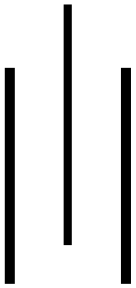
Tribhuvan University
Institute of Science and Technology

Named Entity Recognition for Nepali Text using Support Vector Machine



Dissertation
Submitted to

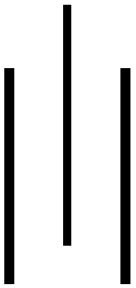
Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information Technology

By
Surya Bahadur Bam
05 July, 2013

Tribhuvan University
Institute of Science and Technology

Named Entity Recognition for Nepali Text using Support Vector Machine



Dissertation
Submitted to

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information Technology

By
Surya Bahadur Bam
05 July, 2013

Supervisor
Prof. Dr. Shashidhar Ram Joshi

Co- supervisor
Asst. Prof. Sarbin Sayami

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science & Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...
Surya Bahadur Bam
05 July, 2013

Supervisor's Recommendation

I hereby recommend that this dissertation prepared under my supervision by **Surya Bahadur Bam** entitled "**Named Entity Recognition for Nepali Text using Support Vector Machine**" in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Information Technology be processed for the evaluation.

……
Supervisor
Prof. Dr. Shashidhar Ram Joshi
Department of Electronics & Computer Engineering, Institute of Engineering, Pulchowk, Kathmandu, Nepal
05 July, 2013

……...
Co- supervisor
Asst.Prof. Sarbin Sayami
Central Department of Computer Science & Information Technology, Kirtipur Tribhuvan University, Kathmandu, Nepal
05 July, 2013

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science & Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Masters Degree in Computer Science and Information Technology.

Evaluation Committee

... ...
Prof. Dr. Tanka Nath Dhamala
Central Department of Computer Science & Information Technology,
Tribhuvan University, Kathmandu, Nepal
(Head of Department)

....
Prof. Dr. Shashidhar Ram Joshi
Department of Electronics & Computer Engineering, Institute of Engineering,
Pulchok, Kathmandu, Nepal
(Supervisor)

... ...
Assoc. Prof. Bal Krishna Bal
Kathmandu University
(External Examinar)

....
Mr. Bishnu Gautam
CDCSIT, Tribhuvan University
(Internal Examinar)

Date: ... ...

ACKNOWLEDGEMENT

With deep sense of gratefulness I express my genuine thanks to my respected supervisor **Prof. Dr. Shashidhar Ram Joshi**, Electronics & Computer Engineering Department Pulchowk, (Kathmandu, Nepal) for his valuable guidance in carrying out this work under his effective supervision and enlightenment.

I want to express my deep thanks to my honored co-supervisor **Asst. Prof. Sarbin Sayami**, Central Department of Computer Science and Information Technology (Kathmandu, Nepal) for his inspiration, trust, the insightful discussion, thoughtful guidance, critical comments, and correction of the thesis.

I would like to thank my respected promoter **Prof. Dr. Tank Nath Dhamala**, Head of Central Department of Computer Science & Information Technology, TU (Kathmandu, Nepal.

I would like to express my gratitude to respected teachers **Prof. Dr. Subarna Shakya, Prof. Sudarshan Karanjeet, Asst. Prof. Min Bahadur Khati, Asst. Prof. Nawraj Poudel, Asst. Prof. Dhiraj Pandey, Asst. Prof. Lalita Sthapit, Mr.Bishnu Gautam, Mr. Jagdish Bhatt, Mr. Arjun Singh Saud, Mr. Bikash Balami** and others staffs of CDCSIT for their full cooperation and help.

I cannot remain without admiring the efforts put by my friend **Mr. Tej Bahadur Shahi, Mr. Ashok Kumar Pant** for their exceptional participation on this work.

Finally, I thank my family for their love, support and encouragement.

ABSTRACT

Named Entity Recognition aims to identify and to classify rigid designators in text such as proper names, biological species, and temporal expressions into some predefined categories. It resolves who, where and how much problems in information extraction and leads to the resolution of the what and how problems in further processing. There has been growing interest in this field of research since the early 1990s. Named Entity Recognition have vital role in different field of natural language processing such as Machine Translation, Information Extraction, Question Answering system and various other fields. In this thesis, Named Entity Recognition for Nepali Text, based on the support vector machine is present which is one of the machine is learning approaches and domain independent work.

A set of features are extracted from training data set. Accuracy and efficiency of SVM classifier is analyzed in three different size of training data set. Recognition systems are tested with ten datasets for Nepali text. The strength of this work is the efficient feature extraction and the comprehensive recognition techniques. The support vector machine based named entity recognition is limited to use a certain set of features and it use a small dictionary which affects its performance.

The learning performance of recognition system is observed and found that it can learn well from the small set of training data and increases the rate of learning on the increment of training size.

Keywords:

Named Entity, Named Entity Recognition, Support Vector Machine, Classification, Feature Extraction.

Table of Contents

Lists of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ACE	Automatic Content Extraction
CRF	Conditional Random Field
CoNLL	Conference on Computational Natural Language Learning
CLR	Consortium for Lexical Research
FAC	Facilities
F	F-score
GPE	Geo Political Entity
HMM	Hidden Markov Model
LDC	Language Data Consortium
LOC	Location
ML	Machine Learning
MDP	Markov Decision Process
ME	Maximum Entropy
MUC	Message Understanding Conference
MISC	Miscellaneous
NE	Named Entity
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NRaD	Naval Research and Development
NL	Nepali Languages
ORG	Organization
POS	Part of Speech
PER	Person
P	Precision
R	Recall
SVM	Support Vector Machine
VC	Vapnik-Chervonenkis

CHAPTER 1
INTRODUCTION

1.1 Introduction

The term Named Entity (NE) was evolved during the sixth Message Understanding Conference (MUC-6, 1995); people who were focusing on Information Extraction (IE) [1]. 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][2]. Though support vector machine (SVM) [3] 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 NE classes, such as Person name (PER), Location name (LOC), Organization name (ORG) and Miscellaneous name (MISC) [4][5]. The Miscellaneous name include date, times, monetary amounts, percentages, designation etc.

NER involves the identification of proper names in text and their classification into different types of named entities (e.g., persons, organizations, locations). NER is not only important in IE but also in lexical acquisition for the development of robust NLP systems [6] [7]. Moreover, NER has proven fruitful for tasks such as documents indexing, and maintenance of databases containing identified named entities.

During the last decade, NER has drawn much attention at MUC, both rule-based and machine learning NER systems have had some success [8]. Previous rule-based approaches have used manually constructed finite state patterns, which match text against a sequence of words. Such system does not need too much training data and can encode expert human knowledge. However, rule-based approaches lack robustness and portability. Each new

source of text requires a significant tweaking of the rules to maintain optimal performance; the maintenance costs can be quite steep.

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 [9][10]. These can be classified into three main classes, namely rule-based NER, machine learning-based NER and hybrid NER [11][12].

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 [13]. 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.

Named Entity Recognition is a subtask of machine translation and information extraction. Named entities are words which belong to certain categories like persons, places, organizations, numerical quantities, expressions of times etc. A large number of techniques have been developed to recognize named entities for different languages. Some of them are Rule based and others are Statistical techniques. Some of the well-known machine learning approaches used in NER are Hidden Markov Model (HMM) [11], Maximum Entropy (ME) (New York University's MENE) in [9], Support Vector Machine[14][15], Decision Tree [16] and CRF [13]. The rule based approach uses the morphological and contextual evidence of a natural language and consequently determines the named entities. This eventually leads to formation of some language specific rules for identifying named entities. The statistical techniques use large annotated data to train a model (like Hidden Markov Model) and subsequently examine it with the test data. Both the methods mentioned above require the efforts of a language expert. An appropriately large set of annotated data is not yet to be made

available for the Nepali Languages. Consequently, the application of the statistical technique for Nepali Languages is not very feasible

1.1.1 Challenges in Named Entity Recognition

Named Entity Recognition was first introduced as part of MUC-6 in 1995 and a related conference MET-1 in 1996 introduced in non-English text. In spite of the recognized importance of names in applications such as search systems, spelling checkers, and document management systems, do not treat proper names correctly. This suggests proper names are difficult to identify and interpret in unstructured text. Generally, names can have innumerable structure in and cross languages. Names can overlap with other names and other words. Simple clues like capitalization can be misleading for English and mostly not present in non western languages like Nepali.

NE identification in Nepali languages is difficult and challenging as:

1.1.1.1 No Capitalization

Capitalization, when available, is the most important feature for NE extraction. English and many other European languages use it to recognize proper names. Orthography of Nepali does not support capitalization. English systems easily recognize acronyms by using capitalization, but in Nepali they are quite difficult to recognize. For example, बिबिसि (transcribed BBC) in Nepali cannot be recognizing as an acronym.

1.1.1.2 Agglutinative Nature

Agglutinative means that some additional features can be added to the word to add more complex meaning. Agglutinative language form sentences by adding a suffix to the root forms of the word. 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. For example, let us consider the root word as राजा and suffix as ईश्वर then if we combine these two words then it becomes राजेश्वर as new word.

1.1.1.3 Proper Name Ambiguity

Ambiguity in proper name present in Nepali language as in English. The names like White are ambiguous in English- name or color. 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. There is a surprising amount of ambiguity even among proper names. For example:

• People vs. Companies: टाटा, फोर्ड
• People vs. Locations: पशुपति (Pashupati)
• People vs. Organizations: त्रिभुवन (person vs. university)
• Acronyms vs. Organizations: MRI (Magnetic Resonance Imaging vs. Mental Research Institute)
• People vs. Months: बैशाख (Baisakh)

1.1.1.4 Word Order

Languages like Nepali have a different word-order than English and some have a free word-order. Nepali mostly has a word order but depending upon the domain the word order is respected. For example, *कमलले पानीको पूरा गिलास पियो र पानीको गिलास कमलले पुरा पियो both translate to Kamal drank a whole glass of water.*

1.1.1.5 Loan words in Nepali

Nepali has a number of loan words. Loan words are words that are not indigenous to Nepali. The named entity recognizer that is based on simple morphological cues will fail to recognize a large number of proper nouns. For example Osama Bin Laden, बिन (Bin) an Arabic cue needs to be used in the middle of the name for the person name.

1.1.1.6 Nested Entities

The named entities that are classified as nested contain two proper names that are nested together to form a new named entity. An example in Nepali is Kathmandu University where Kathmandu is the location name and University marks the whole entity as an organization.

1.1.1.7 Resource Challenges

NER approaches are either based on rule engine or inference engines. In each approach some type of corpus is required; lack of a NE tagged corpus for deriving rules is an issue for Nepali language. Nepali is a resource poor language annotated corpora, name dictionaries; good morphological analyzers, POS taggers etc. are not yet available in the required measure. Although Nepali language have a very old and rich literary history, technological

development are of recent origin. 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.

1.2 Motivation

Nepali is morphologically rich language with great cultural diversities and to build a language model for such language, one has to consider many features. The support vector machine based NER has been implemented in for a Bengali language which is also morphologically rich and shown the outstanding performance [18]. In rich feature set has been used to model the language characteristic [18]. SVM are recently developed supervised learning method having good performance and generalization [19][20]. SVM has been successfully applied in text classification and shown that SVM can handle large features and is resist of over fitting [14]. Another important motivation was to create sufficiently large Nepali, NE tagged data, gazetteers, POS taggers, bilingual dictionaries etc. for NER, Transliteration as well as for other application areas. NER in Nepali language is very difficult and challenging and there is no any works have done for Nepali NER when this work was started.

1.3 State of the art

Research indicates that even state-of-the-art NER systems are brittle, meaning that NER systems developed for one domain do not typically perform well on other domains. Considerable effort is involved in tuning NER systems to perform well in a new domain; this is true for both rule-based and trainable statistical systems.

Early work in NER systems in the 1990s was aimed primarily at extraction from journalistic articles. Attention then turned to processing of military dispatches and reports. Later stages of the automatic content extraction (ACE) evaluation also included several types of informal text styles, such as weblogs and text transcripts from conversational telephone speech conversations. Since about 1998, there has been a great deal of interest in entity identification in the molecular biology, bioinformatics, and medical natural language processing communities. The most common entity of interest in that domain has been names of genes and gene products.

1.4 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 objectives are given below:

1. To analyze the SVM based named entity recognition system for Nepali language.
2. To compare SVM results with different size of training data size.

1.5 Organization of Thesis

The rest of this thesis is organized as: chapter 2 gives a brief discussion of basic concept related to this work, chapter 3 is a survey of the major existing named entity recognition system, chapter 4 presents the methodlogy of the support vector machine based named entity recognition algorithm, chapter 5 gives the detail implementation of support vector machine based named entity recognition, chapter 6 presents the analysis of our work and chapter 7 concludes the thesis, summarizing its achievements and further recommendations.

CHAPTER 2

BACKGROUND AND PROBLEM DEFINITION

2.1Background

2.1.1 Natural Language Processing

NLP has been developed in 1960 as a subfield of Artificial Intelligence and Linguistics [21]. The aim of NLP is studying problems in the automatic generation and understanding of natural language. A Natural Language is any of the languages naturally used by humans, i.e. not an artificial or machine language such as a programming language likes C, java, Perl etc.

NLP is a convenient description for all attempts to use computers to process natural language. NLP is also an area of Artificial Intelligence (AI) research that attempts to reproduce the human interpretation of language for computer system processing. The ultimate goal of NLP is to determine a system of language, words, relations and conceptual information that can be used by computer logic to implement AI language interpretation. NLP includes anything a computer needs to understand natural language (written or spoken) and also generate the natural language. To build computational language systems, we need Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLG systems convert information from computer databases into normal-sounding human language, and NLU systems convert samples of human language into more representation that are easier for computer programs to manipulate. Some of important levels of NLP are as follows:

Phonological Analysis: Phonology is the study of sound system in a language. The minimal unit of sound system is the phoneme which is capable of distinguishing the meaning in the words. The phonemes combine to form a higher level unit called syllable or syllables combine to forms the words. Therefore, the organization of the sounds in a language exhibits the linguistic as well as computational challenges for its analysis. This level deals with the interpretation of speech sounds within and across words. There are, in fact, three types of rules used in phonological analysis: 1) phonetic rules- for sounds within words: 2) phonemic rules- for variations of pronunciation when words are spoken together, and: 3) prosodic rules for- fluctuation in stress and intonation across a sentence. In an NLP system that accepts spoken input, the sounds waves are analyzed and encoded into a digitized signal for

interpretation by various rules or by comparison to the particular language model being utilized.

Morphological analysis: This level deals with the componential nature of words, which are composed of morphemes- the smallest unit of semantic meaning. For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix pre, the root 'registra', and the suffix 'tion'. Since the meaning of each morpheme remains the same across words, humans can break down an unknown word into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning.

Lexical Analysis: At this level, humans, as well as NLP systems, interpret the meaning of individual words. Several types of processing techniques contribute to word-level understanding– the first of these being assignment of a single part-of-speech tag to each word. In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of-speech tag based on the context in which they occur. The lexical level [21] may require a lexicon, and the particular approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon.

Syntactic Analysis: Syntactic analysis [21] must receive the results of morphological analysis to build a structural description of the sentence. The goal of this process, called parsing, is to convert the flat list of words that forms the sentence into a structure that defines the units that are represented by that flat list. The important thing here is that a flat sentence has been converted into a hierarchical structure and that the structures correspond to meaningful units when semantic analysis is performed. The process involves the phrase structure rules and derivation.

Semantic Analysis: It derives an absolute meaning from lexicon; it determines the possible meaning of a sentence in a context. The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between individual words into appropriate objects in the knowledge base or data base. It must create the correct structures to correspond to the way the meaning of the individual words combine with each other. The structures for which no such mapping is possible are rejected [21]. The sentence can be interpreted

semantically taking the semantic inputs from the terminal leaves and composing them in upward fashion till the topmost node. And, finally the semantics of the whole sentence is interpreted. However, example like *colorless green ideas…* in English would be rejected semantically using some other semantic restrictions.

Pragmatic Analysis: It derives knowledge from external commonsense information; it means understanding the purposeful use of language in situations, particularly those aspects of language which require world knowledge [21]. Example: If someone says *the door is open* then it is necessary to know which door (world's knowledge) the word *door* refers to. So, one needs to know the intention of the speaker that the speaker could mean 'to close the door'. It could be a pure statement of fact, could be an explanation of how the cat got in, or could be a request to the person addressed to close the door.

Discourse Integration: The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it [21]. Example: the word "it" in the sentence, "you wanted it" depends on the previous discourse context.

2.1.2 Major Applications of Natural Language Processing

NLP is having a very important place in our day- to-day life due to its large natural language applications. By means of these NLP applications the user can interact with computers in their own mother tongue by means of keyword and a screen. The few NLP processes are:

- Part of speech tagging
- Information retrieval
- Machine translation
- Named entity recognition
- Question answering
- Spoken dialogue system
- Text simplification
- Speech recognition
- Natural language generation etc.

2.1.3 Computational Linguists

Computational linguists are the study of language (i.e. statistical and/or rule-based modeling of natural language) from a computational perspective. Traditionally, computational

linguistic was usually performed by computer scientists who had specialized in the application of computers to the processing of natural language. Computational linguistic often work as members of interdisciplinary teams, including linguists (specifically trained in linguistics), language experts (person with some level of ability in the language relevant to a given project), and computer scientists. In general, computational linguistics draws upon the involvement of linguists, computer scientists, and experts in the artificial intelligence, mathematicians, logicians, cognitive scientists, cognitive psychologists, psycholinguists, anthropologists and neuroscientists, amongst others. Some of the areas of research that are studied by computational linguistics include:

- Computational complexity of a natural language, largely modeled on automata theory, with the applications of context-sensitive grammar.
- Machine translation.
- Design of taggers like POS-taggers.
- Computer-aided corpus linguistics.
- Design of parsers or chunkers for natural languages.
- Computational semantics comprises defining suitable logics for linguistic meaning representation, automatically constructing them and reasoning with them.

2.1.4 Machine learning

It is the recent field of AI which aim to make a machine able to learn as human learns the things. Marvin Minsky (1986) defined learning as "*it is making useful change in the working of our mind*". Machine learning exists in various forms: supervised learning, unsupervised learning, semi supervised or minimally supervised learning, reinforcement learning etc, in its basic form, machine learn the knowledge from some sources and then generalize that knowledge for other instances.

2.1.4.1 Supervised learning

Supervised learning is a technique in which the algorithm uses predictor and target attribute value pairs to learn the predictor and target value relation. Support vector machine is a supervised learning technique for creating a decision function with a training dataset. The training data consist of pairs of predictor and target values. Each predictor value is tagged with a target value. If the algorithm can predict a categorical value for a target attribute, it is

called a classification function. Class is an example of a categorical variable. Positive and negative can be two values of the categorical variable class. Categorical values do not have partial ordering. If the algorithm can predict a numerical value then it is called regression. Numerical values have partial ordering.

