****

**Named Entity Recognition for Nepali Text using Support Vector Machine**

**A Dissertation Proposal**

**Submitted by:**

Surya Bahadur Bam

Roll No. 14

Batch (2009 - 2011)

**Submitted to:**

Central Department of Computer Science and Information Technology

Tribhuvan University

Kirtipur, Nepal

**Under the supervision of**

Prof. Dr. Shashidhar Ram Joshi (IOE, TU)

&

**Co-supervision of**

Mr. Sarbin Sayami (CDCSIT, TU)

1. **Introduction**

Named Entity (NE) is the structured information referring to predefined proper names, like persons, locations, and organizations etc. NE task is to identify all named locations, named persons, named organizations, date, times, monetary amounts, percentages etc. in text. Named Entity Recognition (NER) aims to classify each word of a document into predefined target named entity classes and is now-a-days considered to be fundamental for many Natural Language Processing (NLP) tasks such as information retrieval, machine translation, information extraction, question answering systems [1]. Though support vector machine (SVM) [15] technique has been widely applied to NER in several well studied languages, the use of SVM technique to Nepali Languages (NLs) is very new. The system makes use of the different contextual information of the words along with the variety of features that are helpful in predicting the four different Named Entities (NE) classes [2, 13], such as Person name, Location name, Organization nameand Miscellaneous name*.* The Miscellaneous name include date, times, monetary amounts, percentages, designation etc.

Proper identification and classification of NEs are very crucial and pose a very big challenge to the NLP researchers. The level of ambiguity to NER makes it difficult to attain human performance. Named entity identification is difficult and challenging for Indo-Aryan language like Nepali due to lack of resources.

In recent years, automatic NER systems have become a popular research area in which a considerable number of studies have been addressed on developing these systems. These can be classified into three main classes [4, 5], namely rule-based NER, machine learning-based NER and hybrid NER.

Now- a- days, Machine-Learning (ML) approaches are popularly used in NER because these are easily trainable, adaptable to different domains and languages as well as their maintenance are also less expensive [16]. On the other hand, rule-based approaches lack the ability of coping with the problems of robustness and portability. Each new source of text requires signiﬁcant tweaking of rules to maintain optimal performance and the maintenance costs could be quite high. Some of the well-known machine learning approaches used in NER are Hidden Markov Model (HMM) [7], Maximum Entropy (ME) (New York University’s MENE) in [4], Decision Tree [12] and CRF [5].

1. **Problem Definition**

The Named Entity Recognition is the problem which asks for the classification of each word of a document into predefined target Named Entity classes. In this research work, problem of Nepali named entity recognition is addressed. The recognition task is carried out with supervised machine learning using Support Vector machine (SVM) [9]. Feature selection plays a crucial role in the Support Vector Machine (SVM) framework. Experiments should be carried out in order to find out the most suitable feature for NER in Nepali.

Given a set of classes, all strings that are labels of instances of these classes within a text fragment are found. For example,

राम गोरखामा जन्मेको थियो । [राम, PER] [गोरखा, LOC] मा जन्मेको थियो ।

The main feature for the NER task should be identifying based on the different possible combination of available word and tag set. The sub problems in the domain of Nepali Named Entity Recognition such as, feature selection, part of speech (POS) information, word suffix, word prefix, context word feature, digit features Gazetteer lists etc. have huge impact on named entity recognition procedure. These sub problems are also addressed with the most suitable solutions in the literature for this type research work. In general, even though there has been lots of researches done in named entity recognition in other languages, but still there is no such work done for Nepali language.

1. **Challenge of NER Nepali Language**

Named Entity (NE) identification in Nepali languages is difficult and challenging as:

1. Unlike English and most of the European languages, Nepali lacks capitalization information, which plays a very important role in identifying NEs.
2. Nepali person names are more diverse compared to the other languages and a lot of these words can be found in the dictionary with some other specific meanings.
3. Nepali is a highly inflectional language providing one of the richest and most challenging sets of linguistic and statistical features resulting in long and complex word forms.
4. Nepali is a relatively free order language.
5. Nepali is a resource poor language annotated corpora, name dictionaries; good morphological analyzers, POS taggers etc. are not yet available in the required measure.
6. Although Nepali language have a very old and rich literary history, technological development are of recent origin.
7. Web sources for name lists are available in English, but such lists are not available in Nepali forcing the use of transliteration for creating, such lists.
8. **Objectives**

The objective of this study is to implement and analyze the algorithms for Nepali Named Entity Recognition viz. Support Vector Machine (SVM). And, hence to build a model, that will result Nepali Named Entity for Nepali text. The main objective is given below,

1. To analyze the SVM based named entity recognition system for Nepali language.
2. **Literature Review**

There are lots of researches which have been done in the field of Named Entity Recognition, but there is no any work in Nepali NER.

