Chapter 6

6 NAMED ENTITY RECOGNITION USING GAZETTEER LIST AND HIDDEN MARKOV MODEL(n-gram technique)

Named Entity Recognition (NER) is a process to find the Named Entities (NEs) in a text document and then categorize these NEs into different Named Entity classes such as Name of Location, Person, River, Organization etc. We are concentrating in performing NER in the Indian languages (IL) with special reference to Nepali. In this thesis we describe different techniques of NER(already discussed in Introduction and Literature reviewed section) and a brief introduction on Gazetteer method and Hidden Markov Model specially n-gram technique.

6.1 Gazetteer Method

Gazetteer Method is the creation of different gazetteer classes (list) for different Named Entities and then applies search operations to classify the names [Padmaja Sharma , Utpal Sharma, Jugal Kalita 2011]. This method needs two types of input rather collection of gazetteer, one for each named entity classes of interest and second for other class that give example of entities that we do not want to extract. To build such type of gazetteer classes we need a very large corpus. But it fails to resolve ambiguities in a given document. For example if in a document we have a name Ganga. That means when we prepare the gazetteer list then Ganga may be in the list of person name and in the list of river name. So there ambiguity exists. And it is difficult task for gazetteer method to correctly identify or tag the Ganga.

6.2 Advantages of Gazetteer Method

a) The Gazetteer method gives very fast result of NER.

b) The accuracy of Gazetteer method depends on completeness of the Gazetteer used.

c) Creating the gazetteer manually is effort-intensive, error-prone and subjective.

d) But the problem is how to automatically create a gazetteer with less effort, in less time and with high accuracy using a given document.

Disadvantages of Gazetteer Method

a) Ambiguity resolution is difficult.

b) Since the words are created repeatedly. So keeping a gazetteer list for these words up-to-date is challenging.

c) Without ambiguity resolution the precision is low.

6.3 Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical Markov Model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a Hidden Markov model the state is not directly visible, but the output dependent on the state is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'.[Sujan Kumar Saha, Sudeshna Sarkar, Pabitra Mitra 2008]

Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bio-informatics.

A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation are related through a Markov process rather than independent of each other.[Anastasia Rita Widiarti, and Phalita Nari Wastu 2009]

6.4 N-Gram Technique under HMM

N-grams have been widely investigated for a number of text processing and retrieval applications. An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus [W. Li and A. McCallum, Sept 2003].

An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram", size 3 is a "trigram". Larger sizes are sometimes referred to by the value of n, e.g., "four-gram", "five-gram", and so on.

An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of an $(n-1)$ order Markov model. N-gram models are now widely used in probability, communication theory, computational linguistics (for instance, statistical natural language processing), computational biology (for instance, biological sequence analysis), and data compression. The two core advantages of n-gram models (and algorithms that use them) are relative simplicity and the ability to scale up by simply increasing n. A model can be used to store more contexts with a well-understood space time trade-off, enabling small experiments to scale up very efficiently [M. N. Karthik, Moshe Davis 2010].

We in this paper concerned with word sequences and we referred up to five-gram (here n = 5 words) since few name of PERSONS could be up to five words (MEHBOOB HASAN BEN SERAJ MAZARBHUIYA), few Organizations are there whose name consists of five to six words.

Table 9: Different Approaches of NER and Their Accuracy

Author	Language	Approach	Words	Accuracy
[Ijaz, M., Hussain, S. 2007]	Telugu	CRF	13,425	91.95%
[Ijaz, M., Hussain, S. 2007]	Telugu	ME	--	50.00% aprx
[Asif Ekbal, Rajewanul Hague, Amitava Das, Venkateswarlu Poka and Sivaji Bandyopadhyay 2008]	Tamil	CRF	94,000	80.44%
[A. Goyal 2008]	Hindi	ME	25,000	81.52%
[A. Goyal 2008]	Hindi	CRF	--	60%
[Sujan Kumar Saha, Sudeshna Sarkar, Pabitra Mitra 2010]	Hindi	CRF	--	--
[A. Ekbal, R. Hague, and S. Bandyopadhyay 2013]	Bengali	CRF	150,000	90.7%
[A. Ekbal and S. Bandyopadhyay 2008]	Hindi	SVM	502,974	77.17%
[Kashif Riaz 2010]	Bengali	SVM	122,467	84.00% aprx
[Kashif Riaz 2010]	Hindi	ME	--	75.89%
[Kashif Riaz 2010]	Bengali	SVM	150,000	90.00% aprx

6.5 System Design briefing

Initially raw text file has been taken as input for stemming. As example: छ "वर्षसम्म" टि. मार्शल हान "जुनियरले" भद्र र शान्त जर्ज बुस शैलीमा निगम अभिग्रहण गरे।

In the above Nepali sentence two highlighted words are not root word. So they are to be stemmed. After stemming the root word is separated from their corresponding prefixes and suffices. Example: छ "वर्ष" "सम्म" टि. मार्शल हान "जुनियर" "ले" भद्र र शान्त जर्ज बुस शैली मा निगम अभिग्रहण गरे ।

In the above Nepali sentence "वर्ष" and "जुनियर" are the root word and "सम्म" and "ले" are the suffices.