2.1.4.2 Unsupervised learning

Unsupervised learning is a technique in which the algorithm uses only the predictor attribute values. There are no target attribute values and the learning task is to gain some understanding of relevant structure patterns in the data. Each row in a data set represents a point in n-dimensional space and unsupervised learning algorithms investigate the relationship between these various points in n-dimensional space. Examples of unsupervised learning are clustering, density estimation and feature extraction.

2.1.4.3 Semi supervised or minimally supervised learning

Semi-supervised learning is a class of machine learning techniques that make use of both labeled and unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent (e.g. to transcribe an audio segment) or a physical experiment (e.g. determining the 3D structure of a protein or determining whether there is oil at a particular location). The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

2.1.4.4 Reinforcement learning

Reinforcement learning is an area of machine learning in computer Science, concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward. The problem, due to its generality, is studied in many other disciplines, such as game theory, control theory, operation research, information theory, simulation based

learning, statistics, and genetic algorithms. In the operations research and control literature the field where reinforcement learning methods are studied is called approximate dynamic programming. The problem has been studied in the theory of optimal control, though most studies there are concerned with existence of optimal solutions and their characterization, and not with the learning or approximation aspects. In economics and game theory, reinforcement learning may be used to explain how equilibrium may arise under bounded rationality.

2.1.4.5 Classification

Given the example data $\{(x_i,y_i), i=1,\ldots\ldots,n\}$, where the x_i is the input vector and the y_i is its associated label or class. Then the classification task is to learn the discriminative function

$y=f(x)$,

which correctly classify the example data and optimized so that it will make minimal error on the classification of unseen data.

If the label 'y' is not discrete as above, then this task is called regression. Based on these examples (x_i,y_i), one is particularly interested to predict the answer for other cases before they are explicitly observed. Hence, learning is not only a question of remembering but also of generalization to unseen cases.

2.1.5 Support Vector Machine

SVM, first introduced by Vapnik [20], 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 [19]. Suppose, we have a set of training data for a two-class problem: $\{ (x_1,y_1),\ldots\ldots\ldots,(x_N, y_N)\}$,where $\mathbf{x_i} \in R^D$ is a feature vector of the i^{th} sample in the training data and $y \in \{+1 ,- 1 \}$ 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: $(w .x)+b =0$ $w \in R^n, b \in R$.

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,

$$sign(f(x, w, b)) \quad 2.1$$

$$f(x, w, b) = < w . x > + b \quad 2.2$$

Where, x is the example to be classified.

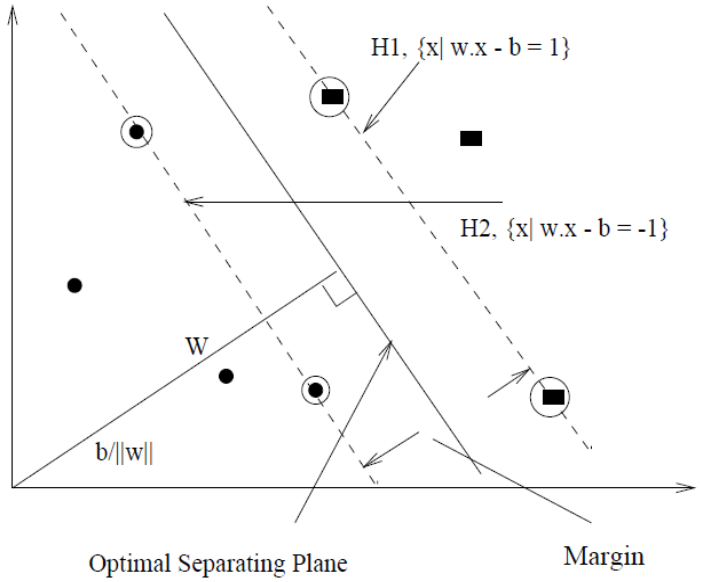


Figure 2.1: Two class SVM with support vectors and supporting hyperplane [20].

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

$$w. x_i - b \geq 1 \; if \; y_i = 1 \quad 2.3$$

And

$$w. x_i - b \leq -1 \; if \; y_i = -1 \quad 2.4$$

In compact form we may combine these two equations in

$$y_i(w.x_i - b) \geq 1 \quad 2.5$$

Or $$-y_i(w.x_i - b) - 1 \leq 0 \quad 2.6$$

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

$$w.x - b = 0 \quad 2.7$$

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 $\frac{\boldsymbol{b}}{|\boldsymbol{w}|}$.

As depicted in Figure 2.1, the perpendicular distance from the origin to hyperplane H1:

$$w.x_i - b = 1 \; is \; \frac{|1+b|}{|w|} \quad 2.8$$

And the perpendicular distance from the origin to hyperplane H2:

$$w.x_i - b = -1 \; is \; \frac{|b-1|}{|w|} \quad 2.9$$

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 $\frac{2}{|w|}$. 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 $\frac{2}{|w|}$ or equivalently minimize the $\frac{|w|^2}{2}$.

Therefore the optimization problem can be formulated as follows

$$\text{Minimize f}=\frac{|w|^2}{2} \quad 2.10$$

$$\text{Subject to constraints} \quad y_i(w.x_i - b) \geq 1 \quad 2.11$$

This problem can be solved by using standard Quadratic programming technique [20].

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 [20].

2.1.5.1 Multi Class SVM

The SVM described in the section 2.1.5 is used for binary classification and which classify data in binary class. But in this work there are five classes, so multiclass SVM is used [14][20]. 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.

2.1.5.2 Kernel Trick: Dual Problem

To deal with nonlinear separation, the same formulation and technique 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. The transformed space is called the feature space. The original data space is called the input space [20].

The basic idea is to map the data in the input space X into feature space F via a nonlinear mapping "∅",

∅:X->F

x-> ∅(x)

After mapping the original data set $\{(x_1, y_1), (x_2, y_2),\ldots\ldots, (x_r, y_r)\}$ becomes:

$\{(\emptyset(x_1), y_1), (\emptyset(x_2), y_2),\ldots\ldots, (\emptyset(x_r), y_r)\}$

Then perform linear separation in this feature space. Geometrically it can be shown as in fig 2.2.

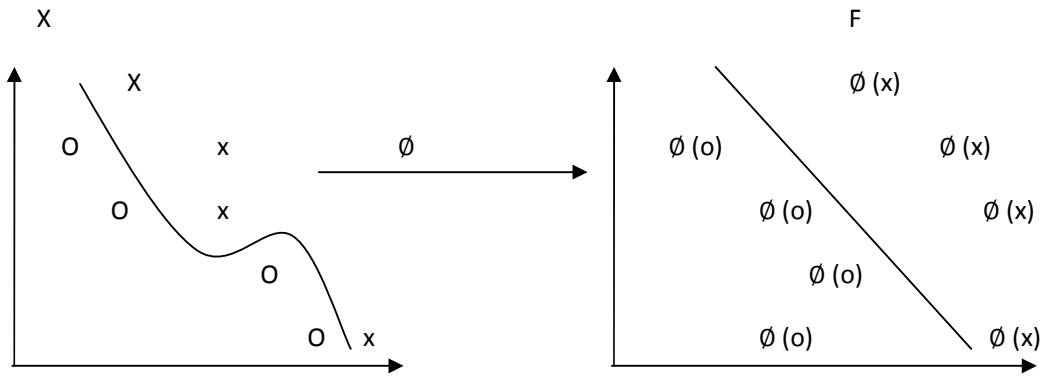


Figure 2.2 Feature Mapping [20]

The potential problem with this explicit data transform and then applying the linear SVM is that it may suffer from the curse of dimensionality [20].

2.1.5.3 Kernel Trick: Inner Product summarization

An inner product represents the dot product of the data vectors used. The dot product of nonlinearly mapped data can be expensive. The kernel trick just picks a suitable function that corresponds to dot product of some nonlinear mapping instead [15][20]. Some of the most commonly chosen kernel functions are linear kernel function, polynomial kernel function, sigmoid kernel function [20]. A particular kernel is only chosen by trial and error on the test set, choosing the right kernel based on the problem or application would enhance SVM's performance. In SVM, the kernel function is represented by K,

K(x,z)= < Ø(x). Ø(z)>.

2.1.5.4 Kernel Functions

The idea of the kernel function is to enable operations to be performed in the input space rather than the potentially high dimensional feature space. Hence the inner product does not need to be evaluated in the feature space. We want the function to perform mapping of the attributes of the input space to the feature space. The kernel function plays a critical role in SVM and its performance. It is based upon reproducing Kernel Hilbert Spaces [21] [22] [23] [24].

$$K(x, x') =< \emptyset(x), \emptyset(x') > \qquad 2.12$$

If K is a symmetric positive definite function, which satisfies Mercer's Conditions [20],

$$K(x, x') = \sum_m^{\infty} \alpha_m \emptyset_m(x) \alpha_m(x'),\ \alpha_m \geq 0, \qquad 2.13$$

$$\iint K(x, x^{'}) g(x) g(x^{'}) dx dx^{'} > 0, g \in L_2 \qquad 2.14$$

Then the kernel represents a legitimate inner product in feature space. The training set is not linearly separable in an input space. The training set is linearly separable in the feature space. This is called the "Kernel trick" [16] [20].

2.1.5.5 SVM for Classification

SVM is a useful technique for data classification. Even though it's considered that Neural Networks are easier to use than this, however, sometimes unsatisfactory results are obtained. A classification task usually involves with training and testing data which consist of some data instances [25]. Each instance in the training set contains one target values and several attributes. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes [20].

Classification in SVM is an example of Supervised Learning. Known labels help indicate whether the system is performing in a right way or not. This information points to a desired response, validating the accuracy of the system, or be used to help the system learn to act correctly. A step in SVM classification involves identification as which are intimately connected to the known classes. This is called feature selection or feature extraction. Feature selection and SVM classification together have a use even when prediction of unknown samples is not necessary. They can be used to identify key sets which are involved in whatever processes distinguish the classes [26].

2.1.6 Optimization

Many situations arise in machine learning where we would like to optimize the value of some function. It turns out that in the general case, finding the global optimum of a function can be a very difficult task. However, for a special case of optimization problems, known as convex optimization problems [27], we can efficiently find the global solution in many cases. Here efficiently has a both practical and theoretical connotation: it means that we can solve many real world problems in a reasonable amount of time, and it means that theoretically we can solve problems in time that depends only polynomial on the problem size.

A convex optimization problem is an optimization problem of the form

Minimize f(x)

Subject to xεC

Where f is a convex problem, C is a convex set.

2.1.7 Evaluating Named Entity Recognition

In this work the following measures are used to evaluate the accuracy of the method or model. The measures taken are: precision (P), recall (R) and F-score (F) [18].

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

$$P = \frac{NEs\ correctly\ retrieved\ by\ the\ system}{NEs\ retrieved\ by\ the\ system} \quad 2.15$$

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

$$R = \frac{NEs\ correctly\ retrieved\ by\ the\ system}{NEs\ present\ in\ the\ test\ set} \quad 2.16$$

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

$$F = \frac{2(P*R)}{P+R} \quad 2.17$$

2.1.8 Methods of NE identification

A number of cues are used to identify named entities. The authors of [26] introduced the concepts of internal evidence (e.g. Ltd. within ORG entities) and external or contextual evidence (e.g. CEO or Dr. before PER entities) by which many may be recognized. Most early systems consisted primarily of manually-built lists of such cues.

The primary alternative approach uses statistical machine learning (ML) in which a system learns patterns from an annotated training corpus, allowing it to predict the most likely NE in a given context. Assuming the availability of appropriate training texts, a single machine-learning system may easily be applied to varying languages, domains or classification schemes.

Two of the top four entrants in MUC-7 used machine learning approaches: Among the early adopters of a ML approach [12] used a series of class-specific Hidden Markov Model

(HMM) in their commercially-successful IdentiFinder to build a model of the language associated with each entity type. Since HMMs rely on having previously seen patterns, their approach uses a number of back-off strategies.

Maximum entropy modeling, as used by [10][17], allowed for many features to be incorporated without a back-off scheme, and their best results were achieved by using the output of multiple high-performance rule-based systems in addition to linguistic features. The machine learning focus of the CONLL-2002 [28] evaluation encouraged various statistical techniques, and allowed for cross-linguistic application and evaluation that was not as feasible with manual rule construction [29].

Models included Support Vector Machines (SVM), AdaBoost, transformation-based learning and maximum entropy modeling. The top system at CONLL-2003 [30] combined the classification decisions of a number of machine learners [30]. In addition to the applicability to new languages and domains, [9][10] emphasizes the fact that statistical systems are able to take advantage of a diverse range of knowledge sources in predicting NE annotations, and are not as subject to the human bias present in manual rule construction. One result of the CONLL-2002 shared task was the realization that while choosing an appropriate machine learning technique affected performance, “the choice of features is at least as important. Their [30] overview of entrants in the CONLL-2003 evaluation compares the types of features used in each competing system.

2.2 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 work, problem of Nepali named entity recognition is addressed. The recognition task is carried out with supervised machine learning using Support Vector machine (SVM) [31]. Feature selection plays a crucial role in the 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 पोखरा गयो । राम <PER> पोखरा <LOC>गयो<O>

When a word is assigned the tag “O”, it does not a named entity word. For example, in the case of राम <PER> पोखरा <LOC>गयो<O>, the word गयो is not named entity.

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, 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.

CHAPTER 3

LITERATURE REVIEW

3.1 Existing Corpus Review

A corpus is a valuable resource in Natural Language Processing. The existence of corpus in correct form makes the NLP a more fruitful process. The most well known corpora for English are probably the Brown Corpus and the Penn Treebank corpus. The Brown University Standard Corpus of Present-Day American English (or just Brown Corpus) was compiled in the 1960s by Henry Kucera and W. Nelson Francis at Brown University, Providence, Rhode Island as a general corpus (text collection) in the field of corpus linguistics. It contains 500 samples of English-language text, totaling roughly one million words, compiled from works published in the United States in 1961. Now – a - days corpora tend to be much larger, and are compiled mainly through projects and initiatives such as the Language Data Consortium (LDC), the Consortium for Lexical Research (RLC) etc. The purpose of these associations is to provide language-related education, research and technology development by creating and sharing linguistic resources: data, tools and standards. Until few years ago, the existing corpora were all of the English Language. Nevertheless, the success and applicability of corpus in Linguistics as well as in NLP, has raised a wide interest and caused its quick extension to other languages. The following are the some example of available Named Entity tagged corpora:

3.1.1 CoNLL-2002 and CoNLL-2003 (British newswire)

The shared task of CoNLL-2003[1] concerns language independent named entity recognition. The author of [27] used concentrate on four types of named entities: persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. The shared task of CoNLL-2002[2] dealt with named entity recognition for Spanish and Dutch [30]. The participants of the 2003 shared task have been offered training and test data for two other European languages: English and German. They have used the data for developing a named entity re cognition system that includes a machine learning component. The shared task organizers were especially interested in approaches that made use of

[1] http://cnts.uia.ac.be/conll2003/ner/

[2] http://cnts.uia.ac.be/conll2002/ner/

resources other than the supplied training data, for example gazetteers and unannotated data. It support the multiple language like Spanish, Dutch, English, German and it contains four NE tags as Person, Location, Organization, Misc.

3.1.2 MUC-6 and MUC-7 (American newswire)

MUC-6[3], the sixth in a series of Message Understanding Conferences, was held in November 1995. This conference, like the previous five MUCs, was organized by Beth Sundheim of the Naval Research and Development group (NRaD) of NCCOSC (previously NOSC). These conferences, which have involved the evaluation of information extraction systems applied to a common task, have been funded by ARPA to measure and foster progress in information extraction.

Prior MUCs had focused on a single task of "information extraction": analyzing free text, identifying events of a specified type, and filling a data base template with information about each such event. Over the course of the five MUCs, the tasks and templates had become increasingly complicated. A meeting in December 1993, following MUC-5, and chaired by Ralph Grishman, defined a broader set of objectives for the forthcoming MUCs: to push information extraction systems towards greater portability to new domains, and to encourage more basic work on natural language analysis by providing evaluations of some basic language analysis technologies.

NYU and NRaD worked together to develop specifications for a set of four evaluation tasks:

- named entity recognition
- conference
- template elements
- scenario templates (traditional information extraction)

These tasks were refined in 1994 and early 1995 through a process of corpus annotation and extensive e-mail discussion by the MUC-6 Planning/Annotation Committee. This was followed by an anonymous "dry run" evaluation, which was held in April 1995.

[3] http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html

The formal MUC-6 evaluation was held in September 1995, and the MUC-6 Conference was held in Columbia, Maryland in November 1995. A proceeding of this conference, including descriptions of the systems from all the participants, is being assembled and will be distributed by Morgan Kaufmann.

The Named Entity task for MUC-6 involved the recognition of entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions. This task is intended to be of direct practical value (in annotating text so that it can be searched for names, places, dates, etc.) and an essential component of many language processing tasks, such as information extraction.

It support the language English and it contains Seven Named Entity (NE) tags as Person, Location, Organization, Time, Date, Percent, Money.

3.1.3 Automatic Content Extraction (ACE)

It contains Five Named Entity (NE) tags as Location, Organization, Person, FAC, GPE (Geo Political Entity). The corpus consists of data of various types annotated for entities, relations and events was created by Linguistic Data Consortium with support from the ACE[4] Program, with additional assistance from LDC. This data was previously distributed as an e-corpus (LDC2005E18) to participants in the 2005 ACE evaluation.

The objective of the ACE program is to develop automatic content extraction technology to support automatic processing of human language in text form.

In November 2005, sites were evaluated on system performance in five primary areas: the recognition of entities, values, temporal expressions, relations, and events. Entity, relation and event mention detection were also offered as diagnostic tasks. All tasks with the exception of event tasks were performed for three languages, English, Chinese and Arabic. Events tasks were evaluated in English and Chinese only. The current publication comprises the official training data for these evaluation tasks.

3.1.4 BBN (Penn Treebank)

The Penn Treebank, a corpus [32] consisting of over 4.5 million words of American English. During the first three year phase of the Penn Treebank project (1989-1992), this corpus has

[4] http://www.ldc.upenn.edu/Projects/ACE/

been annotated for part-of-speech (POS) information. In addition, over half of it has been annotated for skeletal syntactic structure

It contains Twenty two Named Entity (NE) tags as Animal, Cardinal, Date, and Diseases etc.

