Chapter 2

2 REVIEW OF LITERATURE

Named Entity Recognition is a process to discover the Named Entities (NEs) in a text or document and then categorize these Named Entities NEs into different Named Entity classes.

Broadly speaking, named entities are proper nouns. However, named entity tasks often include expressions for date and time, names of sports and adventure activities, terms for biological species and substances as named entities. MUC-7 classifies named entities into following categories and subcategories:

a. Entity (ENAMEX): person, organization, location.

b. Time expression (TIMEX): date, time.

c. Numeric expression (NUMEX): money, percent [Anup Patel Ganesh Ramakrishnan Pushpak Bhattacharya 2009]

2.1 Named Entity Recognition and Classification (NER)

Named Entity Recognition (NER) is essential to identify information units like name of person, organization and location names, and numeric expressions including time, date, money and percent expressions for various Information Retrieval or Information Extraction and NLP tasks. Identification of these entities in text was acknowledged as one of the important sub-tasks of Information Retrieval and Information Extraction and was called “Named Entity Recognition and Classification (NER)”.
Though this sounds clear, special cases arise to require lengthy guidelines, e.g., when is " The Times of India " an artifact , and when is it an organization? When is "White House" an organization, and when a location? Are branch offices of a bank an organization? Is a garment factory a location or an organization? Is a street name a location? Is a phone number a numeric expression or is it an address (location). Is mid-morning a time? In order to achieve human annotator consistency, guidelines with numerous special cases have been defined for the Seventh Message Understanding Conference, MUC-7 .

Most of the researches on Named Entity Recognition (NER) systems has been structured as taking an un-annotated block of text, for e.g.: <PERSON>

श्री हान</PERSON> फाइदा को श्रेय <ORG>कम्पनी</ORG> लाई <QUANTIFIER >सब </QUANTIFIER> भन्दा बढी जानकारी भएको कुरा मा ध्यान केन्द्रित गर्ने दर्शन लाई दिन्छन् |[Arindam Dey, Abhijit Paul, Dr.Bipul Syam Purkayastha 2014]

2.2 NER Applications

Named Entity Recognition (NER) uses most of the applications of natural language processing (NLP). These applications are listed below.

1) In search engines Named Entity Recognition (NER) is very useful. Textual information can be structured using Named Entity Recognition (NER), and structured information helps in efficient indexing and retrieval of documents for search.

2) In the context of Cross-Lingual Information Access Retrieval (CLIR), given a query word, it is very important to find if it is a Named Entity or not. If a query word is a Named Entity, we need to transliterate a query word, rather than translating it.

3) The new generation of news aggregation platforms is powered by Named Entity Recognition. A lot of information can be analyzed using Named Entities, like plotting the popularity of entities over time and generating geospatial heat maps. However, the main improvement to traditional news aggregation brought by Named Entity (NEs) is how they connect between people and things.

4) In machine translation the uses of Named Entity Recognition (NER) is very crucial. Usually, those entities that are identified as Named Entities are transliterated but not translated.

5) If the reader, before reading an article, could know about the Named Entities, the reader would be able to get a reasonable idea about the contents of the article.

6) Automatic indexing of Books: Most of the words indexed in the back index of a book are Named Entities.

7) Named Entity Recognition (NER) detection is very useful in Biomedical domain to identify medicines, vitamins, proteins, diseases, etc.

8) NER detection is usually a sub-task in most of the information retrieval, information extraction tasks because it adds structure to raw information. [Bowen Sun 2010]

2.3 Approaches of NER

There are three approaches of NERs. They are (i) Rule based approach and (ii) Statistical Approach and (iii) Hybrid Approach. [Arindam Dey, Abhijit Paul, Dr.Bipul Syam Purkayastha 2014][Arindam Dey, Dr.Bipul Syam Purkayastha 2013][Anastasia Rita Widiarti, and Phalita Nari Wastu 2009][Anup Patel Ganesh Ramakrishnan Pushpak Bhattacharya 2009]

The Rule Based Approach can either be List lookup Approach or a Linguistic Approach.