In support vector machine [6] method, data consisting of two categories is classified by dividing space with a hyperplane. It is shown that when the margin between example that belong to one category and example that belong to other category in the training data is larger, the probability of incorrectly choosing categories in test data is small. Hence the maximizing the margin becomes the optimization problem. The SVM [6] is basically binary classifier but it can be extended to multiclass classification using one of the methods: one versus rest, pair wise.

Support Vector Machines (SVMs) based NER system [15] was proposed by Yamada et al. [14] for Japanese. His system is an extension of Kudo’s chunking system [9] that gave the best performance at CoNLL-2000 shared tasks. The other SVM-based NER systems can be found in [1, 3].

The author of [4,11] had shown that Conditional Random Fields (CRFs) are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes. A conditional random field (CRF) is a type of discriminative probabilistic model used for the labeling sequential data such as natural language text. Conditionally trained CRFs can easily include large number of arbitrary non independent features. The expressive power of models increased by adding new features that are conjunctions to the original features. When applying CRFs to the named entity recognition problem an observation sequence is the token sequence of a sentence or document of text and state sequence is its corresponding label sequence.

The author of [8] had shown that the maximum entropy [4, 5] framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived from training data, expressing some relationship between features and outcome. The probability distribution that satisfies the above property is the one with the highest entropy. It is unique, agrees with the maximum-likelihood distribution, and has the exponential form

|  |  |
| --- | --- |
|  | (1) |

Where  refers to the outcome, h the history (or context), and Z(h) is a normalization function. In addition, each feature function fj (h, o) is a binary function.

The author of [7] had shown that Name recognition may be viewed as a classification problem, where every word is either part of some name or not part of any name. In recent years, hidden Markov models (HMM’s) [16] have enjoyed great success in other textual classification problems most notably part-of-speech tagging. Given this success, and given the locality of phenomena which indicate names in text, such as titles like “Mr.” preceding a person name, they [16] have chosen to develop a variant of an HMM for the name recognition task. By definition of the task, only a single label can be assigned to a word in context. Therefore, our model will assign to every word either one of the desired classes or the label NOT-A-NAME to represent “none of the desired classes”.

1. **Research Methodology**

The methods used in the main module of this research work are described in this chapter. Sections are organized as follows. Data collection for the named entity recognition is presented in section 6.1. Implementation model for Nepali NER is presented in section 6.2 Preprocessing steps for the Named Entity recognition is described in section 6.3. Section 6.4 contains the named entity feature.

**6.1 Data Collection**

The vocabulary contains the Nepali news texts had been collected during 2009-2011 from daily news papers as well as the weekly, half monthly and monthly magazines of different domains such as national news, international news, sports, health, economics etc. NE tagged corpus will be created manually.

* 1. **Proposed Implementation Model for Nepali NER**

As an implementation model, we will be implementing the Support Vector Machine algorithms so as to extract the Name entities from the given input set of data. The algorithms will be simulated for various size of corpus. The implementation model is given below:

Named Entity Tagged Corpus

(Manually tagged)

Feature Extraction

(Construction of Feature Vector)

SVM training

(Learn SVM classifier)

Tagged NEs

Figure 6.1 Implementation Model for Nepali NER

The SVM light [12] will be used to learn the model from these training vectors.

**6.3 Preprocessing**

Using a supervised machine learning technique relies on the existence of annotated training data. Such data is usually created manually by humans or experts in the relevant field. The training data needs to be put in a format that is suitable to the solution of choice. New data to be classified also requires the same formatting. Depending on the needs of the solution, the textual data will be tokenized, normalized, scaled, and mapped to numeric classes, prior to being fed to a feature extraction module. To reduce the training time with large training data, some techniques such as chunking or instance pruning (filtering) may need to be applied. There is no NE tagged Nepali corpus, so we have to create manually tagged corpus for this research work.