3.2 A Review of NER Approaches

Considerable amount of work has already been done in the field of NER for English and other language like German, Spanish, Chinese, and Bengali etc. But there is no any work for Nepali language has been done yet. Different approaches like the rule based approach, the stochastic approach and the transformation based learning approach along with modification have been tried and implemented. However, if we look at the same scenario for South-Asian language such as Bangla, Hindi, and Nepali, we find out that not much work has been done in the area of NER

Early work in NER systems in the 1990s was aimed primarily at extraction from journalistic articles. Attention then turned to processing of military dispatches and reports. Later stages of the automatic content extraction (ACE) evaluation also included several types of informal text styles, such as weblogs and text transcripts from conversational telephone speech conversations. Since about 1998, there has been a great deal of interest in entity identification in the molecular biology, bioinformatics and medical natural language processing communities. The most common entity of interest in that domain has been names of genes and gene products.

3.2.1 Conditional Random Fields based Named Entity Recognition

The author of [13][15] 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. In the special case in which the output nodes of the graphical model are linked by edges in a linear chain, CRFs make first order Markov assumption and can be viewed as conditionally trained probabilistic finite automata (FSMs)

The conditional probability of a state sequence
S=<s_1,s_2,……..,s_T>given an observation sequence O=<o_1,o_2,……..,o_T> is calculated as:

$$P\Lambda(S/O) = \frac{1}{Zo} exp \sum_{t=1} \sum_k \lambda_k f_k(S_{l-1}, S_l, O, t) \qquad 3.1$$

Where

$$f_k(S_{l-1}, S_l, O, t) \qquad 3.2$$

is a feature function whose weight λ_k is to be learned via learning. CRFs define the conditional

probability of a label sequence based on total probability over the state sequences,

$$\mathrm{P(l/o)} = \sum_{\mathrm{s:l(s)=1}} \mathrm{P(S/O)} \qquad 3.3$$

where l(s) is the sequence of labels corresponding to the labels of the states in sequences $_{Zo}$ is a normalization factor over all state sequences.

To make all conditional probabilities sum up to 1, we must calculate the normalization factor

$$Z_0 - \sum_s exp \sum_{t=1}^{T} \sum_k \lambda_k f_k(S_{l-1}, S_l, O, t) \qquad 3.4$$

The feature functions could ask arbitrary questions about two consecutive states, any part of the observation sequence and the current position. For example a feature function may be defined to have a value 0 in most cases and have value 1 when s_{t-1}, s_t are certain states and the observation has certain properties.

According to the author of [4] the Recall, Precision and F-Score of CRF based NER is 80.02%, 80.21%, 80.15%, while in case of SVM based NER it is found to be 81.57%, 79.09%, 80.29%,respectively which shows that SVM is better than that of the CRF in the case of Bengali Language.

3.2.2 Maximum Entropy based Named Entity Recognition

The author of [26] had shown that the maximum entropy [ME] [15], 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

$$\boldsymbol{p}(o|h) = \frac{1}{Z(h)} \prod_{j=1}^{k} \propto_j^{f_j(h.o)} \qquad 3.5$$

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

It solves the problem of multiple feature representation and long term dependency issue faced by HMM. It has generally increased recall and greater precision than HMM [33]. It has Label Bias Problem [33]. The probability transition leaving any given state must sum to one. So it is biased towards states with lower outgoing transitions. The state with single outgoing state transition will ignore all observations. To handle Label Bias Problem we can change the state-transition.

According to the author of [4] the Recall, Precision and F-Score of ME based NER is 78.64%, 76.89%, 77.75%, while in case of SVM based NER it is found to be 81.57%, 79.09%, 80.29%,respectively which shows that SVM is better than that of the ME based NER for Bengali Language.

3.2.3 Hidden Markov Model Named Entity Recognition

The author of [12] 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) have enjoyed great success in other textual classification problems most notably part-of-speech tagging [11]. Given this success, and given the locality of phenomena which indicate names in text, such as titles like "Mr." preceding a person name, they [11] 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, [12] 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".

It is advantageous as its basic theory is elegant and easy to understand. Hence it is easier to implement and analyze [33]. It uses only positive data, so they can be easily scaled. The main disadvantage of this method is in order to define joint probability over observation and label sequence HMM needs to enumerate all possible observation sequence. Hence it makes various assumptions about data like Markovian assumption i.e. current label depends only on

the previous label. Also it is not practical to represent multiple overlapping features and long term dependencies. Number of parameter to be evaluated is huge. So it needs a large data set for training.

3.2.4 Decision Tree based Named Entity Recognition

Decision tree is a classification approaches which construct the tree in top down manner using the attribute the data satisfies.

The decision tree [34] uses part of speech, character type, and special dictionary information to determine the probability that a particular type of name opens or closes at a given position in the text. The output is generated from the consistent sequence of name opens and name closes with the highest probability. This system does not require any human adjustment. Experiment indicate good accuracy with a small amount of training data, and demonstrate the systems portability.

Using the training, a decision tree is built. It learns about the opening and closing of named entities based on the three kinds of information of the previous, current and following tokens. The three types of information are the part- of- speech, character type and special dictionary information which contain the list of entities created based on JUMAN [16] dictionary entries.

3.2.5 Support Vector Machine based Named Entity Recognition

In support vector machine [19] 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 [19] 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 [23] was proposed by Yamada et al. [22] for Japanese. His system is an extension of Kudo's chunking system [31] that gave the best performance at CoNLL-2000 shared tasks. The other SVM-based NER systems can be found in [2][18].

3.3 Knowledge sources for NER

Named Entity Recognition systems are only as reliable as their training sources. Rule-based, statistical and unsupervised systems alike may make use of lists of names categorized into entity types, often referred to as gazetteers[5]. Gazetteer-based approaches require additional methods to resolve ambiguity and unknown names. Machine learning approaches to NER are able to take advantage of learned patterns, and such knowledge is contingent on the availability of training data. This suggests that additional sources of annotated training data are able to benefit statistical NER.

3.3.1 Gazetteers

It has often been assumed that reasonable NER performance can be achieved merely by list-lookup for familiar names [29]. With the assumption that larger categorized lists of names may improve system recall, a number of approaches have been implemented to automatically acquire such lists from the web often using context patterns and bootstrapping [29] or from Wikipedia [7] bring extensive arguments against the assumption that larger gazetteers aid NER:

- such lists need to be enormous and cover naming variations;
- There is ambiguity with common nouns and between entities [29] report a perfect-recall list lookup approach.
- Linguistic data sparseness means no list can approach completeness.

NER system [35] was tested without a gazetteer and gave only small increases in error for ORG and PER classes, but significant performance losses (from 6 to 48% error) for LOC, which were largely alleviated with a short list of common locations. This particular dependence on geographic gazetteers seems to be system-specific, though: their implementation relies initially on lists of cues (e.g. Mrs., Ltd., Inc.) that are less available for location identification. Machine learning techniques have been able to produce high accuracy for LOC without gazetteer information, and some authors choose to use only personal name gazetteers. The authors of [29] confirm that gazetteer size is not key, and that lists extracted from the web are most effective when filtered. Evaluations from CONLL-2003 nonetheless reported up to 22% error reduction for the English corpus and 15% for German when

[5] Historically the term gazetteer has referred to exhaustive lists of geographic names with associated information; here the term is applied more generally to extensive lists of names of any class.

gazetteer data was incorporated although one of the best performers in both languages used none at all . It seems that selectivity in the use of lists can provide greater performance value than large gazetteers.

In a novel extension to the use of lists [29] note that for statistical systems, gazetteers do not need to group entities into the target entity classes. Any knowledge source which can be used to attach the same label to semantically similar entities may be added as a feature for machine learning. They improved NER performance by 1.6% F-score with a feature based on a cluster labels for entities. While this approach may have advantages over traditional list methods in resolving ambiguity, it is still only able to provide an advantage for known entities.

3.3.1 Training Corpora

Data-driven statistical approaches are popular in contemporary computational linguistics, although the time and monetary costs of manually producing training corpora are prohibitive [29]. For NER training, the only data widely available are corpora used in conference evaluations of named entity technology (MUC, IEER, ACE and CONLL), or for specific domains such as biomedicine, and many require purchase, relying on copyrighted materials. While these are useful for evaluating and comparing NER systems, they are not necessarily sufficient training data to produce systems capable of high-accuracy real-world NER. For instance, the top-performing system in CONLL-2003 made auxiliary use of two classifiers trained on a private data collection. Training corpora provide patterns and context that NER systems can learn, unlike gazetteers which although easily generated do not provide sufficient information for machine learners. Hence it is appealing to find low-cost ways to generate new corpora.

One approach involves extracting sentences from the web [29] used a simple approach of searching the web for a given unambiguous named entity (in Korean), extracting sentences that contain it, and tagging the known entity for use in a training corpus. This is limited in that it does not provide any evidence for disambiguation, and cannot produce annotated sentences that contain multiple entities (unless all are known). It also loses the applicability of features related to long-distance dependencies that some have found advantageous for NER but unlike the use of lists alone may help identify sentence-internal patterns for NE recognition while simply discarding more difficult sentences. The initial corpus produced by

[29] was much larger than available Korean annotated corpora, and produced marginally improved results in an NER task.

CHAPTER 4

METHODOLOGY

The implementation model for NER is given in Figure 4.1. This describes top level data flow diagram of NER problem, used in this work. The proposed system framework consist preprocessing, feature extraction, training and learning the data for support vector machine algorithm.

Preprocessing engine has Nepali corpus as input. After preprocessing, only important data is stored and feature vectors are extracted from the preprocessed data. Training corpus is the most powerful and is the heart of Named Entity Recognition.

4.1 Implementation Model for Nepali NER

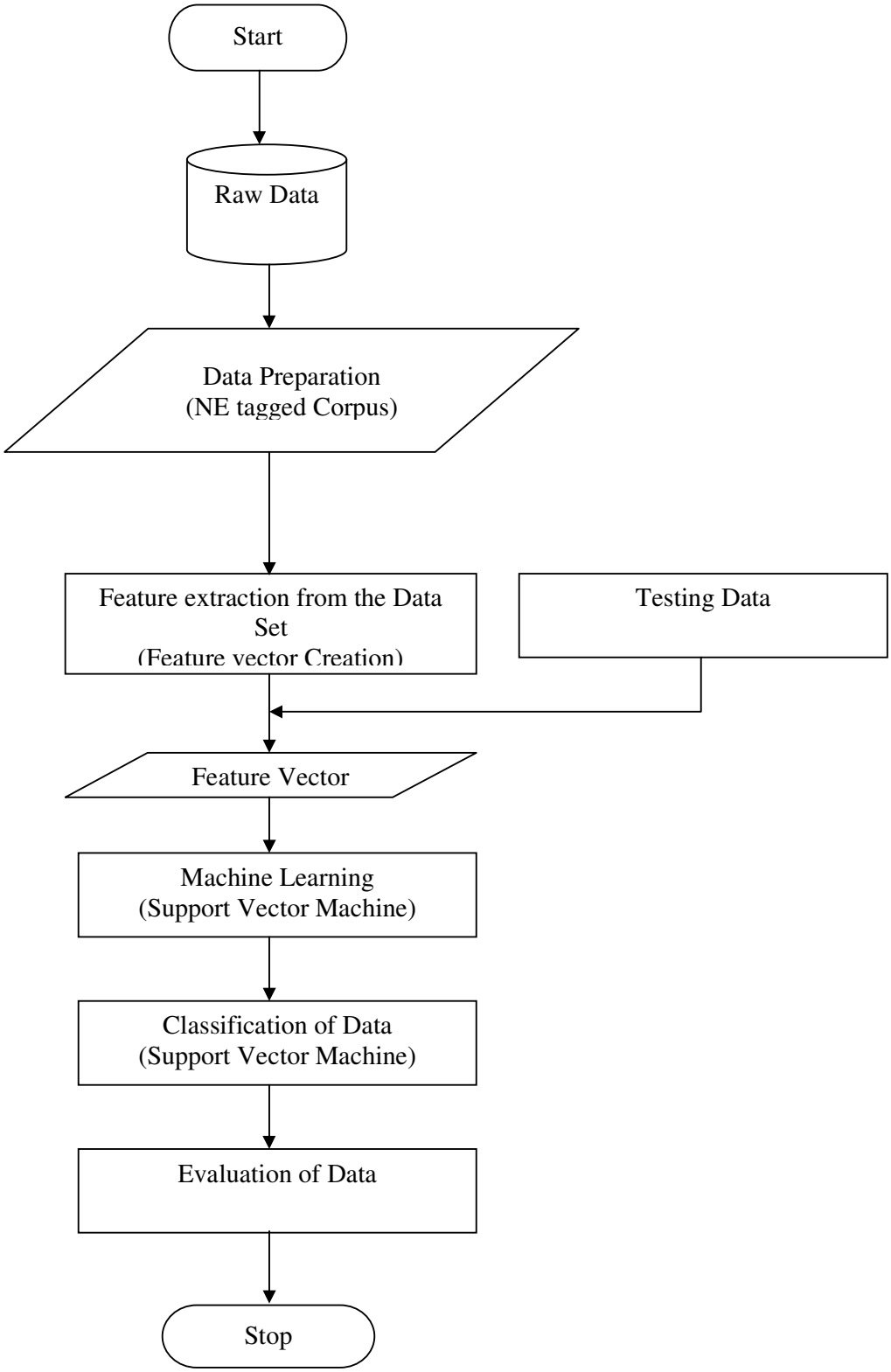


Figure 4.1 Implementation Model for Nepali NER

4.2 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 corpus for training was tagged manually for this thesis.

4.3 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) framework [18]. 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 are extracted. The features used in this work are taken from [18]. Following are the details of the set of features that will be apply to the NER task:

1. First word: This is used to check whether the current token is the first word of the sentence or not. Though Nepali is relatively free order languages, the first word of the sentence is most likely a NE as it appears in the subject position most of the time.
2. Word length: This binary valued feature is used to check whether the length of the current word is less than two or not. This is based on the observation that the very short words are rarely NEs.
3. Digit features: Several binary valued digit features have been defined depending upon the presence and/or the number of digits in a token (e.g., ContainsDigit [token contains digits], FourDigit [token consists of four digits], TwoDigit [token consists of two digits]), combination of digits and punctuation symbols (e.g., ContainsDigitAndComma [token consists of digits and comma], combination of digits and symbols (e.g., ContainsDigitAndSlash [token consists of digit and

slash], ContainsDigitAndHyphen [token consists of digits and hyphen], ContainsDigitAndPercentage [token consists of digits and percentages]). These binary valued features are helpful in recognizing miscellaneous NEs, such as time expressions (Age, Date, Year), measurement expressions (Weight, Height etc) and numerical numbers etc.

4. Gazetteer Lists: Various gazetteer lists are used.
    i. Person name: This list contains the name of persons. The feature PersonName is set to +1 for the current word.
    ii. Location name: This list contains the location names and the feature LocationName is set to +1 for the current word.
    iii. Organization name: This list contains the organization names and the feature OrgnizationName is set to +1 for the current word.
    iv. 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.
    v. 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.
    vi. PersonPrefix: This list contains the person prefix such as श्री, श्रीमान, श्रीमति etc.
    vii. MiddleName: This list contains nepali middle name such as बहादुर, कुमार, कुमारी, देबी, राज, प्रसाद etc.
    viii. SurName: This list contains nepali sur name such as बम, पन्त, जोशि, भट्ट, दाहाल etc.
    ix. CommonLocationWord: This list contains common location word such as रोड, बाटो, राजमार्ग, नगर etc.
    x. Action Verb: A set of action verbs like सुन, भन, गर, खाउ, जाउ etc. often determine the presence of person names. Person names generally appear before action verbs.

xi. Designation Word: This list contains designation word such as प्रोफेशर, डा., मन्त्रि, रास्ट्रपति, सचिब, अध्यक्ष, महासचिव etc.

xii. Organization Suffix Word: This list contains organization suffix word such as मिल, प्रालि, कम्पनि, समिति, संघ, कार्यालय etc.

4.4 Problem Setting

Named entity recognition is a multiclass classification problem since in natural language there are more than two tags. As an instance, for this work, the five tags are used to define to cover all grammatical categories and in which four tags are NE and fifth tag is used to represent the word which does not belongs to the named entity i.e. other than NE. In this work number of tag represents the number of classes. Since SVM are binary classifier so binarization of problem must be performed before apply them to NE tagging. [20] Has suggested the one vs. rest binarization of problem i.e. a SVM is trained for each NE tag in order to distinguish this class and the rest. When tagging the word, the most confident prediction among the all binary SVM is selected. Hence the support vector machine used in this dissertation work is in fact the implementation of support vector machine with one verses rest method is explain in section 4.6.1.

For this work $SVM^{multiclass}$ algorithm [14] is used for classification of the given data into their proper classes. To take the time efficiency into account, the linear kernel type is used.

4.5 Named Entity Tagset for Nepali NER

In this work, the NE tagset used have been further subdivided into the detailed categories in order to denote the boundaries of NEs properly. Table 4.1 shows examples.

NE Tag	Meaning	NE examples
PER	Person name	जनक <PER> जनकजोशी<PER>
LOC	Location name	बुटवल<LOC> बुटवलराजमार्ग<LOC>
ORG	Organization name	त्रिभुवनविश्वविध्यालय <ORG>
MISC	Miscellaneous name	बैशाख<MISC> बैशाख१५ <MISC>
O	Words that are not NE	भिर <O>,पेश <O>

Table 4.1 Named Entity examples

4.6 Support Vector Machine Algorithm

The optimization problem for SVM in its basic form is

$$\text{Minimize } f = \frac{|w|^2}{2} \quad 4.1$$

$$\text{Subject to constraints } y_i(w.x_i - b) \geq 1 \quad 4.2$$

The equivalent dual formulation of this problem can be written

$$\text{Min}_{w,b}\text{max}_{\alpha}\frac{1}{2}\|W\|^2 - \sum_j \alpha_j[y_j(< x_j.w > +b) - 1] \quad 4.3$$

$$\text{Subject to } \alpha_j \geq 0 \quad 4.4$$

Where α's are Lagrangian multipliers [20].

With some specification, the equations can be written as

$$\text{Min}_{w,b}\text{max}_{\alpha}\frac{1}{2}\|W\|^2 - \sum_j \propto_j [y_j(< x_j.w > +b)] + \sum_j \alpha_j \quad 4.5$$

$$\text{Subject to } \alpha_j \geq 0$$

Wishing to minimize both w and b while maximizing α's leaves us to determine the saddle points. The saddle points [20] correspond to those values where the rate of change equals to zero. This is done by differentiating the Lagrangian-primal (LP) equation (4.3), with respect to w and b and setting their derivatives to zero.

$$\frac{\delta L}{\delta w} = 0 \rightarrow w - \sum_j \alpha_j y_j X_i = 0 \quad 4.6$$

$$w = \sum \alpha_j y_j X_i \quad 4.7$$

$$\frac{\delta L}{\delta b} = 0 \rightarrow -\sum_j \alpha_j y_j X_i = 0 \quad 4.8$$

$$\sum_j \alpha_j y_j X_i = 0 \quad 4.9$$

Putting the value of (4.7) and (4.9) in above equation (4.3), we have

$$max_\alpha - \frac{1}{2}\sum_j \alpha_j y_j X_j + \sum_j \alpha_j y_j X_j + \sum_j \alpha_j \quad 4.10$$

$$\text{Equal to } max_\alpha \sum_j \alpha_j - \frac{1}{2}\sum_j \alpha_j y_j X_j \, \alpha_j y_j X_j \quad 4.11$$

Now the optimization problem becomes

$$\max_\alpha L = \sum_j \alpha_j - \frac{1}{2}\sum_{i,j} \alpha_j y_j X_j \, \alpha_j y_j X_j \quad 4.12$$

$$\text{Subjected to } \sum_j \alpha_j y_j = 0 \quad 4.13$$

$$\text{Where, } \alpha \geq 0$$

This is the quadratic optimization problem and can be solved using the decomposition algorithm as in [14]. Decomposition algorithm breaks the whole optimization problem into smaller sets and solves each set iteratively.

