often in NER systems in order to improve the performance. But there is no such publicly available gazetteer list in Indian languages.

Availability of preprocessing tools: The tools like stemmer or root extractor, morphological analyzer, parts-of-speech (POS) tagger, parser etc. are often required during preprocessing in the NER task. Some of these tools with acceptable accuracy are not publicly available in Indian languages.

High ambiguity: In Indian languages the ambiguity is quite high. The

person names are too diverse, many common words are used as names. *AkAsha*[1] (sky), *Alu* (potato), *sambhaba* (possible), *pAkhi* (bird), *nIla* (blue), *kara* (do), *sandhyA* (evening) etc. are the examples of Indian names with their meaning in parenthesis that give an idea of ambiguity in Indian names. Ambiguity between different name classes is also present.

Free word order: The word order is quite free in Indian languages. Due to the free word order, it is difficult to define and rely on context rules for identifying the NEs.

Unavailability of capitalization: Unlike English, there is no capitalization of the characters in Indian language texts. In English, the names are generally capitalized which helps a lot in finding the NEs. Also the capitalization helps to resolve the ambiguity between the names and common words. Unavailability of capitalization makes the Indian language NER task more challenging.

1.5 Motivation and Objectives

The need for NER system in Indian languages motivated us to work on Hindi and Bengali NER. The resources required for a NER task are, (1) NE annotated corpus and (2) additional resources like gazetteer lists (list of names of a particular category) and context patterns. To develop a NER system for a new language we need to create a NE annotated corpus. We want to find out how a NE annotated corpus can be created with less manual effort. We also want to leverage the resources already available and convert these to resources required for the Indian language NER systems.

Apart from corpus, features also have an important role in machine learning based NER systems. Analysis of the corpus to study the NEs and their occurrences in the text need to be done to select a suitable feature set for Indian languages.

We also want to explore the popular machine learning classifiers used for

[1]In this thesis all the Indian language words are written in italics and using the Itrans transliteration, the details of Itrans can be found at www.aczoom.com/itrans/

NER and other sequential labeling tasks. We decide to use three different classifiers, namely, maximum entropy, conditional random fields and support vector machines to evaluate their performance in the context of the Hindi NER task and identify their limitations.

Manual annotation of a sufficiently large training corpus is costly and time consuming. On the other hand, several shortcomings often arise when a NER classifier is built with limited resource. We wish to investigate the approaches to overcome these shortcomings, so that an improved system can be built in such resource poor scenarios deriving the maximum benefit from the available training data.

We have observed that in machine learning classifiers overfitting can take place when a large number of features are used with a limited training data. Reduction of the existing feature set is an effective way to achieve better performance of the classifier by reducing overfitting. We want to explore different feature reduction techniques in the context of the NER task.

The performance of a SVM classifier depends on the kernel function. We want to work on a NER specific SVM kernel. In this kernel we like to capture the NER task specific semantic similarity existing between the feature values like words.

Semi-supervised learning techniques make use of a large unannotated corpus to supplement a small sized annotated corpus. We want to investigate a semi-supervised learning framework appropriate for the NER task.

Our other objective is to build a NER system in Bengali. Since Bengali is quite similar to Hindi, we wish to make use of transfer based methods to quickly develop a Bengali NER system after having developed the resources for Hindi NER.

1.6 Contributions

We have identified some techniques for the development of accurate NER systems under a limited resource condition. We have applied these techniques to the Hindi NER task. We have also explored the applicability of few of these techniques in the biomedical NER task. Apart from this we have also worked on the development of NER system in Bengali. A brief report on the studies (which we shall elaborate in the forthcoming chapters) is presented below.

1. Hindi NER Corpus

We have created a Hindi NE annotated corpus containing about 225K words. The corpus is collected from the popular Hindi newspaper "Dainik Jagaran" and manually annotated. In this corpus we have considered three NE classes, namely, *person*, *location* and *organization*. The corpus contains about 6000 person, 5000 location and 3000 organization entities. We have used this corpus for our development and have made it available for use of other researchers.

2. Study on Indian Language NER Features

Since a model is built in terms of features, the effectiveness of a classifier crucially depends on what features are used. We have studied a Hindi NER corpus to understand which features are relevant for the NER task. For instance, we have observed that a large number of names in India share a common suffix or prefix; some specific words occurring at the surrounding positions of a target word help to predict the category; a postposition after a noun might give an idea about its category; and there is a tendency of using the person names with some qualifier words to make them location or organization entity. Based on our study we have selected an initial set of candidate features. We have performed extensive experiments considering these features individually and in combination in order to get a more restricted set. We have used three different classifiers, namely, MaxEnt, CRF and SVM in our experiments. The baseline Hindi NER system is built using the annotated corpus and the feature set.

3. Preparation of Gazetteer Lists and Context Patterns

Gazetteer lists are important as well as widely used resources in the NER task. We did not have any appropriate gazetteer list in Hindi or Bengali. But we found some lists of person, location and organization names from certain websites in English. To use these English lists in the Hindi NER task, we have proposed a normalized approach by defining a simplified phonetic alphabet. A translator is developed to convert each English name to a normalized form in this alphabet. We have also developed a translator to convert a Hindi word to this normalized form. When we need to check whether a Hindi word is in the name list, we check the translated word in the normalized name list. The use of the gazetteer lists enabled us to achieve better accuracy in the Hindi NER system.