* 1. **Feature Extraction**

In this phase, training and new data is processed order to extract the descriptive information about it. Feature selection plays a crucial role in the Support Vector Machine (SVM) [3] framework. Experiments have been carried out in order to find out the most suitable features for NER in Nepali languages. The main features for the NER task have been identified based on the different possible combination of available word and tag context. Relevant features for NER will be extracted. Following are the details of the set of features that will be apply to the NER task:

1. Context word feature: Preceding and following words of a particular word can be used as the features. This is based on the observation that the surrounding words are very effective in the identiﬁcation of NEs.
2. Named Entity Information: The NE tag(s) of the previous word(s) will also be used as feature. This is the only dynamic feature in the experiment.
3. Digit features: Several binary valued digit features have been deﬁned depending upon the presence and/or the number of digits in a token.
4. Part of Speech (POS) Information: The POS of the current and/or the surrounding word(s) can be used as features.
5. Gazetteer Lists: Various gazetteer lists will be used.
6. Person name: This list contains the name of persons. The feature PersonName is set to +1 for the current word. is set to +1 for the current word.
7. Location name: This list contains the location names and the feature LocationName is set to +1 for the current word.
8. Organization name: This list contains the organization names and the feature OrgnizationName is set to +1 for the current word.
9. Month name: This list contains the name of all twelve different months of both English and Nepali calendars. The feature MonthName is set to +1 for the current word.
10. Day name: This list contains the name of all seven different days of Nepali calendars. The feature DayName is set to +1 for the current word.

**6.5 Machine Learning and Classification**

After the feature extraction phase process of training and testing begins. In the training phase recognition system learns patterns of different classes from input feature vectors. The learning is done in supervised manner. Recognition system is then tested against testing feature vectors and accuracy and efficiency of the system is calculated. In this research, recognition is carried out using Support vector machine. Section 6.5.1 describes the SVM algorithm.

**6.5.1 Two class (Binary) SVM**

The author in [3] described that Support Vector Machines (SVMs), first introduced by Vapnik [7], and is relatively new machine learning approaches for solving two-class pattern recognition problems. SVMs are well-known for their good generalization performance, and have been applied to many pattern recognition problems. In the field of natural language processing, SVMs are applied to text categorization, and are reported to have achieved high accuracy without falling into over fitting even though with a large number of words taken as the features [6]. Suppose, we have a set of training data for a two-class problem: { (x1,y1),………,(xN, yN)} ,where **xi**∈RD is a feature vector of the ith sample in the training data and y ∈{+1 ,- 1 } is the class to which x belongs. In their basic form, a SVM learns a linear hyperplane that separates the set of positive examples from the set of negative examples with maximal margin(the margin is deﬁned as the distance of the hyperplane to the nearest of the positive and negative examples). In basic SVMs framework, we try to separate the positive and negative examples by hyperplane written as: (**w** .**x**)+b =0 **w** ∈**Rn** ,b∈ **R** .

SVMs find the optimal hyperplane which separates the training data into two classes precisely. The linear separator is deﬁned by two elements: a weight vector **w** (with one component for each feature), and a bias b which stands for the distance of the hyperplane to the origin. The classiﬁcation rule of a SVM is,

|  |  |
| --- | --- |
|  | (2) |
|  | (3) |

Where, x isthe example to be classified.

|  |  |
| --- | --- |
|  |  |
|  |  |

****

Figure 6.2: Two class SVM with support vectors and supporting hyperplane.

If data are linearly separable then there exist a d-dimensional vector **w** and a scalar *b* such that

|  |  |
| --- | --- |
|  | (4) |

And

|  |  |
| --- | --- |
|  | (5) |

In compact form we may combine these two equations in

|  |  |
| --- | --- |
|  | (6) |

|  |  |
| --- | --- |
| Or | (7) |

Here (w, b) define the hyper plane that separates data in two class. The equation of the hyperplane is

|  |  |
| --- | --- |
|  | (8) |

Where *w*is normal to the plane, *b* is the minimum distance from the origin to the plane. In order to make each decision surface (*w, b*) unique, we normalize the perpendicular distance from the origin to the separating hyperplane by dividing it by |w| giving the distance as**.**

As depicted in Figure 2, the perpendicular distance from the origin to hyperplane H1:

|  |  |
| --- | --- |
|  | (9) |

And the perpendicular distance from the origin to hyperplane H2:

|  |  |
| --- | --- |
|  | (10) |

The support vectors are defined as the training points on *H*1 and *H*2. Removing any points not on those two planes would not change the classification result, but removing the support vectors will do so. The margin, the distance between the two hyperplane *H*1 and *H*2 is . The margin determines the capacity of the learning machine which in turn determines the bound of the actual risk the expected test error. The wider the margin the smaller is *h*, the VC-dimension of the classifier. Therefore our goal is to maximize margin or equivalently minimize the.

Therefore the optimization problem can be formulated as follows

Minimize f= (11)

Subject to constraints (13)

This problem can be solved by using standard Quadratic programming technique [10].

The above SVM formulations require linear separation. The real life application data are not always linearly separable. To deal with nonlinear separation, the same formulation and techniques as for the linear case are still used. We only transform the input data into another space (usually of a much higher dimension) so that, a linear decision boundary can separate positive and negative examples in the transformed space (feature space) and the original data space is called the input space [10].