4.6.1 Multi Class SVM for classification

For classification problems with multiple classes, different approaches are developed in order to decide whether a given data point belongs to one of the classes or not. The most common approaches are those that combine several binary classifiers and use a voting technique to make the final classification decision. These include: One-Against-All [20], One-Against-One [26], Directed Acyclic Graph (DAG) [26], and Half-against-half method [26]. A more complex approach is one that attempts to build one Support Vector Machine that separates all

classes at the same time. In this section the brief introduction of these multi-class SVM [29] approaches is given.

The SVM described in section 2.1.5 is used for binary classification and which classify data in binary class. But in the case of NER there are five classes, so multiclass SVM is 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> रेडियोनेपाल <ORG>को<O> पोखरा< LOC>स्थित<O> स्टेशनमा<O> काम<O> गर्छ<O>

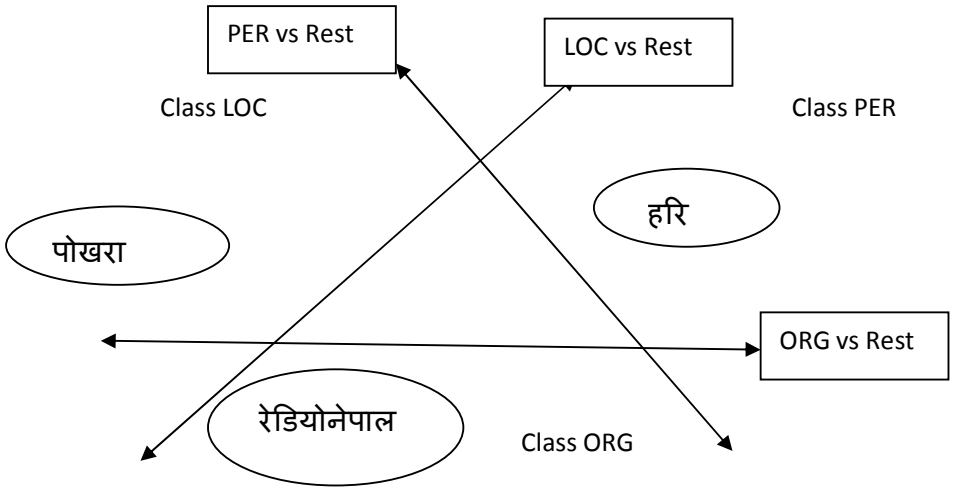


Figure 4.2: One Vs rest classification approaches for NER.

4.6.1.1 One-Against-All Multi-Class SVM

One-Against-All [20] is the earliest and simplest multi-class SVM. For a K-class problem, it constructs K binary SVMs. The ith SVM is trained with all the samples from the ith class against all the samples from the other classes. To classify a sample x, x is evaluated by all of the K SVMs and the label of the class that has the largest value of the decision function is selected.

For a K-class problem, One-Against-One maximizes K hyperplane separating each class from all the rest. Since all other classes are considered negative examples during training of each binary classifier, the hyperplane is optimized for one class only.

4.6.1.2 One-Against-One or Pairwise SVM

One-Against-One [26] constructs one binary machine between pairs of classes. For a K-class problem, it constructs K(K-1)/2 binary classifiers. To classify a sample x, each of K(K-1)/2 machines evaluate x and casts a vote. Finally, the class with the most votes is chosen. Since

One-Against-One separates two classes at a time, the separating hyperplane identified by this approach are tuned better than those found with One-Against-All.

4.6.1.3 All-Together or All-At-Once SVM

An All-Together [26] multi-classification approach is computationally more expensive yet usually more accurate than all other multi-classification methods. This approach builds one SVM that maximizes all separating hyperplane at the same time. Training data representing all classes is used to generate the trained model. With this approach, there are no unclassifiable regions as each data point belongs to some class represented in the training dataset.

The All-together multi-class SVM poses a complex optimization problem as it maximizes all decision functions at the same time. The training time is very slow which makes the approach so far unusable for real-world problems with a large data set and/or a high number of classes.

CHAPTER 5
IMPLEMENATION

5.1 Overview

Java is a programming language originally developed by James gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low level facilities than either of them. Java applications are typically compiled to byte code that can run on any java virtual machine (JVM) regardless of computer architecture. Java is general purpose, concurrent, class-based, object-oriented language that is specially designed to have as few implementation dependencies as possible. Java is one of the most popular languages in use.

Netbeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others. The Netbeans IDE is written in Java and can run on Windows, Linux and other platforms supporting a compatible JVM. Netbeans IDE is an Oracle sponsored free and open source Java integrated development environment.

5.2 SVM Implementation: SVM^{multiclass}

For this works, the SVM^{multiclass} [14] is used. SVM^{multiclass} is an implementation of Support Vector Machines (SVMs) in C programming language. Main features of this system is that we can integrate our own custom kernel very easily. Because of steepest feasible descent and caching of kernel evaluations, SVM^{multiclass} is real fast. It can easily handle thousands of support vectors and several hundred-thousands of training examples.

At first, system learns from training file using customized kernel function and creates a model file. Model file basically learn all the support vectors. This model file is used for classifying new examples. After testing is complete, it produces a prediction file which contains the confidence value of each example for that classification.

5.3 Algorithm for Training

INPUT: Formatted train file.

OUTPUT: SVM models learned for all NE tags.

Step 1: Read the train file.

Step 2: Construct support vector for each tokens.

Step 3: Do step 2 for all the tokens presented in the train file.

Step 4: Use $SVM^{multiclass}$ to learn the model.

Step 5: Stop.

5.4 Algorithm for Testing

1. Read input text.
2. Construct feature vector for each word.
3. Use Multiclass SVM to classify input text.
4. Stop.

5.5 Dictionary

There is no any NE dictionary for Nepali text is created ever yet so a dictionary is created manually from the training corpus which is taken different daily newspapers of 2012 as ekantipur[6], nagriknews[7] as well as from web as onlinekhabar[8], which contains all possible NEs of each NE class. The dictionary contains the four lists as person list which contains the person name, location list which contains the location name, organization list which contains the organization name and miscellaneous list which contains the miscellaneous named entities such as date, time, month name, day name etc.

[6] ekantipur.com

[7] nagariknews.com

[8] onlinekhabar.com

5.6Feature Set

The features used in this work are tabulated in the following table 5.1

Features	**Descriptions**
isDigit	$isDigit_i = \{^{1,\ if\ wi\ contains\ the\ digit}_{-1,\ Othewise}$
fourDigit	$fourDigit_i = \{^{1,\ if\ wi\ contains\ the\ four\ digit}_{-1,\ Othewise}$
digitPercentage	$digitPercentage_i = \{^{1,\ if\ wi\ contains\ the\ digit\ and\ percentage}_{-1,\ Othewise}$
isDate	$isDate_i = \{^{1,\ if\ w_i\ contains\ the\ date}_{-1,\ Othewise}$
isPersonPrefix	$isPersonPrefix_i = \{^{1,\ if\ w_i\ contains\ person\ prefix}_{-1,\ Othewise}$
isMiddleName	$isMiddleName_i = \{^{1,\ if\ w_i\ contains\ Middle\ Name}_{-1,\ Othewise}$
isSurName	$isSurName_i = \{^{1,\ if\ w_i\ contains\ Sur\ Name}_{-1,\ Othewise}$
isCommonLocationWord	$isCommonLocationWord_i = \{^{1,\ if\ w_i\ contains\ common\ loc.\ word}_{-1,\ Othewise}$
isActionVerb	$isActionVerb_i = \{^{1,\ if\ w_i\ contains\ action\ verb}_{-1,\ Othewise}$
isDesignationWord	$isDesignationWord_i = \{^{1,\ if\ w_i\ contains\ designation\ word}_{-1,\ Othewise}$
isOrganizationSuffixWord	$isOrganizationSuffixWord_i = \{^{1,\ if\ w_i\ contains\ org.\ suffix\ word}_{-1,\ Othewise}$
isPersonName	$isPersonName_i = \{^{1,\ if\ w_i\ contains\ person\ name}_{-1,\ Othewise}$
isOrganizationName	$isOrganizationName_i = \{^{1,\ if\ w_i\ contains\ organization\ name}_{-1,\ Othewise}$
isLocationName	$isLocationName_i = \{^{1,\ if\ w_i\ contains\ location\ name}_{-1,\ Othewise}$
First word	$FirstWord_i = \{^{1,\ if\ w_i\ first\ word\ of\ the\ sentence}_{-1,\ Othewise}$
Word length	$WordLength_i = \{^{1,\ if\ w_i \geq 3}_{-1,\ Othewise}$
isMiscellaneous	$isMiscellaneousName_i = \{^{1,\ if\ w_i\ contains\ Miscellaneous\ name}_{-1,\ Othewise}$
isNotNE	$isNotNE_i = \{^{1,\ if\ w_i\ contains\ NotNE}_{-1,\ Othewise}$

Table 5.1 Description of the features, Here *i* represents the position of the current word and w_i represents the current word.

5.7Sample Input and Output

5.7.1 Input

१८माघ, विराटनगर । आन्दोलनरत काँग्रेस-एमालेसहितका विपक्षी दलहरुले आन्दोलनवाटै प्रधानमन्त्री डा. बाबुराम भट्टराई नेतृत्वको सरकार ढलाउने बताइरहेका बेला विपक्षी दल काँग्रेसकी नेतृ सुजाता कोइरालाले भने आफूहरूको आन्दोलनले सरकारलाई कुनै असर नगर्ने वताउनु भएको छ ।

विहीवार मोरङको विराटनगर विमानस्थलमा पत्रकारहरुसंग कुराकानी गर्दै कोइरालाले आफूहरुले गरिरहेको आन्दोलनले भट्टराई सरकार नढल्ने समेत वताउनुभयो । नेतृ कोइरालाले विपक्षी दल र सत्तारुढ दल दुबै सडकमा आउँदा मुठभेडको स्थिति आउने भएकाले सहमतिको विकल्प नरहेको वताउनुभयो । मुलुकलाई निकास दिन दलहरु मिल्नुको विकल्प नरहेको भन्दै उहाँले मुठभेड हुन नदिन दलहरुले सहमति र सहकार्यवाट अघि बढ्नुपर्ने धारणा राख्नु भयो ।
कोइरालाले सरकार पक्ष र विपक्षी दुबै भारतको दिल्लीमा भएको १२ बुँदे सम्झौता विपरित आन्दोलनमा उत्रिएको आरोप लगाउनुभयो । नेतृ कोइरालाले आन्दोलन भन्दा विपक्षी वर्तमान सरकारमै सहभागी भएर अघि बढे एमाओवादीलाई परास्त गर्न सकिने वताउनुभयो ।
१८ माघ, काठमाडौं । दैलेखका पत्रकार डेकेन्द्र थापाको हत्यासम्बन्धी मुद्दामा स्थानीय जिल्ला अदालतमा बिहीबार शुरु भएको सुनुवाइ लम्बिएको छ । बिहान ११ बजेबाट प्रारम्भ भएको बादी प्रतिवादी वकिलहरुको बहस नसकिएकाले शुक्रबार पुनः सुनुवाइ हुने भएको छ । थापाको मुद्दामा सुनुवाइ प्रारम्भ हुँदा मृतक थापाका आफन्त, पत्रकार र स्थानीयवासीहरुले इजलास कक्ष खचाखच भरिएको थियो । मुद्दाको सुनुवाइ न्यायाधीश रामकृष्ण भट्टको एकल इजलाजमा परेको छ । विहीबार सरकारी वकिलसहित वादी अर्थात् सरकारका पक्षबाट ५ वकिलले बहस गरेका छन् । पत्रकार थापाको हत्या आरोपमा पक्राउ परेका प्रतिवादीहरुका तर्फबाट दुई वकिलले विहीबार बहस गरेका छन् । प्रतिवादीका अर्का वकिल गोपाल सिवाकोटीको बहस

आधामात्रै सकिएकाले शुक्रबार पुनः हुने भएको छ । शुक्रबार पनि बहस लम्बियो भने थुनछेक आदेश आइतबार मात्रै हुनसक्ने स्रोतले बतायो ।

5.7.2 Output

Here 'O' represents the other which are not NE.

१८माघ O विराटनगर LOC आन्दोलनरत O काँग्रेस-एमालेसहितका O विपक्षी O दलहरुले O आन्दोलनवाटै O प्रधानमन्त्री MISC डा. MISC बाबुराम PER भट्टराई O नेतृत्वको O सरकार O ढलाउने O बताइरहेका O बेला LOC विपक्षी O दल PER काँग्रेसकी O नेतृ O सुजाता O कोइरालाले O भने O आफूहरूको O आन्दोलनले O सरकारलाई O कुनै O असर O नगर्ने O वताउनु O भएको O छ PER विहीवार O मोरङको O विराटनगर LOC विमानस्थलमा O पत्रकारहरुसंग O कुराकानी O गर्दै O कोइरालाले O आफूहरुले O गरिरहेको O आन्दोलनले O भट्टराई PER सरकार O नढल्ने O समेत O वताउनुभयो O नेतृ O कोइरालाले O विपक्षी O दल O र O सत्तारुढ O दल PER दुबै O सडकमा O आउँदा O मुठभेडको O स्थिति O आउने O भएकाले O सहमतिको O विकल्प O नरहेको O वताउनुभयो O मुलुकलाई O निकास O दिन O दलहरु O मिल्नुको O विकल्प O नरहेको O भन्दै O उहाँले O मुठभेड O हुन O नदिन O दलहरुले O सहमति O र O सहकार्यवाट O अघि O बढ्नुपर्ने O धारणा O राख्नु O भयो O कोइरालाले O सरकार O पक्ष O र LOC विपक्षी O दुबै O भारतको O दिल्लीमा O भएको O १२ O बुँदे O सम्झौता O विपरित O आन्दोलनमा O उत्रिएको O आरोप O लगाउनुभयो O नेतृ MISC कोइरालाले O आन्दोलन O भन्दा O विपक्षी O वर्तमान O सरकारमै O सहभागी O भएर O अघि O बढे O एमाओवादीलाई O परास्त O गर्न O सकिने O वताउनुभयो O १८ O माघ MISC काठमाडौं LOC दैलेखका O पत्रकार O डेकेन्द्र PER थापाको O हत्यासम्बन्धी O मुद्दामा O स्थानीय O जिल्ला O अदालतमा O बिहीबार MISC शुरु O भएको O सुनुवाइ O लम्बिएको O छ O बिहान O ११ O बजेबाट O प्रारम्भ O भएको O बादी O प्रतिवादी O वकिलहरुको O बहस O नसकिएकाले O शुक्रबार MISC पुनः O सुनुवाइ O हुने O भएको O छ O मुद्दामा O सुनुवाइ O प्रारम्भ O हुँदा O मृतक O थापाका O आफन्त O पत्रकार O र O स्थानीयवासीहरुले O इजलास O कक्ष O खचाखच O भरिएको O थियो O मुद्दाको O सुनुवाइ O न्यायाधीश O रामकृष्ण PER भट्टको O एकल O इजलाजमा O परेको O छ O विहीबार O सरकारी O वकिलसहित O वादी O अर्थात् O सरकारका O पक्षबाट O ५ O वकिलले O बहस

o गरेका o छन् o पत्रकार o थापाको o हत्या o आरोपमा o पक्राउ o परेका o प्रतिवादीहरुका o तर्फबाट o दुई o वकिलले o विहीबार o बहस o गरेका o छन् o प्रतिवादीका o अर्का o वकिल MISC गोपाल PER सिवाकोटीको o बहस o आधामात्रै o सकिएकाले o शुक्रबार MISC पुनः o हुने o भएको o छ PO शुक्रबार MISC पनि o बहस o लम्बियो o भने o थुनछेक o आदेश o आइतबार MISC मात्रै o हुनसक्ने o स्रोतले o बतायो o

CHAPTER 6

TESTING AND ANALYSIS

6.1 The Dictionary Data Statistics

To analyze the results of the NER, first of all there is a need of training corpus to test on it. The training corpus is to be build in order to make the training possible. For this work NE tagged corpus has created which is manually tagged and which contains 29,298 unique words. The detail description of training corpus is shown if table 6.1

Dictionary	**No. of entries**
Person Name	5128
Location Name	4970
Organization Name	4608
Miscellaneous Name	5306
Other (Which are not NE)	9286
Total Entry in Dictionary	29298

Table 6.1 NE distribution in Dictionary

6.2 Gazetteer Lists

Gazetteer	**No. of entries**
Person Name	5128
Location Name	4970
Organization Name	4608
Month Name	12
Day Name	7
Person Prefix	7
Sur Name	104
Action Verb	11
Designation Word	50
Organization Suffix Word	16
Middle Name	20
Common Location Name	9

Table 6.2 Number of gazetteers in gazetteer list

6.3 Test Data Analysis

For testing purpose test data are prepared from different news sites. The learning nature of recognizer is evaluated with the different size of training data. The testing is done for three different sizes of the training data. The size of the training data is gradually increased and the performance of recognizer is observed. For each training size there are 10 different experiments performed on the basis of size of the test data. The size of the test data is different in each experiment. The result given by the different experiment is tabulated in section 6.4.

6.4 Result and Discussion

6.4.1 Experiment No. 1(Training Size 5000 tokens)

The sample input for experiment No. 1 is presented in Appendix A.