Context patterns can also help to identify the NEs. Manual identification of a set of context patterns is time consuming and requires linguistic expertise. We have automatically extracted a set of frequently occurring context patterns from the annotated corpus. These extracted patterns are then evaluated using a large raw corpus. The high precision patterns are selected and used in the NER system. When the extracted context patterns are integrated in the baseline Hindi NER system, the performance is improved.

4. Feature Reduction Techniques

During the development of a NER classifier using the annotated data and the features, we have observed that the classifier suffers from overfitting because a high-dimensional feature set is applied on insufficient training data. To improve the performance of the classifier, the feature set should be reduced. For feature reduction in NER we have proposed a few feature selection and feature clustering based techniques. The proposed class association metric based selection technique selects the features which have important role in the recognition task. During feature clustering the features are transformed into a vector representation which uses certain statistics derived from the annotated data. The use of the reduced feature set causes performance improvement of the classifier. We have made a comparative study of other feature reduction techniques in the literature in the context of the Hindi NER task. We have also applied the proposed feature

reduction techniques to the biomedical NER task, where also we have achieved significant performance improvement.

5. NER Specific SVM Kernel

In a linear SVM classifier, the similarity between two feature values is computed by simply inspecting whether these are same or different. For example, the similarity between two feature values of the *previous word* feature, 'professor' and 'Dr.' is zero. But in the context of the NER task these words have some similarity as both of these occur at the previous position of the person NEs. In order to capture such similarity we have proposed two different types of distance functions. The first one defines a feature vector corresponding to each feature value based on its occurrence in the proximity of named entities. The second distance function makes use of a hierarchical clustering algorithm. Both these functions are combined with a suitable weight factor to obtain a composite kernel. In our Hindi NER experiments the proposed kernel performs better than the baseline SVM classifier where a linear kernel is used with binary feature representation. The proposed kernel is also tested in the biomedical NER task. The results look quite promising in the biomedical NER task too.

6. Semi-supervised Learning

Scarcity of annotated data is a big challenge in building high performance NER systems in resource poor languages. The creation of a large NE annotated corpus is costly and time consuming, but raw corpus is often easily available. Hence we have investigated whether we can improve the performance of our Hindi NER system by adopting semi-supervised learning (SSL) that uses a limited annotated corpus along with a large raw corpus. A NER task specific annotation confidence computation technique is proposed for the purpose. A bootstrapping technique is used along with the confidence measure. Apart from bootstrapping, we have proposed two more approaches to SSL. One is based on creation of an additional classifier using a modified confidence computation algorithm and a raw corpus (no training corpus is used in this additional classifier) and then combining this with the supervised classifier. Another approach is the prior modulation of the

classifier where the confidence weights are used as 'prior'. These semi-supervised classifiers perform better than the supervised classifier.

7. Adaptation of Hindi NER System to Bengali

After having developed a Hindi NER system we wanted to quickly develop a Bengali NER system. To develop a Bengali NER system, we could have gone through the same procedure like Hindi. But we wanted to investigate whether we can take the advantage of the existing NER system in Hindi and can prepare a Bengali NER system without using NE resources in Bengali and without investing much time. The baseline Bengali NER system is prepared using a gazetteer look-up approach and context patterns. Gazetteer lists for Bengali are prepared using the Hindi NER system, a large Hindi raw corpus and transliteration. A few additional modules are used to improve the gazetteer look-up based identification. We also worked on transfer of the existing Hindi classifier to Bengali through feature set conversion, where a Hindi NER feature template is converted into Bengali. The classifiers prepared using the aforementioned approaches achieve high precision but suffers from poor recall. To improve the recall we first used the bootstrapping technique where recall is improved but the precision is degraded. Then we have used active learning where the uncertain samples are selected and queried by the system to a human annotator. With these approaches we were able to develop a Bengali NER system with moderate performance.

1.7 Outline of the Thesis

The thesis is organized into eight chapters.

Chapter 2 presents the baseline named entity recognition system. It first presents an overview of the approaches that have been used for developing NER systems in various languages and domains. It also presents a number of works that have been carried out for NER system development in Indian languages and in the biomedical domain. Then it discusses the details of the baseline

NER systems for Hindi and biomedical domain, including NER corpora, features, learning techniques, NER system evaluation metric and the baseline results.

Chapter 3 presents the techniques that are used for the preparation of additional resources. It discusses the proposed approach for preparing gazetteer lists in Hindi with the help of English resources. Furthermore it describes the context pattern extraction module that we have used in the Hindi NER task.

Chapter 4 consists of few proposed feature reduction techniques and their performance in the Hindi and Biomedical NER tasks. Furthermore it presents a comparison of the proposed techniques with other related techniques in the literature.

Chapter 5 introduces a family of SVM kernel functions, which are able to capture the NER task specific semantic similarity between the feature values. It discusses the individual similarity computation techniques as well as their performance in the Hindi and Biomedical NER tasks.

Chapter 6 presents the semi-supervised learning techniques that we have used for the NER task in an environment with a small annotated corpus and a large unannotated corpus. It discusses a NER task specific confidence measure, the semi-supervised learning techniques that we have proposed for the NER task and the comparison of the proposed techniques with the standard bootstrapping technique.

Chapter 7 investigates the techniques for adapting a Hindi NER system to a Bengali NER system where almost no Bengali NER resources are used.

Chapter 8 concludes the thesis by summarizing the basic findings also by suggesting future direction of research in this area.

There is also an appendix that presents a list of publications from this work.