After stemming of the sentences part of speech determination followed by chunking is required for determining the correct proper nouns and common nouns in the document using uni-gram technique of Hidden Markov Model. Example: छ वर्ष सम्म टि.PC मार्शलPC हानPC जुनियरPC ले भद्र र शान्त जर्जPC बुसPC शैली मा निगम अभिग्रहण गरे ।

In the above sentence PC (Proper Common) is tagged to all the NEs (Named Entity) in the sentence. This will determine the presence of proper noun and common noun in the sentence.

After the determination of chunk words Named Entity are tagged using n-gram technique of Hidden Markov Model. Here $n = 5$ i.e atleast five consecutive chunk words can be tagged as Named Entities. Example:

छ वर्ष सम्म < टि. मार्शल हान जुनियर > PERSON ले भद्र र शान्त < जर्ज बुस > PERSON शैली मा निगम अभिग्रहण गरे ।

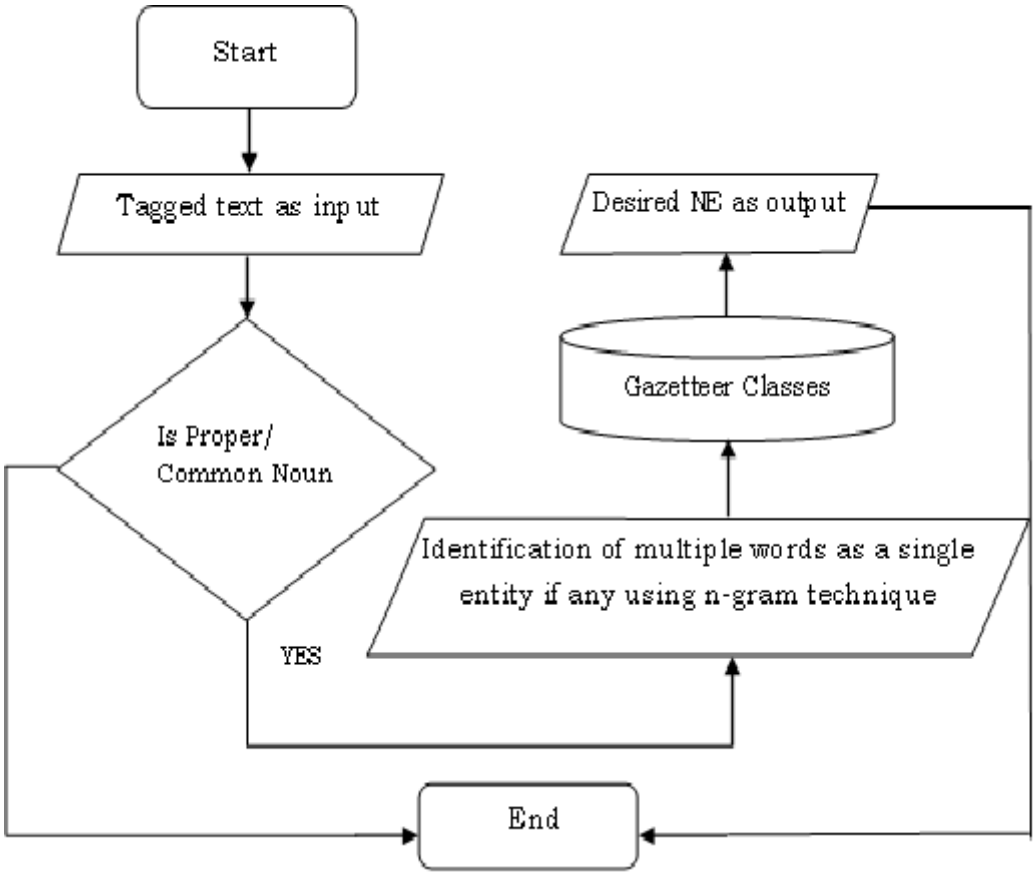

Fig 4: Architecture of NER tool using Gazetteer method

6.6 Result Analysis

We perform n-gram technique (here n=5) with gazetteer method on chunked file which has about 1000 sentences. The searching is done by HASH MAPPING technique which saves time. In our case we have consider three list namely Person(PERSON), Organization(ORG), Location(LOC), Number(NM), Quantifier(QF) and others are left as it as.

Table 10: Total number of tags in the corpus:

	Person(PERSON)	Organization(ORG)	Location(LOC)	Number(NM)	Quantifier (QF)
Total	169	59	31	19	23

In the above table we have collected unique NER tagged entities from 1000 sentences

Table 11: Accuracy is 85.71% from 1000 sentences using n-gram and Gazetteer method.

Accuracy	PERSON		ORG		LOC		NM		QF	
	Total Tag	Correct	Total Tag	Correct	Total Tag	Correct	Total Tag	Correct	Total Tag	Correct
	169	140	59	52	31	28	19	18	23	20
	82.84%		88.14%		90.32%		94.75%		86.95%	

In the above table we have collected individual accuracy details of all the three tags of NER.