Experiment No.	Size of Test Data (in tokens)	Precision (%)	Recall (%)	F-Score (%)
1	1000	69.61	80.46	74.64
2	1500	68.08	80.41	73.73
3	2000	67.88	80.53	73.67
4	2500	62.07	80.58	70.13
5	3000	63.40	80.52	70.94
6	3500	64.69	80.26	71.64
7	4000	67.02	80.50	73.14
8	4500	65.48	80.42	72.19
9	5000	65.06	80.38	71.91
10	5500	66.02	80.22	72.43

Table 6.3 Experiment No. 1(Training Size 5000 tokens)

6.4.2 Bar Diagram of Experiment No. 1

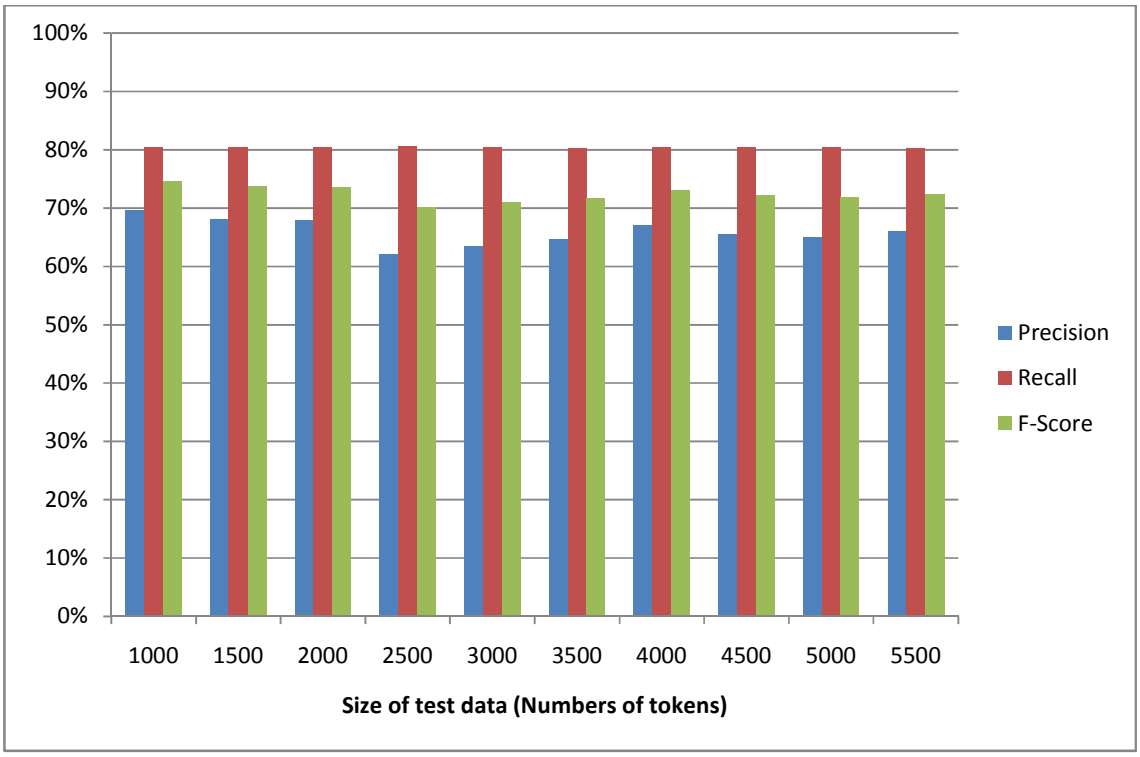


Figure 6.1 Bar Diagram for Precision, Recall and F-Score for training size 5000 tokens.

6.4.3 Experiment No. 2(Training Size 15000 tokens)

The sample input for experiment No. 2 is presented in Appendix B.

Experiment No.	Size of Test data (in tokens)	Precision (%)	Recall (%)	F-Score (%)
1	1000	82.58	97.96	89.62
2	1500	84.84	97.49	90.73
3	2000	86.57	98.51	92.15
4	2500	79.52	94.72	86.46
5	3000	75.53	95.73	84.44
6	3500	83.12	98.08	89.98
7	4000	84.42	97.91	90.66
8	4500	81.41	97.12	88.57
9	5000	84.63	98.16	90.90
10	5500	84.07	97.09	90.11

Table 6.4. Experiment No. 2(Training Size 15000 tokens)

6.4.4 Bar Diagram of Experiment No. 2

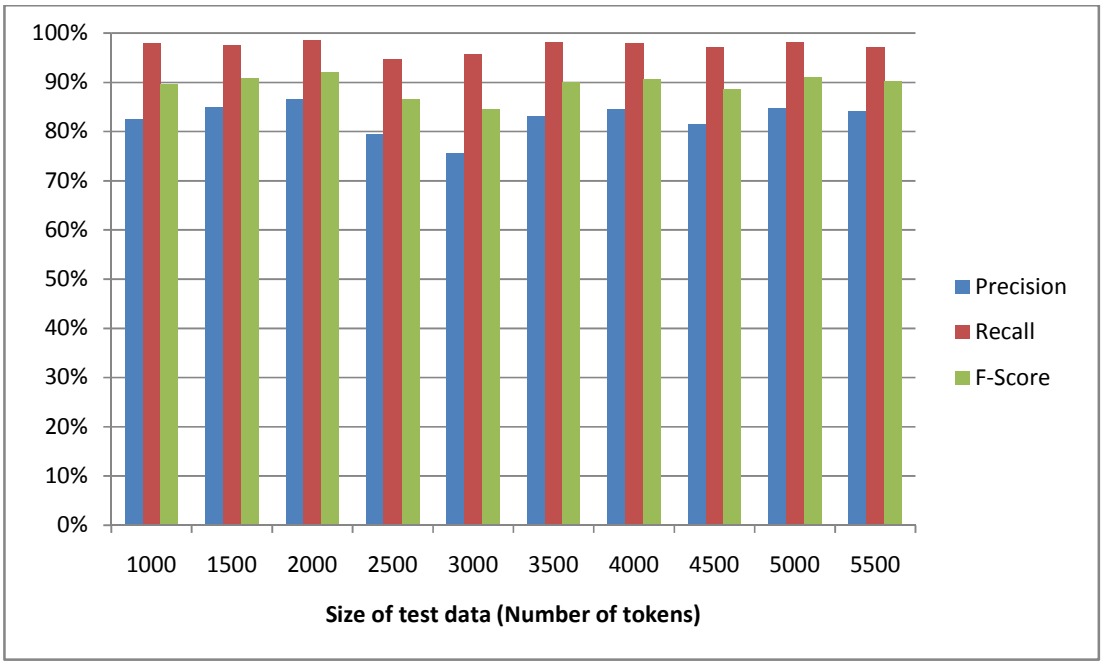


Figure 6.2 Bar Diagram for Precision, Recall and F-Score for training size 15000 tokens.

6.4.5 Experiment No. 3(Training Size 29298 tokens)

The sample input for experiment No. 3 is presented in Appendix C.

Experiment No.	Size of Test data (in tokens)	Precision (%)	Recall (%)	F-Score (%)
1	1000	89.51	98.99	94.01
2	1500	88.96	98.72	93.59
3	2000	90.76	99.29	94.83
4	2500	81.51	97.61	88.84
5	3000	86.15	98.56	91.94
6	3500	85.57	98.57	91.61
7	4000	88.57	98.77	93.39
8	4500	86.47	98.41	92.06
9	5000	85.85	98.66	91.81
10	5500	85.19	97.80	91.06

Table 6.5 Experiment No. 3(Training Size 29298 tokens)

6.4.6 Bar Diagram of Experiment No. 3

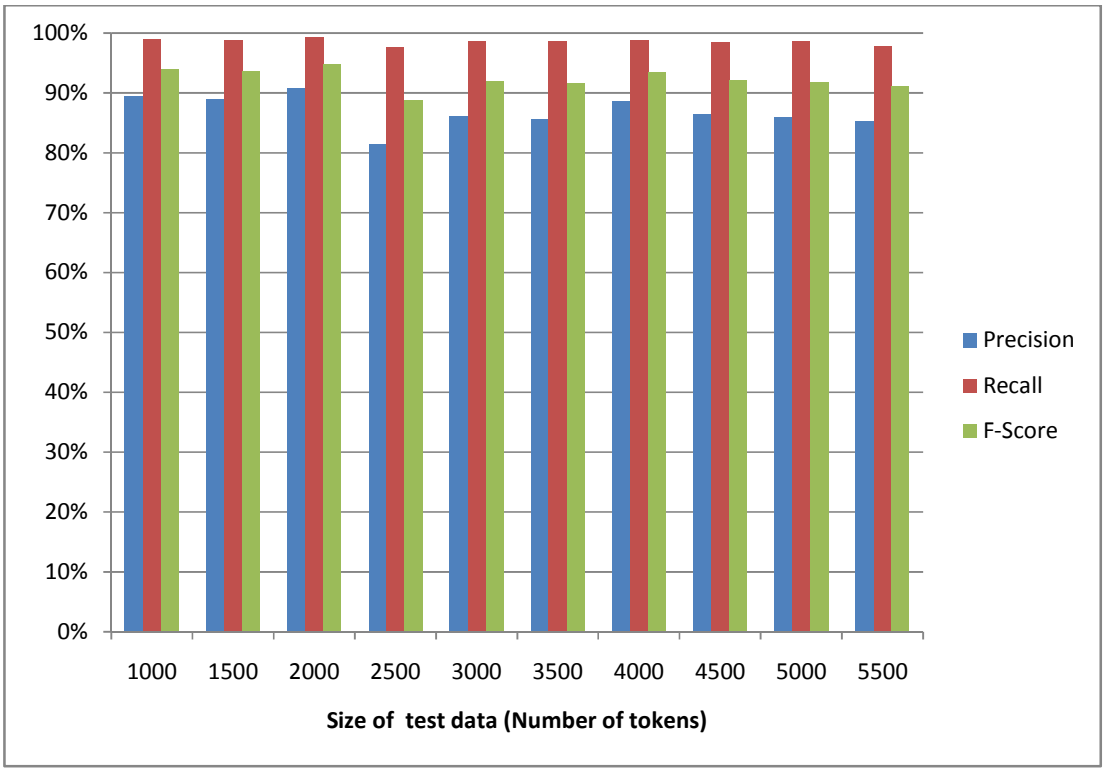


Figure 6.3 Bar Diagram for Precision, Recall and F-Score for training size 29298 tokens.

6.4.7 The Precision, Recall and F-Score for different training data size

Training Data Size (in tokens)	Precision	Recall	F-Score
5000	65.93%	80.42%	72.44%
15000	82.66%	97.27%	89.36%
29298	86.85%	98.53%	92.31%

Table 6.6 Overall Precision, Recall and F-Score for different training data size

The corresponding line curve is presented in figure.6.4

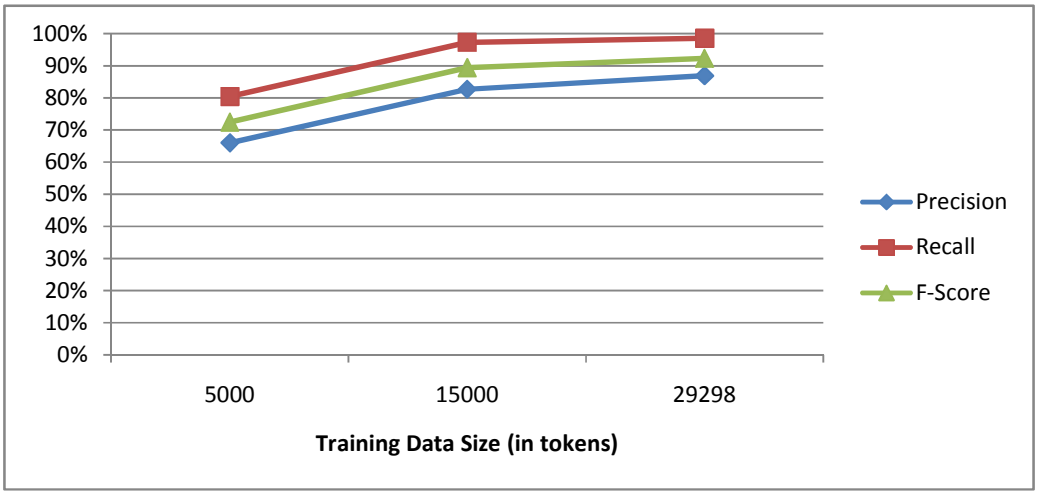


Figure 6.4 Overall Precision, Recall and F-Score for different training data size.

The line graph shows that the gradual increment in Precision, Recall and F-Score for the large size of the training data

CHAPTER 7

CONCLUSION AND FURTHER RECOMMENDATIONS

7.1 Conclusion

In this work, the method for extracting named entities from data of various domains had been presented which is a system useful in the identification and classification of names. The work for Nepali NER is very complex due to the nature of Nepali language which is order free and the lack of research work in Nepali text. There are no any corpus exists for Named Entity so it is difficult and tedious work to create such corpus. For this work the NE corpus is created manually.

The scalability issues associated with solving the named entity recognition problem using support vector machines and high-dimensional input. The usability of the machine learning environment and the available tools are also assessed. Training an SVM to classify multiple independent classes at once is a complex optimization problem with many variables.

The study has gone through the empirical analysis of the performance of the recognizer for morphologically rich and order free language like a Nepali. Here, during the development of the model, the impact of the size of the training data and test data on the performance was observed. The experiment was done for three different sizes of the train data; it is shown that the performance of the method depends on the size of train data. Here, in this work, the Recall, Precision and F-score for experiment no. 1 is 65.93%, 80.42%, 72.44%, for experiment no. 2 is 82.66%, 97.27%, 89.36% and for experiment no. 3 is 86.85%, 98.53%, and 92.31% respectively.

7.2 Further Recommendations

For future research, this work could be used on natural language processing using machine learning in several directions including the extension of the database solution to support the recommended service-oriented architecture, multi-word named entity recognition, more features including part-of-speech tags, and unsupervised learning.

One of the drawbacks of the SVM based work is the speed. It is found that the system to be slow in training phase, so, to increase the performance of the system, the empirical analysis to find the optimal set of features may be the future work which may concentrate on speed optimization of SVM based NE recognizer.

References

1. M.S Bindu, S. M. Idicula.: *Named Entity Recognizer employing Multiclass Support Vector Machines for the Development of Question Answering Systems International Journal of Computer Applications (0975 – 8887) Volume 25– No.10, July 2011.*
2. E. Asif and B. Sivaji: *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).*
3. Y.C. Wu, T.K. Fan, Y.L., Yen, S.: *Extracting Named Entities using Support Vector Machines. In: Springer- Verlag (2006).*
4. E. Asif and B. Sivaji: *Named Entity Recognition Using Appropriate Unlabeled Data, Post-processing and Voting Informatica 34,pp. 55–76 (2010).*
5. T. Joachims, *"Making large-scale support vector machine learning practical," pp. 169–184, (1999).*
6. R. Grishman, J. G. Carbonell, Ed. Frascati:. *"Information Extraction: Techniques and Challenges," in Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology Springer, 1997, pp. 10-26*
7. S. Coates-Stephens:. *"The Analysis and Acquisition of Proper Names for Robust Text Understanding," in Dept. of Computer Science. London: City University, 1992.*
8. N. Chinchor:. "*MUC-6 Named Entity Task Definition (Version 2.1)," presented at the 6th Message Understanding Conference, Columbia, Maryland, 1995.*
9. A. Borthwick,: *Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University (1999).*
10. A. Borthwick, , Sterling, J., Agichtein, E., Grishman, R.: *NYU: Description of the MENE Named Entity System as Used in MUC-7. In: MUC-7, Fairfax (1998).*
11. G. Zhou Su, J.: *Named Entity Recognition using an HMM-based Chunk Tagger. In: Proceedings of ACL, Philadelphia pp.473–480 (2002).*
12. 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).*

13. 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.*
14. T. Joachims.: *Multi-Class Support Vector Machine,* Cornell University, 2008.
15. Y.C. Wu, T.K. Fan, Y.L., Yen, S.: *Extracting Named Entities using Support Vector Machines. In: Springer- Verlag (2006).*
16. S Sekine.: *Description of the Japanese NE System used for MET-2. In: MUC-7, Fairfax, Virginia (1998).*
17. E. Riloff and R. Jones: *Learning dictionaries for information extraction by multi-level bootstrapping. In Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference, pages 474–479 (1999).*
18. E. Asif and B. Sivaji: *Named Entity Recognition using Support Vector Machine: A Language Independent Approach International Journal of Electrical and Electronics Engineering 4:2 (2010).*
19. C. Cortes V. Vladimir *Support-Vector Networks Machine Learning, 20, pp.273-297 AT&T Bell Labs., Holmdel, NJ 07733, USA (1995).*
20. C. Nello and S. T. John, An Introduction to Support Vector Machines and Other Kernel- based Learning Methods, Cambridge University Press pp. 126(2002).
21. D. Jurafsky, Martin, J. H. Speech and Language Processing: An Introduction to Speech Recognition Natural Language Processing and Computational Linguistic, (2006).
22. H. Yamada, , Kudo, T., Matsumoto, Y.: *Japanese Named Entity Extraction using Support Vector Machine. In Transactions of IPSJ 43, pp.44–53 (2001).*
23. T. Michael and E. Riloff.: *A bootstrapping method for learning semantic lexicons using ex- traction pattern contexts. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pages 214–221(2002).*
24. C. Alessandro, L.Danilo, and V.Paola.: *Automatic semantic tagging of unknown proper names. In Proceedings of the 17th International Conference on Computational Linguistics, pages 286–292(1998).*
25. H. Leong Chieu, Hwee Tou Ng *Named entity recognition: a maximum entropy approach using global information (2002).*

26. N. Joel.: *Learning NER from Wikipedia 2008*
27. F. Erik, T. Kim Sang.: *Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. In Proceedings of the 6th Conference on Natural Language Learning, pages 1–4(2002).*
28. F. Erik T. Kim Sang and D. M. Fien.: *Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proceedings of the 7th Conference on Natural Language Learning, pages 142–147(2003).*
29. T. Kudo, Matsumoto, Y.: *Chunking with Support Vector Machines. In: Proceedings of NAACL pp. 192–199 (2001).*
30. P. Mitchell Marcus, S. Batrice, Ma. Mary Ann rcinkiewicz.: *Building a large annotated corpus of English: the Penn Treebank (1993).*
31. K. Darvinder, G. Vishal.: *A survey of Named Entity Recognition in English and other Indian Languages. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 6, November 2010.*
32. S. Satoshi, G. Ralph, S. Hiroyuki: *A Decision Tree Method for Finding and Classifying Names in Japanese Texts.*
33. P.P. Talukdar, T. Brants, L. Mark, and P. Fernando.: *A context pattern induction method for named entity extraction. In Proceedings of the 10th Conference on Computational Natural Language learning, pages 141–148 (2006).*
34. T. Antonio, M. Rafael, and M. Monica.: *Named entity WordNet. In Proceedings of the 6th International Language Resources and Evaluation Conference (2008).*
35. M. Andrei, M. Marc, and C. Grover.: *Named entity recognition without gazetteers. In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics, pages 1–8, Bergen, Norway(1999).*