For NER detection using lookup approach or linguistic approaches, a lot of human effort is required. A large Gazetteer list has to be built for different Named Entity classes under lookup approach. Then, search operations are performed to find that the given word in the corpus is under which category of the Named Entity Classes. In a linguistic approach, a linguist set the rules and algorithms to determine Named Entities (NEs) in a corpus and also classifies these Named Entities(NEs) into respective Named Entity Classes.[A. Goyal 2008][Asif Ekbal, Rajewanul Hague, Amitava Das, Venkateswarlu Poka and Sivaji Bandyopadhyay 2008][A. Ekbal, R. Hague, and S. Bandyopadhyay 2012][Bowen Sun 2010]

In Statistical Approach very less amount of human labour is required. It is an automated approach. It is of following types:
A. Hidden Markov Model(HMM)
B. Maximum Entropy Model(MEM)
C. Conditional Random Field(CRF)
D. Support Vector Machine(SVM)
E. Decision Tree(DT)[A. Goyal 2008][Arindam Dey, Abhijit Paul, Dr.Bipul Syam Purkayastha 2014]

In Hybrid Approach two approaches can be merged together. It improves the performance of NER system. It can be the combination of Linguistic and Statistical models like Gazetteer list and HMM, HMM and CRF or CRF and MEM etc.

Hidden Markov Model(HMM)

When the state of a method cannot be examined directly, it must be estimated from some sequence of observations. For example, the emotional state of another agent cannot be inspected without peeking into its head, but the emotional state is responsible for the agent's actions so we should be able to estimate the agent's inner state by observing what it is doing.

Hidden Markov Model approaches are used to represent those processes that are not completely observed. They expand the n-gram model with a set of activities that can be observed, and create a probabilistic map between actions and states. A first-order Hidden Markov Model (HMM) is a tuple M = (S, A, p, q) where:
a. S is the set of states in the process,
b. is the set of actions that can be observed,
c. p is the transition probability function, where $p(s_t|s_{t-1})$ signifies the probability of transition from state s_{t-1} to state s_t and
d. q is the action observation probability function, where $q(a_t|s_t)$ denotes the probability of observing action a_t at time t given state s_t.

Maximum Entropy Model(MEM)

The Maximum Entropy model produces a probability distribution for the PP-attachment decision using only information from the verb phrase in which the attachment occurs. We denote the partially parsed verb phrase, i.e. the verb phrase without the attachment decision, as a history h, and the conditional probability of an attachment as $p(d|h)$, where $d \in \{0, 1\}$ and corresponds to a noun or verb attachment (respectively). The probability model depends on certain features of the whole event (h, d) denoted by $f_i(h, d)$. An example of a binary-valued feature function is the indicator function that a particular (V, P) bi-gram occurred along with the attachment decision being V, i.e. $f_{print,on}(h, d)$ is one if and only if the main verb of h is "print", the preposition is "on", and d is "V". The ME principle leads to a model for $p(d|h)$ which maximizes the training data log-likelihood,

$$\frac{\sum}{h,d}\overline{p}(h,d)logp(d|h)$$

where $p^{\sim}(h,w)$ is the empirical distribution of the training set, and where $p(d|h)$ itself is an exponential model:

$$p(d|h) = \frac{\prod_{i=0}^{k} e^{\lambda_i f_i(h,d)}}{\sum_{d=0}^{1} \prod_{i=0}^{k} e^{\lambda_i f_i(h,d)}}$$

At the maximum of the training data log-likelihood, the model has the property that its k parameters, namely the λ_i's, satisfy k constraints on the expected values of feature functions, where the i^{th} constraint is,

$$E_m f_i = \overline{E} f_i$$

The model expected value is,

$$E_m f_i = \sum_{h,d} \overline{p}(h) p(h,d) f_i(h,d)$$

and the training data expected value, also called the desired value, is

$$\overline{E} f_i = \sum_{h,d} \overline{p}(h) p(h,d) f_i(h,d)$$

The values of these k parameters can be obtained by one of many iterative algorithms. For example, one can use the Generalized Iterative Scaling algorithm of Darroch and Ratcliff. As one increases the number of features, the achievable maximum of the training data likelihood increases.