**6.5.2 Multi Class SVM**

The SVM described in the section 6.5.1 is used for binary classification and which classify data in binary class. But in the case of NER we have to use four classes, so multiclass SVM will be used. Since SVM are binary classifier so binarization of problem must be performed before apply them to NER. A SVM is trained for each NE tag in order to distinguish this class and the rest.

This can be explained with an example

[हरि, PER] [रेडियो नेपाल, ORG]को [पोखरा, LOC]स्थित स्टेशनमा काम गर्छ।

PER vs Rest

LOC vs Rest

ORG vs Rest

Class LOC Class PER

Class ORG

Figure 6.3: One Vs rest classification approaches for NER.

**6.5.3 Performance Metrics**

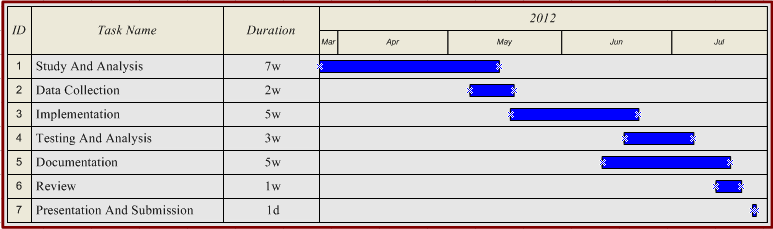
We use the following measures to evaluate the accuracy of the proposed method or model [3]. The measures taken are: precision (P), recall (R) and F-score (F).

**Precision:** The number of correctly retrieved NEs by the system divided by the number NEs retrieved by the system. Mathematically,

**Recall:** The number of NEs retrieved by the system divided by the number of NEs present in the test set. Mathematically,

**F-Score:** Harmonic mean of precision and recall. Mathematically,

1. **Working Schedule**



**References**

1. Asif Ekbal and Sivaji Bandyopadhyay *Bengali Named Entity Recognition using Support Vector Machine Proceedings of the IJCNLP08 Workshop on NER for South and South East Asian Languages,pp. 51–58, Hyderabad,India,January(2008).*
2. Asif Ekbal and Sivaji Bandyopadhyay *Named Entity Recognition Using Appropriate Unlabeled Data, Post-processing and Voting Informatica 34,pp. 55–76 (2010).*
3. Asif Ekbal and Sivaji Bandyopadhyay *Named Entity Recognition using Support Vector Machine: A Language Independent Approach International Journal of Electrical and Electronics Engineering 4:2 (2010).*
4. Borthwick, A.: *Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University (1999).*
5. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: *NYU: Description of the MENE Named Entity System as Used in MUC-7. In: MUC-7, Fairfax (1998).*
6. Corina Cortes Vladimir Vapnik *Support-Vector Networks Machine Learning, 20, pp.273-297 AT&T Bell Labs., Holmdel, NJ 07733, USA (1995 ).*
7. D. M. Bikel, R. L. Schwartz, and R. M. Weischedel, *“An Algorithm that Learns What’s in a Name,” Machine Learning, vol. 34, no. 1-3, pp. 211–231, (1999).*
8. Hai Leong Chieu Hwee Tou Ng *Named entity recognition: a maximum entropy approach using global information (2002).*
9. Kudo, T., Matsumoto, Y.: *Chunking with Support Vector Machines. In: Proceedings of NAACL pp. 192–199 (2001).*
10. Nello Cristianini and John Shawe Taylor*,* An Introduction to Support Vector Machines and Other Kernel- based Learning Methods, Cambridge University Press pp. 126(2002).
11. N.V Pabitra Mitra S.K. Ghosh: *Conditional Random Field Based Named Entity Recognition in Geological Text Sobhana ©2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 3.*
12. Sekine, S.: *Description of the Japanese NE System used for MET-2. In: MUC-7, Fairfax, Virginia (1998).*
13. T*.* Joachims, *“Making large-scale support vector machine learning practical,” pp. 169–184, (1999).*
14. Yamada, H., Kudo, T., Matsumoto, Y.: *Japanese Named Entity Extraction using Support Vector Machine. In Transactions of IPSJ 43, pp.44–53 (2001).*
15. Y.C. Wu, T.K. Fan, Y.L., Yen, S.: *Extracting Named Entities using Support Vector Machines. In: Springer- Verlag (2006).*
16. Zhou, G., Su, J.: *Named Entity Recognition using an HMM-based Chunk Tagger. In: Proceedings of ACL, Philadelphia pp.473–480 (2002).*