Appendix A

Sample Input for Experiment No. 1

१८माघ, पोखरा । मनाङ मर्स्याङ्दी क्लबका प्रशिक्षक कृष्ण थापाले पदबाट राजीनामा दिएका छन् । राष्ट्रिय लिगमा राम्रो प्रदर्शन गर्न नसकेको भन्दै थापाले नैतिकताका आधारमा राजीनामा दिएको घोषणा गरेका हुन् ।

पोखरामा बिहीबार पत्रकार सम्मेलन गरी थापाले जारी ए डिभिजन लिगमा यसअघि भएको एनसेल कप र सफल पोखरा कपमा हराएका टोलीहरुभन्दा पनि पछि परेपछि नैतिकताको आधारमा राजीनामा दिएको घोषणा गरे । राम्रो खेलेपनि गोल गर्न नसक्दा मनाङले राष्ट्रिय लिगमा साचेजस्तो सफलता हात पार्न नसकेकेा उनको भनाई छ । उनले मनाङमा हाल रहेका खेलाडीहरुकै भरमा आगामी प्रतियोगिता जित्न सम्भव नहुने समेत बताउनु भयो ।

राष्ट्रिय टोलीको पूर्व प्रशिक्षक थापा एक वर्षअघि नेपाली फुटबलका सर्बाधिक महंगा प्रशिक्षकका रुपमा मासिक ५० हजार तलबमा मनाङ गएका थिए ।

उनको प्रशिक्षणमा मनाङले एनसेल कप र सफल पोखरा कप जितेको थियो । तर, आहा गोल्डकपको पहिलो खेलबाटै बाहिएको मनाङ ए डिभिजन लिगमा पाँचौं स्थानमा छ ।

१८ माघ, काठमाण्डौ । कार्यकाल नसकिँदै फिर्ता बोलाइएका नेपालका लागि चिनियाँ राजदूत याङ होउलान र प्रधानसेनापति गौरव शम्शेर जबरावीच विदाई भेटवार्ता भएको छ । नेपाली सेनको जंगी अड्डामा विहीवार भएको भेटमा उहाँरुवीच नेपाली सेनालाई चीनले उपलब्ध गराउँदै आएको सहयोगका वारेमा छलफल भएकेा थियो ।

भेटमा प्रधानसेनापति जबराले नेपाली सेनालाई चीनले उपलब्ध गराउँदै आएको सहयोग महत्वपूर्ण रहेको भन्दै कृतज्ञता ब्यक्त गर्दै आगामी दिनमा पनि सहयोगलाई निरन्तरता दिन आग्रह गर्नु भएको थियो । होउलानले नेपाली सेनाले देश विकासमा पुऱ्याएको योगदान महत्वपूर्ण रहेको भन्दै चीनले आगामी दिनमा पनि सहयोग गर्ने प्रतिबद्धता जनाउनु भएको थियो ।

पटक-पटक हात्ती नियन्त्रणका लागि प्रशासन गुहार्दै आएका स्थानीय हात्ती पीडितले हरिपुर-३ की ५५ वर्षीया मौलीदेवी मुखियाको हिजो राति जङ्गली हात्तीले मारेपछि आक्रोशित भएर सडक आन्दोलनमा ओर्लिएका हुन् ।

Appendix B

Sample Input for Experiment No. 2

प्रकाश तिमल्सिना, काठमाडौं, माघ १६- मधेसी जनअधिकार फोरम (लोकतान्त्रिक)का अध्यक्ष तथा उपप्रधानमन्त्री विजयकुमार गच्छदारले आफ्नो पार्टीले प्रतिबद्धता गरेअनुसार १० हजार सर्वसाधारण सभामा ल्याउन नसक्ने जानकारी दिएका छन्।

सत्तापक्ष संघीय लोकतान्त्रिक गणतान्त्रिक गठबन्धनको नाममा बुधबार हुने आमसभामा गच्छदारले १० हजार सर्वसाधारण ल्याउन नसक्ने भन्दै पन्छिएको सहभागी स्रोतले बतायो। 'केटाहरु (गणेश लामा)ले १० हजार मान्छे ल्याउने जिम्मा हाम्रो पार्टीका भनेका रहेछन्, तर हामीले ल्याउन नसक्ने भयौं,' गच्छदारले बैठकमा भने, 'म र महासचिव (जीतेन्द्र देव) दुवै ब्यस्त भएकाले त्यत्रो संख्यामा मानिस उतार्न नसक्ने भयौं।' गठबन्धनको केही दिन अघिको बैठकमा केन्द्रीय सदस्य गणेश लामाले १० हजार मानिस आमसभामा उतार्ने प्रतिबद्धता जनाएका थिए। तर मंगलबारको बैठकमा अध्यक्ष गच्छदारले भने नसकिने जनाऊ दिएका हुन्।

फागुन, काठमाण्डौ । प्रधानन्यायाधीश खिलराज रेग्मीको नेतृत्वमा चुनावी सरकार बनाउने एकीकृत नेकपा माओवादीको प्रस्तावलाई नेपाली कांग्रेस र एमालेले अस्वीकार गर्ने भएका छन्।

सोमबार तीन दलको वैठकमा खिलराजलाई मान्न सैद्धान्तिकरुपमा सहमत देखिएका कांग्रेस एमालेले मंगलबार केन्द्रीय समितिको वैठक बोलाई त्यसलाई मान्न नसकिने निर्णय गरेका छन् । खिलराजलाई मान्ने/नमान्नेबारे दुवै दलमा तीब्र बिबाद देखिए पनि प्रस्ताव अस्वीकार गर्नेको बहुमत देखिएको छ ।

एमालेको औपचारिक निर्णय

नेकपा एमालेले प्रधानन्यायाधीशको नेतृत्वमा चुनावी सरकार गठन गर्ने एमाओवादीको प्रस्ताव अस्वीकार गरेको छ । मंगलबार बसेको एमाले केन्द्रीय कमिटीको बैठकले चुनावी सरकारको नेतृत्व राजनीतिक दलभित्रैबाट खोज्नुपर्नेमा जोड दिएको छ । बैकल्पिक केन्द्रीय सदस्य ठाकुर गैरेले अनलाइनखबरलाई दिनुभएको जानकारी अनुसार बिभिन्न विकल्पमा छलफल हुँदा दलकै नेतृत्वमा चुनावी सरकार गठन हुनुपर्ने निर्णय सर्वसम्मत रुपमा भएको हो ।

बैठकमा नेता घनश्याम भुषालले एमाओवादीले भारतको एजेण्डा ल्याएको भन्दै गम्भीर बन्न शीर्ष नेताहरुको ध्यानाकर्षण गराउनुभएको थियो । 'एकीकृत माओवादीले अरुलाई चलाएको भन्ने गरिन्छ, तर वास्तवमा एकीकृत माओवादीलाई भारतले चलाएको छ' बैठकमा भुषालले भन्नुभयो-'भारतको एजेण्डा बोकेर हिँडेकाले राष्ट्रिय स्वाधिनता र सार्वभौमसत्तासँग ख्याल गरेर जान आवश्यक छ ।'

पन्त प्रधानन्यायाधीशको पक्षमा

बैठकमा नेता रघुजी पन्तले बहालवाला प्रधानन्यायाधीशकै नेतृत्वमा भएपनि चुनावी सरकार गठन गरेर देशलाई निकास दिनुपर्ने धारणा राख्नुभएको थियो । तर अरु सदस्यहरुले खराबमध्ये पनि खराब विकल्पको रुपमा पूर्व प्रधानन्यायाधीश वा स्वतन्त्रको नेतृत्व मान्न सकिने तर बहालवाला प्रधानन्यायाधीश कुनै हालतमा स्वीकार्न नसकिने अडान राखेका थिए । बैको कुरा सुनिसकेपछि पार्टी अध्यक्ष झलनाथ खनालले केन्द्रीय कमिटीको निर्णय अनुसार नै संयुक्त बैठकमा पेश हुने जानकारी गराउनुभयो । यसअघि सोमबार साँझ शितल निवासमा भएको बैठकमा एमालेले पूर्व प्रधानन्यायाधीशको नेतृत्व मान्न सकिने जनाउ दिएको थियो ।

कांग्रेसमा पनि नमान्नेकै बहुमत

मंगलबार बसेको कांग्रेस केन्द्रीय कार्यसमितिको बैठकमा बहुमत सदस्यहरुले प्रधानन्यायाधीशको नेतृत्वमा सरकार गठन गर्न नहुने धारणा राखेका छन् । सुशील कोइरालाको विकल्पमा जान सकिने तर, दलभित्रैबाट समाधान खोजिनुपर्ने अधिकांश कांग्रेस नेताहरुको धारणा रहेको केन्द्रीय सदस्य मीना सुब्बाले अनलाइनखबरलाई बताउनुभयो । 'दलको नेतृत्वमा सरकार गठन भएन भने बरु आन्दोलनमा जानुपर्छ तर प्रधानन्यायाधीशको नेतृत्व स्वीकार्नु हुँदैन भन्ने अधिकांश नेताहरुले बैठकमा धारणा राख्नुभएको छ ।' सुब्बाले भन्नुभयो- 'नेताहरुले बोल्ने क्रम अझै नसकिएकाले बुधबार दिउँसो १ बजे फेरि बैठक बस्दैछ ।'

कांग्रेस स्रोतका अनुसार मंगलबार बोल्ने २८ नेतामध्ये बरिष्ठ नेता शेरबहादुर देउवाले बाबुराम भट्टराईले राजीनामा दिन्छन् भने दलबाहिरको नेतृत्वमा सरकार बनाउने प्रस्तावलाई स्वीकारेर अघि जानुपर्ने बताउनुभएको छ । 'बाबुरामले राजीनामा दिए भने न्यायाधीशको नेतृत्वमा चुनावमा जानुपर्छ ।'- देउवाको भनाइ उधृत गर्दै स्रोतले भन्यो ।

त्यसैगरी नेतृ सुजाता कोइरालाले वीपी कोइरालाको मेलमिलाप नीतिबाट पाठ सिक्दै सबैले मेलमिलापका आधारमा अघि बढ्नुपर्ने धारणा राख्नुभयो । 'पार्टीहरुले एकलौटीमात्रै गर्नुहुँदैन, आन्दोलन गर्नुभन्दा मेलमिलापका आधारमा अघि बढ्नुपर्छ । वीपीले आफूमाथि आठवटा मुद्दा लागेको अवस्थामा पनि मेलमिलाप गर्नुभएको थियो ।'- सुजाताको भनाइ उधृत गर्दै स्रोतले अनलाइनखबरसँग भन्यो ।

देउवा र सुजाताले सहमतिका लागि लचिलो कुरा गरे पनि अर्का नेता डा.मनमोहन भट्टराई भने निकै कडारुपमा प्रस्तुत हुनुभएको थियो । 'तत्कालीन राजा महेन्द्रले ०१४ सालमा यस्तै प्रस्ताव ल्याएका थिए, अहिले प्रचण्डले त्यस्तै गर्न खोज्दैछन्, यो सत्ता लम्ब्याउने खेलमात्रै हो, हामीले आन्दोलनलाई नै निरन्तरता दिनुपर्छ ।'-भट्टराईको भनाइ थियो । बैठकमा बोल्ने क्रममा देउवा निकट भनेर चिनिने पूर्णबहादुर खड्का, डीना उपाध्याय लगायतले पनि माओवादी प्रस्ताव मान्न नसकिने अडान राखेका थिए ।

फागुन, काठमाडौं । एसियाली फुटबल महासङ्घको कार्यक्रमअनुसार अखिल नेपाल फुटबल सङ्घको आयोजनामा आगामी फागुन १९ देखि २३ सम्म सञ्चालन हुने एएफसी च्यालेन्ज २०१४ छनौटअन्तर्गत समूह डिका लागि एन्फाले खेलाडी छनोट गरेको छ ।

काठमाडौँको दशरथ रङ्गशालामा सञ्चालन हुने प्रतियोगितामा नेपाललगायत बङ्गलादेश, प्यालेस्टाइन र नर्दन मारियाना आइल्याण्डका राष्ट्रिय फुटबल टिमको सहभागीता रहने एन्फाले जारी गरेको प्रेस विज्ञप्तिमा उल्लेख छ ।

सो प्रतियोगिताको लागि किरण चेम्जोङ, विराज महर्जन, विजय धिमाल, सन्दिप राई, विक्रम लामा, विजय गुरुङ, अनिल ओझा, सन्तोष साहुखल छनोट भएका छन् ।
त्यसैगरी छनोटमा परेका अन्य खेलाडीहरूमा राजेन्द्र रावल, सविन्द्र श्रेष्ठ, चेतन घिमिरे, दिपक राई, विशाल राई (ए), जगजित श्रेष्ठ, निराजन खड्का, अनिल गुरुङ, रविन श्रेष्ठ, रितेश थापा, विकेश कुथु, भोला सिलवाल, जुमानु राई, जितेन्द्र कार्की, राजु तामाङ, भरत खवास, नवयुग श्रेष्ठ, सागर थापा, अमर डङ्गोल र रोहित चन्द छानिएका छन् ।

Appendix C

Sample Input for Experiment No. 3

सुन्दरीजल (काठमाडौं), पुस १६ - कांग्रेस सभापति सुशील कोइरालाले आफू प्रधानमन्त्री पदको लोभी नभएको बताएका छन् । 'मलाई प्रधानमन्त्रीको पद चाहिँदैन,' प्रधानमन्त्री बाबुराम भट्टराईले कुर्सीको लोभ गरेको संकेत गर्दै भने, 'माओवादीले छक्याए भने ठानेको होला, यो उसका लागि नै घातक हो ।'

वीपी कोइरालाको ३७ औं राष्ट्रिय एकता तथा मेलमिलाप नीतिको अवसरमा सुन्दरीजलमा सोमबार भएको कार्यक्रममा कोइरालाले एमाओवादी अध्यक्ष पुष्पकमल दाहालले बैठक पिच्छे बोली फेरे पनि त्यो स्वयं उसैका लागि घातक भएको बताए । माओवादीले अधिनायकबाद लाद्न खोजेको भन्दै उनले त्यसको जनताले सशक्त प्रतिवाद गर्ने बताए ।

'०७ सालमा राणाले, २०१७ सालमा राजा महेन्द्रले प्रजातन्त्र खोसे,' कोइरालाले इतिहास कोट्याए, 'ज्ञानेन्द्रले खोजे आखिर उनीहरुको अवस्था के भयो ? अब माओवादी अबस्था के हुन्छ । बेलैमा सोचे हुन्छ ।' गणतन्त्रविरुद्ध कसैलै जान्छौं भन्नु दिवा स्वप्नबाहेक अरु केही नभएको उनले बताए ।

माओवादीले जति बोलि फेर्छ त्यो स्वयं उसैलाई घाटा हुने उनले बताए । 'बोली फेर्दा हामीले राम्रा हुन्छौं भनेर माओवादीले सोचेका होलान् त्यो भ्रममात्र हो, नेपाली जनताले बुझिसकेका छन् अब कति दिन छक्याउँछन्,' उनले भने ।

राप्रपा अध्यक्ष पशुपति शमशेर राणाले माओवादीको अधिनायकवाद नटिक्ने भन्दै त्यसको प्रतिवाद गरिने बताए । 'सम्झौता गर्ने कार्यान्वयन नगर्ने यो कहाँको प्रजातन्त्र हो ?' उनको प्रश्न थियो । राष्ट्रियता स्वाधिनता खतरामा परेको भन्दै अब सबै पार्टीले सोच्नु पर्ने उनको भनाइ थियो ।

मुमारमखनाल तीन विपरीत दिशामा मुख फर्काएर नेकपामाओवादी , नेकपाएमाले र मधेसीजनअधिकारफोरम ले अग्रगमनमा गएर घरजम गर्ने वाचा गरेका छन्। तर, तीनतिर फर्केर एउटै गाडी हाकिरहेका यी दलहरूको यात्राले गणितको भेक्टर बलको सिद्धान्त अनुसार तटस्थ हुन गई जहँको त्यही शून्य दूरी पार गर्ने छन् भन्ने बुझ्न अप्ठ्यारो छैन। निश्चय नै माओवादीलाई लतारएिर भए पनि अग्रगमनसम्म पुग्नैपर्ने बाध्यता छ। किनभने, १० वर्षो

जनयुद्धमा बगेको रगतमा टेकेर १ लाख २५ हजार रुपियाको पलङमा चैनसग सुत्न प्रचण्ड लाई अझै केही वर्षगाह्रै होला। तर, ठूला महलभित्र रहेका त्यसभन्दा कैयन गुना महगा पलङमा सुते पनि फाटेका चप्पल लगाउने वर्गको राजनीति गर्न नछाडेका एमालेहरूलाई अग्रगमनसम्म पुग्न कुनै हतारो र चटारो छैन। अर्कोतिर, पद र पैसाको विनियोजनमा एक लाल तलमाथि हुनेबित्तिकै भाड मे जाए मधेसी अधिकार भन्ने मूल नारामा एकत्रित फोरम कुन अग्रगमनको यात्रामा हिडेको छ भन्ने अज्ञात छ। माओवादीको हतारो अग्रगमन, एमालेको लतारो अग्रगमन र फोरमको अज्ञात अग्रगमनको घरजमबाट समुन्नत नयँ नेपाल को शिशु जन्मने आशामा नेपाली जनता त्यसको सालनाल काट्ने औजारसहित बसेका छन्। सामाजिक गतिशीलताको यो विशिष्ट विन्दुमा स्थापित सामाजिक समरूपताले मान्छेलाई सामाजिक लिस्नोमा चढेर एकैचोटि स्वीट्जरल्यान्डको छत समातेर झुन्डिन सकिने बताएको छ। तर, पुजीवादी लोकतन्त्रभित्रबाट हर्ुकिएको सामाजिक गतिशीलताले उसको मुख्य दिशा अर्थात् पातालतिर लैजान्छ भन्ने त्यसको सामान्य अवधारणा हो। र, आधुनिक पाताल भनेको पुराणको नर्क होइन, वर्तमान रुवान्डा, कङ्गो र जिम्बाबे हो भन्ने बुझ्न पनि उत्तिकै आवश्यक छ। त्यसकारण इतिहासदेखि शब्द जञ्जालभित्र फस्दै आएका नेपाली जनताले वर्तमान सरकारको न्यूनतम साझा कार्यक्रमप्रति त्यति ठूलो अभिरुचि लिएको देखिएन। वर्तमान गठबन्धन सरकारले सार्वजनिक गरेको न्यूनतम साझा कार्यक्रमप्रति उल्लेख्य चर्चासमेत नहुनु त्यसप्रतिको उदासीनता र अविश्वसनीयताको बलियो प्रमाण हो। अझ थप रोचक त के भने सहमत भइसकेको नीति तथा कार्यक्रमलाई लात्ताले पन्छाएर एमालेले मर्यादाको छिर्के दाउ हान्यो। उसको त्यो दाउ र मालदार मन्त्रालयमा लुछाचुडीको कुरूप कास्िटङ देखिसकेका नेपालीले यो नया त्रिशङ्कु फिल्म पनि पर्ूण्ारूपमा फ्लप हुन्छ भन्ने बुझ्न कत्ति समय लगाएनन्। भारतीय फिल्मका मसलाहरूलाई हुबहु नक्कल गरेर हिट नेपाली फिल्म बनाउन कस्िसने नेपाली सिने जगत्का कँचा र लठेब्रा निर्देशकहरू जस्तै बनेका नया राजनीतिक हस्तीहरू अहिले चराको प्वँखजत्तिकै हलुका भएका छन्। लोकतन्त्रको बालुवामा मुन्टो घुसारे पनि उनीहरूलाई छलङ्ग देख्न सकिने लोकतान्त्रिक संस्कृतिले जनतालाई दिएको फाइदा यत्ति हो। अब जाऔ, माओवादी, फोरम र एमालेको साझा प्रतिबद्धतातिर। सामान्यतया यस्ता प्रतिबद्धताहरू हरेक नया सरकार बन्दा औपचारकिताका लागि सार्वजनिक गरन्िछन्। तर, वर्तमान नीति तथा कार्यक्रमको साझा सहमतिलाई सामान्य अर्थमा हेरनु भनेको १० वर्षो जनयुद्ध र १९ दिने जनआन्दोलनको अपमान हो। अझ भनौ, महान् अग्रगामी उद्देश्य बोकेका दर्ुइ भिन्न चरत्रिका आन्दोलन र त्यसको नेतृत्वको लाचारीलाई र्समर्थन दिनु हो। जनयुद्ध र