Conditional Random Field(CRF)

CRFs are discriminative models, as they model the conditional distribution over labelling given some contextual observations, $p(s|o)$ where s is the labelling and o is the context. This contrasts with generative models, which model the joint distribution over labelling and the context, p(s,o). These models are commonly used for decoding test instances where only the context is observed. In this case the maximising labelling of the conditional

$p(s|o)$ is required, $s^* = argmax_s p(s|o)$. Discriminative models can be used directly in this instance, where generative models first require normalisation, $p(s|o) = \frac{p(s,o}{\sum s`p(s`,o)}$. This is an advantage of discriminative models, which are trained to maximise the conditional likelihood of the training sample. Discriminative models allow a richer feature representation, which provides more natural and accurate modelling. This benefit often comes at the cost of increased training complexity and reduced flexibility with partially observed data. However, for many NLP tasks the advantages of discriminative models outweigh the disadvantages.

CRFs are most commonly used to model sequencing tasks, where the contextual observations are a sequence of tokens, o = o_1, o_2, . . . , o_N, and the labelling is a sequence of labels of the same length, s = s_1, s_2, . . . , s_N. This corresponds to labelling each token with a single label, as is the case for most tagging tasks. These sequencing CRFs are often referred to as linear chain CRFs; this refers to the chain graphical structure used to describe Markov assumptions over the label sequence. The name Conditional Random Field denotes the modelling of the labelling, S = s, as a network of inter-dependent random variables (a random field), while conditioning over another set of random variables: the context, O = o.

Support Vector Machine(SVM)

The Support Vector Machine (SVM) algorithm (Cortes and Vapnik, 1995) is probably the most widely used kernel learning algorithm. It achieves relatively robust pattern recognition performance using well established concepts in optimization theory. Despite this mathematical classicism, the implementation of efficient SVM solvers has diverged from the classical methods of numerical optimization. This divergence is common to virtually all learning algorithms. The numerical optimization literature focuses on the asymptotical performance: how quickly the accuracy of the solution increases with computing time. In the case of learning algorithms, two other factors mitigate the impact of optimization accuracy.

Consider logistic regression, where the probability $p(y = 1|x; \theta)$ is modelled by $h_\theta(X) = g(\theta^t X)$. We would then predict “1” on an input x if and only if $h_\theta(X) \geq 0.5$, or equivalently, if and only if $\theta^T X \geq 0$.

Consider a positive training example (y = 1). The larger $\theta^T X$, the larger also is $h_\theta(X) = p(y = 1|x; w, b)$, and thus also the higher our degree of confidence that the label is 1. Thus, informally we can think of our prediction as being a very confident one that $y = 1$ if $\theta^T X \gg 0$. Similarly, we think of logistic regression as making a very confident prediction of y = 0, if $\theta^T X \ll 0$. Given a training set, again informally it seems that we would have found a good fit to the training data if we can find θ so that $\theta^T X^{(i)} \gg 0$ whenever $y^{(i)} = 1$ and $\theta^T X^{(i)} \ll 0$ whenever $y^{(i)} = 0$, since this would reflect a very confident (and correct) set of classifications for all the training examples. This seems to be a nice goal to aim for, and we'll soon formalize this idea using the notion of functional margins. For a different type of intuition, consider the following figure, in which x's represent positive training examples, o's denote negative training examples, a decision boundary (this is the line given by the equation $\theta^T X = 0$, and is also called the separating hyper plane) is also shown, and three points have also been A, B and C.

Decision Tree(DT)

A likelihood-based approach to decision tree induction requires a probabilistic model of the process by which data are generated. For a given input x, we assume that a sequence of probabilistic decisions are taken that result in the generation of a corresponding output y. We do not require that this sequence of decisions have a direct correspondence to a process in reality, rather the decisions may simply represent an abstract set of twenty questions that specify, with increasing precision, the location of the conditional mean of y on a non-linear manifold that relates inputs to mean outputs.