जनआन्दोलनको उपलब्धिमाथि ठिङ्ग उभिएर यथास्थितिसामु आत्मर्समर्पण गर्न तम्सिएको वर्तमान सरकारको चरित्रिलाई हर्ेदै उनीहरूको प्रतिबद्धताप्रति चोर औला ठड्याउन जरुरी छ।बाहिरबाट हर्ेदाको अवस्थात्रिशङ्कु सरकारको प्रतिबद्धताको रूप उटको जस्तो छ। अर्थात्, व्यष्टिमा राम्रो र समष्टिमा कुरूप। राष्ट्रियता, राष्ट्रिय एकता र राष्ट्रिय हितको सम्र्बद्धन यसको पहिलो अध्याय हो। सम्भवतः ०१७ सालपछिका सबै सरकारहरूले यस्तो प्रतिबद्धता व्यक्त गरेका छन्। सायद यो सरकारलाई पनि यो रेडिमेड अध्याय राजा महेन्द्र कै पालाको वाक्यांशमा दर्ुइचार शब्द थप्ने वा झिक्ने झन्झट मात्र गर्नुपरेको मात्र होला। अर्थात्, सङ्घीय लोकतान्त्रिक गाणतन्त्रको कार्यान्वयन र राज्यको पुनःसंरचनाभित्र खँदिएका बुदाहरूले पुराना सरकारहरूकै निरन्तरतालाई टेवा दिइरहेका छन्। जनआन्दोलनको सफलतालगत्तै पुनःस्थापना भएको चटके अन्तरमि विधायिकासंसद्ले जस्तै यो सरकारले पनि केही चटके प्रावधानहरू राखेको छ। राजाको सम्पत्तिको छानबिन र खोजी, सङ्घीय लोकतान्त्रिक गणतन्त्रसग बाझिने ऐनकानुनको खारेजी, पर्ूवसरकारले गरेका सहमतिको कार्यान्वयन, मानव अधिकारको सुनिश्चितता, सेना, पुलिस तथा निजामती प्रशासनलाई राजनीतिबाट स्वतन्त्र र प्रचलित ऐनकानुन र नियमावलीका आधारमा प्रयोग, स्थानीय निकायको अन्तरमि व्यवस्थापन, राज्यको समावेशीकरण र पुनःसंरचना आदिजस्ता पक्षहरू ख अध्यायमा समेटिएका छन्। दिगो शान्ति र सुरक्षाको प्रत्याभूतिको अध्यायले पनि पुरानै अनुहारलाई थोरै शृङ्गारपटार मात्र गरेको छ। दिगो शान्तिको काममा सधै लिप्त भएको प्रतिबद्धता काङ्ग्रेसको पनि प्रतिबद्धता थियो। उसको भन्दा खासै ठूलो भिन्नता नभएको यो अध्यायले शान्तिप्रक्रियालाई तार्किक निष्कर्षा पुर्याउने, सेना समायोजन गर्ने, हतियार व्यवस्थापन गर्ने, सुरक्षा आयोग बनाउने, शान्तिका लागि विभिन्न आयोगलाई एउटा आयोगमार्फ समन्वयीकरण गर्ने, दण्डहीनता र अराजकताको अन्त्य गर्ने कुरा लिएर आएको छ। त्यस्तै, अध्याय घले तात्कालिक राहत र पुनःनिर्माणको पोको बोकेर ल्याएको छ। यो पोकोभित्र जनयुद्ध, जनआन्दोलन र मधेसी आन्दोलनका घाउहरू पर्ुर्ने औषधिहरू छन्। विस्थापितहरूलाई क्षतिपर्ूर्ति, पुनःस्थापना र राहतका प्याकेजहरू पनि छन्। त्यस्तै जनयुद्धका घाइतेहरूको उपचार, क्षतिपर्ूर्ति, द्वन्द्वमा नष्ट भएका भौतिक संरचनाको पुनःनिर्माण, महगी नियन्त्रण, पेट्रोलियम पदार्थलगायत दैनिक उपभोग्य वस्तुहरूको सहज आपर्ूर्ति, गरबि जनताका लागि सुपथ मूल्यमा अत्यावश्यक वस्तुको वितरण आदि प्रावधानहरू पनि यस खण्डमा समावेश छन्। आर्थिक सामाजिक रूपान्तरणको अर्को अध्यायले नया नेपालको सपना बोकेको छ। स्वाभाविक पनि हो, दह्रो खुट्टाले नटेक्ने हो भने बाकी शरीर ठडिन सक्तैन। त्यसले साझा कार्यक्रमको यो खण्ड संयुक्त सरकारको खुट्टो हो। तर, यसको

रूप पनि सग्लो देखिन्न। सबै आर्थिक क्षेत्रहरू र सामाजिक रूपान्तरणको कुनै पनि कुनासम्म नछोडेको यो खण्ड नै सरकारको मुख्य नीतिगत आधार हो। भारतमा सन् १९४७ को स्वतन्त्रता समारोहमा महात्मागान्धी ले भाग लिएनन्। उनले त्यो स्वतन्त्रतालाई हाम्रो स्वतन्त्रता पनि भनेनन्। त्यसको मूल कारण थियो, देशमा भएको भनिएको विकासको ७५ प्रतिशत भाग दलाल पुजीपतिहरूको भागमा थियो। त्यसकारण सामाजिक, आर्थिक रूपान्तरणको यो पक्षलाई सामान्य सुधारबाट परविर्तन गर्न सकिन्छ भन्ने ठानेर माओवादी अगाडि बढेको छ भने भगवान् नै आए पनि उसलाई दर्ुघटनाबाट रोक्न सक्ने छैनन्। भित्रै छिर्दाको दृश्यसंयुक्त सरकारको साझा कार्यक्रमभित्र छिरेर हर्ेर्ने हो भने देशको अराजक सडकजस्तै हरेक बुदामा ढुङ्गामुढा तेस्याइएका छन्। कुनै बुदामा टायर बालिएका छन् र कुनै बुदा त अनिश्चित कालसम्म बन्द छन्। राष्ट्रिय एकता, राष्ट्रियता र राष्ट्रिय हितको सर्म्बर्द्धनलाई आफ्नो साझा कार्यक्रमको टाउको बनाएको यो गठबन्धनले इतिहासको सबैभन्दा बढी वैदेशिक हस्तक्षेपको सामना गरेको छ। राष्ट्रिय स्वतन्त्रता र सार्वभौमसत्ताको रक्षाको कुरा गर्दा स्वतन्त्र ढङ्गबाट भारत बाहेक अर्को राष्ट्रमा भ्रर्ूर्रमण गर्न पनि आपत्तिको विषय बनिसकेको छ। विगतकै निरन्तरतामा थपिएका राष्ट्रियता बुदाहरूले केवल राजनीतिक व्यङ्ग्यको मात्र अर्थ बोकेका छन्। सङ्घीय लोकतान्त्रिक गणतन्त्रको कार्यान्वयन र राज्यको पुनःसंरचना संविधानको लेखनपछि मात्र पर्ूण्ाता पाउने विषयहरू हुन्। तर, यस्तो महत्त्वपर्ूण्ा अन्तरवस्तु बोकेको र संविधान लेखनका लागि निर्मित संविधानसभा बेरोजगार छ। स्थानीय निकायमा रहेको सत्ता शून्यतालाई कुन आधारमा परपिर्ूर्ति गर्ने हो भन्ने प्रश्नको जवाफ पनि अनिश्चित छ। विगतमा भएका सम्झौताहरूको पालना गर्ने भनिएको छ तर तिनको पृष्ठभूमि नै विवादित छ। राज्य पुनःसंरचनाको प्रश्नमा पनि त्यत्तिकै मतभिन्नताहरू कायम छन्, जसले संविधानसभाबाट संविधान निर्माणको प्रक्रियालाई नै अवरुद्ध पार्न सक्ने ताकत राख्छन्। शान्तिप्रक्रियालाई तार्किक निष्कर्षा लैजाने प्रसङ्गमा समग्र द्वन्द्वको समाधानको प्रश्न जोडिएर आइहाल्छ। द्वन्द्वका कारणहरूभित्र रहेका कुनै पनि समस्याको स्थायी निदानबिना नै दिगो शान्तिको सपना असफल हुनेछ। छ महिनाको अवधिभित्र सेना समायोजन र हतियार व्यवस्थापन गरसिकिने भनिएको छ। तर,नेपालीसेना , प्रहरीसङ्गठन , प्रतिपक्ष, अन्तर्रर्ाष्ट्रय सङ्घसंस्था र शक्तिलाई एउटै विन्दुमा ल्याउन सरकारले चढ्नुपर्ने उकालो धेरै ठाडो छ। स्वयम् सहमतिमा रहेका दलहरूको गन्तव्य विन्दु नै केन्द्रिकृत नभएको अवस्थामा यति धेरै शक्तिलाई साझा कार्यक्रमको म्यान्डेटमा ल्याएर सेना समायोजन एवम् हतियार व्यवस्थापन गर्न माओवादीले आफ्नो रूप र रङलाई समेत कुनकुन

रूपमा फर्ेर्नुपर्ने हो हाम्रोजस्तो सामन्ती लोकतन्त्र भएको ठाउमा आर्थिकसामाजिक रूपान्तरणका लागि जनमुखी र लोकतान्त्रिक अर्थव्यवस्थाको झन्डामुनि निजी क्षेत्रलाई देशै बेच्ने अधिकार दिने राजनीतिज्ञहरूको कमी हुदैन। त्यसकारण अहिले लोकतन्त्रका नाममा हुन थालेको सामाजिक रूपान्तरण भनेको बजारतन्त्र हो भन्ने बुझ्न पनि कठिन छैन। बजारतन्त्रमा सबैभन्दा बढी बदनाम बनाइने विषय भनेका जनता र लोकतन्त्र हुन्। ९३ प्रतिशत जनताले कृषि श्रम गरेर पनि भोकमरी पर्ने, छिमेकी राष्ट्रले सबै नदीनाला ओगटेर पनि कहिल्यै बिजुली ननिकाल्ने, उद्योगपतिले ब्याङ्क डुबाउने, यातायात जगत्ले सिन्डीकेट लागू गर्ने, सञ्चार क्षेत्रले आफ्नै गुणगान गाउने, औद्योगिक क्षेत्रले देशै डुबाउने आदिजस्ता परम्परागत विशेषताहरूमा एकाएक विजय पाउने फत्तुर योजनाको आन्तरकि पक्षले त्यसको विपरीत परण्ािाम दिन्छ। संविधानसभामा देखिएका समावेशी अनुहारहरूलाई देखाएर मात्र मुलुकको नया आर्थिक जग निर्माण सम्भव हुदैन। विदेशी लगानीमाथि नियन्त्रण, उनीहरूका नाफामूलक उद्योगमा हस्तक्षेप, सामन्तहरूको जग्गा अधि्रहण, सम्पर्ूण्ा जातीय, क्षेत्रीय, लैङ्गिक विभेदको अन्त्य, क्रान्तिकारी भूमिसुधारजस्ता कामहरू गर्ने हैसियत र अँट यो सरकारमा हुने कुरा होइन। किनभने, त्यस्तो आट गर्नेबित्तिकै यो सरकारको आयु पनि सिद्धिन्छ। त्यसकारण आवरणमा क्रान्तिकारीजस्तो देखिनुपर्ने र अर्न्तर्यमा यथास्िथतिबाट सामान्य सुधारतर्फपनि जान नसक्ने ट्राफिक जाममा सरकार फस्दैछ। त्यसैले जसलाई जस्तो सुन्ने बानी परेको छ, त्यस्तै सुनाइदिने बाध्यता यसलाई छ। त्यसैले गर्दा यो नीति तथा कार्यक्रम क्रान्तिकारी, दक्षिणपन्थी र यथास्िथतिवादी सबैलाई आफ्नो जस्तै लागेको हो। माथि जेसुकै लेखिएको भए तापनि भन्ने वाक्यांशले ऐनकानुनका सबै बुदाहरूलाई खारेज गरेजस्तै यो साझा कार्यक्रमको सञ्चालनको विधि र मान्यताले सबै साझा कार्यक्रमका बुदाहरूलाई निरस्त पारेको छ। किनभने, सबै बुदाहरूको अन्तिम कार्यान्वयन गर्ने अधिकार सरकारमा सहभागी दलहरूको अन्तिम सल्लाहमा सन्निहित छ।

Appendix D

Source code for training and testing

```
package thesis;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.HashMap;

  public class Main {
  public static ArrayList<String>  personList;
  public static  ArrayList<String> organizationList;
  public static  ArrayList<String> locationList;
  public static  ArrayList<String> miscList;

  public static ArrayList<String> personPrefixList;
  public static  ArrayList<String> middleNameList;
  public static  ArrayList<String> surNameList;
  public static  ArrayList<String> commonLocationList;
  public static ArrayList<String> actionWordList;
  public static  ArrayList<String> designationWordList;
  public static  ArrayList<String> organizationSuffixWordList;
  public static ArrayList<String> personNameList;
  public static  ArrayList<String> organizationNameList;
  public static  ArrayList<String> locationNameList;
  public static  ArrayList<String> miscNameList;
  public static ArrayList<String> nNEList;

  public static HashMap<String, Integer> NEhash=new HashMap<String, Integer>();
  public static String [][] features=new String[200000][20];
  public static ArrayList<String> featureVector=new ArrayList<String>();
//   String outputFile="src/tests/test10/inputWithTokens.dat";
//   String inputFileSVM ="src/tests/test10/input13.dat";

  String outputFile="src/train/inputWithTokens.dat";
  String inputFileSVM ="src/train/input13.dat";
  public static int feature_index=0;
  public static int npOfFeature=20;
  public static void main(String[] args)throws Exception{
    //Creation of Hash list of different named entity terms
   Main obj=new Main();
   obj.run();
   }
  void run() throws Exception
  {
    String filename="src/tests/test10/test10.txt";
     //String filename="src/train/train.txt";
```

```
//        for(int i=0;i<1000;i++){
//            firstWordFlag[i]='N';
//        }

       for (int i=0;i<200000;i++){
          for (int j=0;j<20;j++){
             features[i][j]="0";
          }
       }
```

Source code for extracting features

```
  public void ExtractFeature(String s) throws Exception
    {

       String []sentences = s.split("[?|]");

        for(int j=0;j<sentences.length;j++){
          sentences[j]=sentences[j].trim();
         String []tokenString=sentences[j].split("[ ,]");

          ArrayList<String> listOfWords=new ArrayList<String>();

           for(int i=0;i<tokenString.length;i++){
              tokenString[i]=tokenString[i].trim();
              if(!tokenString[i].isEmpty()){
                 listOfWords.add(tokenString[i]);
                  }
               }

          String tokens="";
          for(int i=0;i<listOfWords.size();i++){
             tokens=listOfWords.get(i);
             //Class of the token
             features[feature_index][0]=ThesisFunctions.assignClass(tokens).toString();

              //Token itself
             features[feature_index][1]=tokens;

             //First word feature
             if (i==0){
                features[feature_index][2]="1";
             }
             else{
                features[feature_index][2]="0";
             }
```

```
            // Word length feature (threshold: word_length >=3 )(i.e.,  word with length greater
or equal to 2 is more likely to be NE)
             if (tokens.length()>=3){
                features[feature_index][3]="1";
            }
            else{
                features[feature_index][3]="0";
            }
            //Digit feature
              if (ThesisFunctions.isDigit(tokens)){
                features[feature_index][4]="1";
            }
            else{
                features[feature_index][4]="0";
            }

            //Four digit number feature
            if (ThesisFunctions.fourDigit(tokens)){
                features[feature_index][5]="1";
            }
             else{
                features[feature_index][5]="0";
            }

            //digit following percentage feature
            if (ThesisFunctions.digitPercentage(tokens)){
                features[feature_index][6]="1";
            }
            else{
                features[feature_index][6]="0";
            }

            //Date feature
            if (ThesisFunctions.isDate(tokens)){
                features[feature_index][7]="1";
            }
            else{
            features[feature_index][7]="0";
            }
            // Gazeletter Lists Features
            if (ThesisFunctions.isPersonPrefix(tokens)){
                features[feature_index][8]="1";
            }
            else{
            features[feature_index][8]="0";
            }

             if (ThesisFunctions.isMiddleName(tokens)){
                features[feature_index][9]="1";
```

```
 }
 else{
 features[feature_index][9]="0";
 }

  if (ThesisFunctions.isSurName(tokens)){
    features[feature_index][10]="1";
 }
 else{
 features[feature_index][10]="0";
 }

  if (ThesisFunctions.isCommonLocationWord(tokens)){
    features[feature_index][11]="1";
 }
 else{
 features[feature_index][11]="0";
 }

if (ThesisFunctions.isActionVerb(tokens)){
    features[feature_index][12]="1";
 }
 else{
 features[feature_index][12]="0";
 }

 if (ThesisFunctions.isDesignationWord(tokens)){
    features[feature_index][13]="1";
 }
 else{
 features[feature_index][13]="0";
 }

 if (ThesisFunctions.isOrganizationSuffixWord(tokens)){
    features[feature_index][14]="1";
 }
 else{
 features[feature_index][14]="0";
 }

  if (ThesisFunctions.isPersonName(tokens)){
    features[feature_index][15]="1";
 }
 else{
 features[feature_index][15]="0";
 }

  if (ThesisFunctions.isOrganizationName(tokens)){
    features[feature_index][16]="1";
```

```
            }
            else{
            features[feature_index][16]="0";
            }

             if (ThesisFunctions.isLocationName(tokens)){
                features[feature_index][17]="1";
            }
            else{
            features[feature_index][17]="0";
            }

            if (ThesisFunctions.isMiscellaneous(tokens)){
                features[feature_index][18]="1";
            }
            else{
            features[feature_index][18]="0";
            }

            if (ThesisFunctions.isNotNE(tokens)){
                features[feature_index][19]="1";
            }
            else{
            features[feature_index][19]="0";
            }
            feature_index=feature_index+1;
            }
        }
    }
```