We consider regression models in which is a real-valued vector and classification models in which is either a binary scalar or a binary vector with a single non-zero component.In either case the goal is to formulate a conditional probability density of the form, where is a parameter vector. Maximizing a product of such densities with respect to(where is the sample size) yields a maximum likelihood estimate of. Bayesian maximum a posterior estimation can be handled by incorporating a prior on the parameter vector. In a later section, we consider a Markov model in which the likelihood of a data sequence is not simply the product of independent densities.

Current status of Named Entity Recognition (NER)for Indian Language(ILS)

Research in the field of Named Entity Recognition (NER) in Indian languages is still in the initial stage as compared to other foreign languages such as English, Spanish, and Chinese etc. For European Languages especially for English and for East Asian language Accurate Named Entity Recognition (NER) systems are now available. The problem of Named Entity Recognition (NER) is still far from being solved for south and South East Asian languages. For Indian languages there are many issues which make the nature of the problem different.

For example:- The number of frequently used words (common nouns) which can also be used as names (Proper nouns) is very large for European language where a large proportion of the first names are not used as common words.

2.4 Challenges in NER

Named Entity Recognition was first introduced as part of Message Understanding Conference (MUC-6) in 1995 and a related conference MET-1 in 1996 introduced named entity recognition in non-English text. In spite of the recognized importance of names in applications, most text processing 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 across 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.

The goal of NER is first to recognize the potential named entities and then resolve the ambiguity in the name. There are two types of ambiguities in names, structural ambiguity and semantic ambiguity. Wacholder et al. (1997) describes these ambiguities in detail. Non- English names pose another dimension of problems in NER e.g. the most common first name in the world is Muhammad, which can be transliterated as Mohmmed, Muhammad, Mohammad, Mohamed, Mohd and many other variations. These variations make it difficult to find the intended named entity. This transliteration problem can be solved if the name Muhammad is written in Arabic script as محمّد

2.5 Related Works

Although over the years there has been considerable work done for NER in English and other European languages, the interest in the South Asian languages has been quite low until recently. One of the major reasons for the lack of research is the lack of enabling technologies like, parts of speech taggers, gazetteers, and most importantly, corpora and annotated training and test sets. One of the first NER study of South Asian languages and specifically on Urdu was done by Becker and Riaz (2002) who studied the challenges of NER in Urdu text without any available resources at the time. The by-product of that study was the creation of Becker-Riaz Urdu Corpus (2002).

Another notable example of NER in South Asian language is DARPA's TIDES surprise language challenge where a new language is announced by the agency to build language processing tools in a short period of time. In 2003 the language chosen was Hindi. Li and McCallum (2003) tried conditional random fields on Hindi data and reported f-measure ranging from 56 to 71 with different boosting methods. Mukund et al. (2009) used CRF for Urdu NER and showed f-measure of 68.9%.

By far the most comprehensive attempt made to study NER for South Asian and South East Asian languages was by the NER workshop of International Joint Conference of Natural Language Processing in 2008. The workshop attempted to do Named Entity Recognition in Hindi, Bengali, Telugu, Oriya, and Urdu. Among all these languages Urdu is the only one that has Arabic script. Test and training data was provided for each language by different organizations therefore the quantity of the annotated data varied among different languages. Hindi and Bengali led the way with the most amounts of data; Urdu and Oriya were at the bottom with the least amount of data. Urdu had about 36,000 thousand tokens available. A shared task was defined to find named entities in the languages chosen by the researcher. There are 15 papers in the final proceedings of NER workshop at IJCNLP 2008, all cited in the references section, a significant number of those papers tried to address all languages in general, but resorted to Hindi, where the most number of resources were available. Some papers only addressed specific languages like Hindi, Bengali, Telugu and one paper addressed Tamil. There was not a single paper that focused on only Urdu Named Entity Recognition. The papers that tried to address all languages, the computational model showed the lowest performance on Urdu. Among the experiments performed at Named Entity Workshop on various Indian languages and Urdu, almost all experiments used CFR with limited success.

Saha et.al(2008) describes the Maximum Entropy model for the development of Hindi NER. The training data consists of about 234k words, collected from a newspaper domain "Dainik Jagaran" and is tagged manually with 17 classes of Named Entity (NE) including which contains 16,482 NEs. This paper also focused on the development of a module for context pattern semi-automatic learning system. The Named Entity Recognition (NER) was examined using a blind test corpus of 25K words having 5 classes and achieved a F-measure of 81.52%.

Goyal (2008) used Conditional Random Field model for building Hindi Named Entity Recognition (NER). His methodology was used for the evaluation of two test cases where he attains a maximum F-measure of about 49.2%. For individual test cases he obtain a maximum F-measure of 50.1% for case 1 and 44.97% for case 2.

Saha et.al(2008) has developed Maximum Entropy model based Hindi NER system which has identified some appropriate features for Hindi NER detection. English gazetteer list was built for two-phase transliteration method which is very useful in Hindi NER task. The tool has given a very positive response and a considerable performance after using the transliteration method based on gazetteer list. A very effective response was shown by the transliteration approach when it is applied to Bengali NER detection. The Maximum Entropy based system achieved a highest F-measure of 75.89% which is again increased to 81.2% by the introduction of transliteration approach based on gazetteer list.

Li and McCallum (2004) describe Hindi Named Entity Recognition (NER) with the application of Conditional Random Field (CRF) feature. They by providing a large array of lexical text discovered relevant features of CRF and by using this feature they construct the conditional likelihood matrix.

Combination of early-stopping based on the results of 10-fold cross validation and Gaussian prior they reduced over fitting problems.

Gupta and Arora (2009) describes the observation made from the experiment conducted on CRF model for developing Hindi NER. It shows some features which makes the development of NER system complex. It also describes the different approaches for NER. The data used for the training of the model was taken from Tourism domain and it is manually tagged in IOB format.

David Nadeau et al. proposed a named-entity recognition (NER) detection system with their two major limitations which were frequently discussed. First, there is no need for manual labeling of training or creating gazetteer list . Secondly, this system can operate with more than three classes of named entities such as (person, location, and organization).Their proposal of named entity recognition was a combination of simple form of named entity disambiguation with named entity extractions. They use some heuristic approaches which are simple but very highly effective, to perform named entity disambiguation.

Deepti Chopra et al. have discussed about Named Entity Recognition (NER) and its challenges in different Indian languages. They have also describes about different NER detection approaches. They used Hidden Markov Model (HMM) approach to detect NER in Punjabi language and obtain an accuracy and F-measure of about 88.4% .

EXISTING WORK ON DIFFERENT INDIAN LANGUAGES

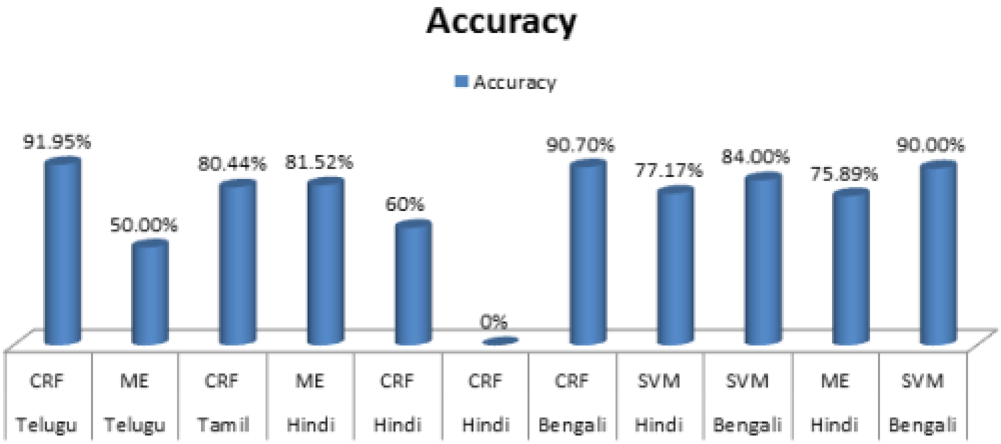

Fig 1: Different Approaches and Their Accuracy