Functions for Features

```
package thesis;

import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;

public class ThesisFunctions {

   public static boolean isDigit(String s){

     if(s.matches("[०१२३४५६७८९]*")){
        //System.out.println("Nepali number="+s);
        return(true);
      }
     else if(s.matches(".*\\d.*")){
        //System.out.println("English number="+s);
```

```
            return(true);
          }
          else{
           return(false);
           }
        }

        public static boolean fourDigit(String s){

           if (s.length()==4){
              if (isDigit(s)){
                 // System.out.println("4 digit number="+s);
                 return(true);
              }
              else{
                 return(false);
              }
           }
           return(false);
        }

        public static boolean digitPercentage(String s){
              if(s.length()>1){
              String temp=s.substring(0, s.length()-1);
              String lastSymb=s.substring(s.length()-1);
//               System.out.println("sub="+temp);
//               System.out.println("last="+lastSymb);
//               System.out.println("str="+s);
              if (isDigit(temp)){
                 if (lastSymb.matches("%")){
                    //System.out.println("digit with percentage="+s);
                    return(true);
                 }
                 return(false);
              }
              else{
                 return(false);
              }
         }
              return(false);
         }

        public static boolean isDate(String s){
            if(s.length()>3){
              // For nepali date like १२बैसाख
              int i=0,index=0;
              int j=s.length(),jndex=0;
              boolean flag=true;
```

```
String dstr="";
String mstr="";

while(i<s.length()-1 && flag==true){
    if(isDigit(s.substring(i,i+1))){
      //System.out.println("substr="+s.substring(i,i+1));
      i=i+1;
      index=index+1;
    }
    else{
        if(s.length()==index+1){ //i.e., string only contating digits
            //System.out.println("no month="+s);
            return(false);
          }

        if(i==0){
           flag =false;
           break;
         }
        dstr=s.substring(0, index);
        mstr=s.substring(index,s.length());
//              System.out.println("dstr="+dstr);
//              System.out.println("mstr="+mstr);
//              System.out.println("str="+s);

          break;
    }
}
 // For nepali date like बैसाख१२
while(j>=0 && flag==false){
    if(isDigit(s.substring(j-1,j))){
      //System.out.println("substr="+s.substring(j-1,j));
      j=j-1;
      jndex=jndex+1;
    }
    else{

        if(s.length()==jndex+1){
            //System.out.println("no month="+s);
            return(false);
          }

        mstr=s.substring(0, j);
        dstr=s.substring(j,j+jndex);
//              System.out.println("mstr="+mstr);
//              System.out.println("dstr="+dstr);
//              System.out.println("str="+s);

       break;
```

```
            }
        }


        if(dstr.length()<=2){
            if
(mstr.matches("बैसाख")|mstr.matches("जेठ")|mstr.matches("असार")|mstr.matches("साउन")|m
str.matches("भाद्र")|mstr.matches("भदौ")|mstr.matches("असोज")|mstr.matches("कार्तिक")|mstr
.matches("मंसिर")|mstr.matches("पुस")|mstr.matches("माघ")|mstr.matches("फाल्गुन")|mstr.m
atches("चैत्र")){
            //System.out.println("nepali date= "+s);
                return(true);
            }
        }
    }

  //For nepali date like २०६४/१/३
    if(s.length()>6) {
        //System.out.println("iam inside="+s);
        String[] date=s.split("[/-]");
        if(date.length>3){
            return(false);
        }
        try{
        if((date[0].length()==4)){
            if(date[1].length()<=2 && date[1].length()>0  && date[2].length()<=2 &&
date[2].length()>0){
//                System.out.println("nepali date="+ s);
              return(true);
            }
        }
        }
        catch(Exception e){
            return(false);
        }
    }

    //For days of weeks

if(s.matches("आईतबार")|s.matches("सोमबार")|s.matches("मंगलबार")|s.matches("बुधबार")|s.m
atches("बिहिबार")|s.matches("शुक्रबार")|s.matches("शनिबार")) {
//          System.out.println("iam inside nepali week="+s);
              return(true);
    }
```

```
        return(false);
  }

    public static boolean isPersonPrefix(String s){
        if (Main.personPrefixList.contains(s)){
//            System.out.println("PersonPrefix: " + s);
           return (true);
        }
        return (false);
      }

    public static boolean isMiddleName(String s){

     if (Main.middleNameList.contains(s)){
//            System.out.println("MiddleName: " + s);
           return (true);
        }
        return (false);
      }

    public static boolean isSurName(String s){
        if (Main.surNameList.contains(s)){
//            System.out.println("SurName: " + s);
           return (true);
        }
        return (false);
      }

    public static boolean isCommonLocationWord(String s){
       if (Main.commonLocationList.contains(s)){
//            System.out.println("CommonLocationWord: " + s);
      return (true);
      }
        return (false);
      }

    public static boolean isActionVerb(String s){
      if (Main.actionWordList.contains(s)){
          // System.out.println("ActionVerb: " + s);
      return (true);
      }
        return (false);
      }

    public static boolean isDesignationWord(String s){
        if (Main.designationWordList.contains(s)){
//          System.out.println("DesignationWord: " + s);
      return (true);
      }
```

```
      return (false);
    }

  public static boolean isOrganizationSuffixWord(String s){
      if (Main.organizationSuffixWordList.contains(s)){
//        System.out.println("OrganizationSuffixWord: " + s);
    return (true);
    }
     return (false);
    }

  public static boolean isPersonName(String s){
     if (Main.personNameList.contains(s)){
//         System.out.println("Person: " + s);
    return (true);
    }
     return (false);
    }

  public static boolean isOrganizationName(String s){
      if (Main.organizationNameList.contains(s)){
//          System.out.println("Organization: " + s);
    return (true);
    }
     return (false);
    }

  public static boolean isLocationName(String s){
    if (Main.locationNameList.contains(s)){
//          System.out.println("Location: " + s);
    return (true);
    }
     return (false);
    }

  public static boolean isMiscellaneous(String s){
   if (Main.miscNameList.contains(s)){
//          System.out.println("Miscellaneous: " + s);
    return (true);
    }
     return (false);
    }

    public static boolean isNotNE(String s){
   if (Main.nNEList.contains(s)){
//          System.out.println("Not NE: " + s);
    return (true);
    }
     return (false);
```

```
    }

   public static ArrayList<String> HashConvert(String filename) {
    ArrayList<String> list=new ArrayList<String>();
    String s=null;
      try{
         FileReader fin=new FileReader(filename);
         BufferedReader bfin=new BufferedReader(fin);
         while((s=bfin.readLine())!=null){
            if(s.isEmpty()){
             continue;
          }
            if(s.charAt(0)==''){
            s=s.substring(1,s.length());
         }
            String []tokens=s.split("[ \t\n]");
            for(int i=0;i<tokens.length;i++){
               if(!tokens[i].trim().isEmpty())
               {list.add(tokens[i]);}
            }

         }

      }
      catch(Exception e){System.out.println(e);}
//       for (Iterator<String> it = list.iterator(); it.hasNext();) {
//           System.out.println(it.next());
//
//       }
    return list;
   }
```

Source code for Assigning Class

```
public static String assignClass(String token){
    if(isPersonClass(token)){
       return("1");
    }
    else if(isLocationClass(token)){
       return("2");
    }
    else if(isOrganizationClass(token)){
       return("3");
    }
    else if(isMiscellaneousClass(token)){
       return("4");
    }
    else{ //Other class
       return("5");
```

```
    }
 }

   public static boolean isPersonClass(String s){

      if (Main.personList.contains(s)){
//          System.out.println("class:(1)"+s);
     return (true);
     }
      return (false);
     }

   public static boolean isLocationClass(String s){

      if (Main.locationList.contains(s)){
//           System.out.println("class:(2)"+s);
     return (true);
     }
      return (false);
     }

   public static boolean isOrganizationClass(String s){

      if (Main.organizationList.contains(s)){
//           System.out.println("class:(3)"+s);
     return (true);
     }
      return (false);
     }

   public static boolean isMiscellaneousClass(String s){
      if (Main.miscList.contains(s)){
//          System.out.println("class:(4)"+s);
     return (true);
     }
      return (false);
     }

}
```

Source code for calculating Precision, Recall and F-Score

```
package Accuracy;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
```

```
public class accuracy {

    public static void main(String[] args) {
        calculateAccuracy();
        taggedOutput();
    }

    public static void calculateAccuracy() {
//        String inputFile = "src/train/input12.dat";
//        String outputFile = "src/train/output12.dat";
//        String accuracyFile = "src/train/accuracy12.dat";

        String inputFile = "src/tests/test10/input13.dat";
        String outputFile = "src/tests/test10/output13.dat";
        String accuracyFile = "src/tests/test10/accuracy13.dat";
        int[] inputArray = new int[200000];
        int[] outputArray = new int[200000];
        int totalNEInInput = 0;
        int totalNERetrived = 0;
        int totalNERetrivedCorrectly = 0;
        int totalEntityRetrivedCorrectly = 0;

        double tempTotalNEInInput;
        double tempTotalNERetrived;
        double tempTotalNERetrivedCorrectly;
        double tempTotalEntityRetrivedCorrectly;

        double precision;
        double recall;
        double fscore;
        double accuracy;

        double tempPrecision;
        double tempRecall;
        double tempFscore;
        double tempAccuracy;

        double[][] result = new double[4][5];

        String s;
        try {
            FileWriter foutAccuracy = new FileWriter(accuracyFile);

            FileReader fin = new FileReader(inputFile);
            BufferedReader bfin = new BufferedReader(fin);
            int inputIndex = 0, outputIndex = 0;
            while ((s = bfin.readLine()) != null) {
                String[] temp = s.split("[ ]");
                //System.out.println("input="+s);
```

```
            inputArray[inputIndex] = Integer.parseInt(temp[0]);
            inputIndex = inputIndex + 1;
        }
        fin.close();
        bfin.close();

        fin = new FileReader(outputFile);
        bfin = new BufferedReader(fin);
        while ((s = bfin.readLine()) != null) {
            String[] temp = s.split("[ ]");
//              System.out.println("output="+s);
            outputArray[outputIndex] = Integer.parseInt(temp[0]);
            outputIndex = outputIndex + 1;
        }

        for (int indx = 0; indx < inputIndex; indx++) {
            System.out.print(inputArray[indx] + " ");
        }
        System.out.println();

        for (int indx = 0; indx < outputIndex; indx++) {
            System.out.print(outputArray[indx] + " ");
        }
        System.out.println();

        if (inputIndex != outputIndex) {
            System.out.println("Input and output dimension do not match. Contact
Administrator.");
            return;
        }

        for (int i = 1; i <= 5; i++) {

            for (int indx = 0; indx < inputIndex; indx++) {
                if (inputArray[indx] == i) {
                    totalNEInInput = totalNEInInput + 1;
                }
            }
            for (int indx = 0; indx < outputIndex; indx++) {
                if (outputArray[indx] == i) {
                    totalNERetrived = totalNERetrived + 1;
                }
            }

            for (int indx = 0; indx < outputIndex; indx++) {
                if (outputArray[indx] == i && inputArray[indx] == i && inputArray[indx] ==
outputArray[indx]) {
                    totalNERetrivedCorrectly = totalNERetrivedCorrectly + 1;
                }
```

```
            }

            for (int indx = 0; indx < inputIndex; indx++) {
               if (inputArray[indx] == outputArray[indx]) {
                  totalEntityRetrivedCorrectly = totalEntityRetrivedCorrectly + 1;
               }
            }

            tempTotalNEInInput = totalNEInInput == 0 ? 0.1 : (double) totalNEInInput;
            tempTotalNERetrived = totalNERetrived == 0 ? 0.1 : (double) totalNERetrived;
            tempTotalNERetrivedCorrectly = totalNERetrivedCorrectly == 0 ? 0.1 : (double)
totalNERetrivedCorrectly;
            tempTotalEntityRetrivedCorrectly = totalEntityRetrivedCorrectly == 0 ? 0.1 :
(double) totalEntityRetrivedCorrectly;

            tempPrecision = (tempTotalNERetrivedCorrectly / tempTotalNERetrived) * 100;
            tempRecall = (tempTotalNERetrivedCorrectly / tempTotalNEInInput) * 100;
            tempFscore = (2 * tempPrecision * tempRecall) / (tempPrecision + tempRecall);
            tempAccuracy = (tempTotalEntityRetrivedCorrectly / (inputIndex - 1)) * 100;
//(inputIndex-1)=inputsize


            System.out.println("============== (Class = " + i +
")=========================");
            System.out.println("Total NE in input = " + tempTotalNEInInput);
            System.out.println("Total NE Retrived = " + tempTotalNERetrived);
            System.out.println("Total NE Retrived Correctly = " +
tempTotalNERetrivedCorrectly);
//          System.out.println("Total inputs = " + inputIndex);
//
System.out.println("================================================================
====");
            System.out.println("Precision=" + tempPrecision);
            System.out.println("Recall=" + tempRecall);
            System.out.println("F-Score=" + tempFscore);

            foutAccuracy.write("\n");
            foutAccuracy.write("============== (Class = " + i +
")=========================");
            foutAccuracy.write("\n");
            foutAccuracy.write("Total NE in input = " + totalNEInInput);
            foutAccuracy.write("\n");
            foutAccuracy.write("Total NE Retrived = " + totalNERetrived);
            foutAccuracy.write("\n");
            foutAccuracy.write("Total NE Retrived Correctly = " + totalNERetrivedCorrectly);
            foutAccuracy.write("\n");
//          foutAccuracy.write("Total inputs = " + inputIndex);
//          foutAccuracy.write("\n");
```

```
//
foutAccuracy.write("==========================================================
=====");


                result[0][i - 1] = tempPrecision;
                result[1][i - 1] = tempRecall;
                result[2][i - 1] = tempFscore;
                result[3][i - 1] = tempAccuracy;

                totalNEInInput = 0;
                totalNERetrived = 0;
                totalNERetrivedCorrectly = 0;
                totalEntityRetrivedCorrectly = 0;

            }

            precision = (result[0][0] + result[0][1] + result[0][2] + result[0][3] + result[0][4]) / 5;
            recall = (result[1][0] + result[1][1] + result[1][2] + result[1][3] + result[1][4]) / 5;
//          fscore=(result[2][0]+result[2][1]+result[2][2]+result[2][3]+result[2][4])/5;
            fscore = (2 * precision * recall) / (precision + recall);
            accuracy = (result[3][0] + result[3][1] + result[3][2] + result[3][3] + result[3][4]) / 5;


//            System.out.println("fscores="+result[2][0]+" "+result[2][1]+" " +result[2][2]+"
"+result[2][3]+" "+result[2][4]);
//            System.out.println("fscore="+(2 * precision * recall) / (precision + recall));


            System.out.println("--------------------Accuracy Calculation-------------------");
            System.out.println("Precision = " + precision);
            System.out.println("Recall = " + recall);
            System.out.println("F-Score = " + fscore);
            System.out.println("Accuracy = " + accuracy);

System.out.println("==========================================================
====");

            foutAccuracy.write("--------------------Accuracy Calculation-------------------");
            foutAccuracy.write("\n");
            foutAccuracy.write("Precision = " + precision);
            foutAccuracy.write("\n");
            foutAccuracy.write("Recall = " + recall);
            foutAccuracy.write("\n");
            foutAccuracy.write("F-Score = " + fscore);
            foutAccuracy.write("\n");
            foutAccuracy.write("Accuracy = " + accuracy);
            foutAccuracy.write("\n");
```

```
foutAccuracy.write("================================================
=====");
            foutAccuracy.write("\n");
            foutAccuracy.close();



        } catch (Exception e) {
            System.out.println(e);
        }

    }

    public static void taggedOutput() {
//        String inputFile = "src/train/inputWithTokens.dat";
//        String outputFile = "src/train/output.dat";
//        String taggedFile = "src/train/taggedOutput.dat";

        String inputFile = "src/tests/test10/inputWithTokens.dat";
        String outputFile = "src/tests/test10/output.dat";
        String taggedFile = "src/tests/test10/taggedOutput.dat";
        int fileSize = 0;

        ArrayList<String> inputList = new ArrayList<String>();
        ArrayList<String> outputList = new ArrayList<String>();
        String[][] taggedList = new String[200000][2];

        String s;
        try {
            //========================================
            FileReader fin = new FileReader(inputFile);
            BufferedReader bfin = new BufferedReader(fin);
            while ((s = bfin.readLine()) != null) {
                String[] tokens = s.split("[ ]");
                inputList.add(tokens[1]);
            }
            fin.close();
            bfin.close();

            //========================================
            fin = new FileReader(outputFile);
            bfin = new BufferedReader(fin);
            while ((s = bfin.readLine()) != null) {
                String[] tokens = s.split("[ ]");
                outputList.add(tokens[0]);
            }
        } catch (Exception e) {
            System.out.println(e);
```

```
        }

//        for(int i=0;i<inputList.size();i++){
//            System.out.println("token="+inputList.get(i));
//
//        }
//        for(int i=0;i<inputList.size();i++){
//            System.out.println("class="+outputList.get(i));
//
//        }

        for (int indx = 0; indx < inputList.size(); indx++) {
            if (outputList.get(indx).matches("1")) {
                taggedList[indx][0] = inputList.get(indx);
                taggedList[indx][1] = "PER";
            } else if (outputList.get(indx).matches("2")) {
                taggedList[indx][0] = inputList.get(indx);
                taggedList[indx][1] = "LOC";
            } else if (outputList.get(indx).matches("3")) {
                taggedList[indx][0] = inputList.get(indx);
                taggedList[indx][1] = "ORG";
            } else if (outputList.get(indx).matches("4")) {
                taggedList[indx][0] = inputList.get(indx);
                taggedList[indx][1] = "MISC";
            } else if (outputList.get(indx).matches("5")) {
                taggedList[indx][0] = inputList.get(indx);
                taggedList[indx][1] = "O";
            }
            fileSize = indx;
        }

        try {
            FileWriter fout = new FileWriter(taggedFile);
            for (int i = 0; i < fileSize; i++) {
                fout.write(taggedList[i][0]);
                fout.write(" ");
                fout.write(taggedList[i][1]);
                fout.write("\n");
            }
            fout.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```