Named Entity Recognition System for Hindi Language: A Hybrid Approach

Shilpi Srivastava *shilpii26@gmail.com*
Department of Computer Science
University of Mumbai, Vidyanagri, Santacruz (E)
Mumbai-400098, India

Mukund Sanglikar *masanglikar@rediffmail.com*
Professor, Department of Mathematics,
Mithibai college, Vile Parle (W), University of Mumbai
Mumbai-400056, India

D.C Kothari *kothari@mu.ac.in*
Professor, Department of Physics,
University of Mumbai, Vidyanagri, Santacruz(E)
Mumbai-400098, India

Abstract

Named Entity Recognition (NER) is a major early step in Natural Language Processing (NLP) tasks like machine translation, text to speech synthesis, natural language understanding etc. It seeks to classify words which represent names in text into predefined categories like location, person-name, organization, date, time etc. In this paper we have used a combination of machine learning and Rule based approaches to classify named entities. The paper introduces a hybrid approach for NER. We have experimented with Statistical approaches like Conditional Random Fields (CRF) & Maximum Entropy (MaxEnt) and Rule based approach based on the set of linguistic rules. Linguistic approach plays a vital role in overcoming the limitations of statistical models for morphologically rich language like Hindi. Also the system uses voting method to improve the performance of the NER system.

Keywords: NER, MaxEnt, CRF, Rule base, Voting, Hybrid Approach

1. INTRODUCTION

Named Entity Recognition is a subtask of Information extraction where we locate and classify proper names in text into predefined categories. NER is a precursor for many natural languages processing tasks. An accurate NER system is needed for machine translation, more accurate internet search engines, automatic indexing of documents, automatic question-answering, information retrieval etc

Most NER systems use a rule based approach or statistical machine learning approach or a combination of these. A Rule-based NER system uses hand-written rules to tag a corpus with named entity (NE) tags. Machine-learning (ML) approaches are popularly used in NER because these are easily trainable, adaptable to different domains and languages and their maintenance is less expensive. A hybrid NER system is a combination of both rule-based and statistical approaches.

Not much work has been done on NER for Indian languages like Hindi. Hindi is the third most spoken language of the world and still no accurate Hindi NER system exists. As some features like capitalization are not available in Hindi and due to lack of a large labeled dataset and of standardization and spelling variations, an English NER system cannot be used directly for Hindi. There is a need to develop an accurate Hindi NER system for better presence of Hindi on the internet. It is necessary to understand Hindi language structure and learn new features for building better Hindi NER systems.

In this paper, we have reported a NER system for Hindi by using the classifiers, namely MaxEnt, CRF and Rulebase model. We have demonstrated a comparative study of performance of the two statistical classifiers (MaxEnt & CRF) widely used in NLP tasks, and use a novel voting mechanism based on classification confidence (that has a statistical validity) to combine the two classifiers among with preliminary handcrafted rules.

Our proposed system is an attempt to illustrate the hybrid approach for Hindi Named Entity Recognition. The system makes use of some POS information of the words along with the variety of orthographic word level features that are helpful in predicting the various NE classes. Theoretically it is known that CRF is better than MaxEnt due to the label bias problem of MaxEnt. The main contribution of this work is to make a comparative study between the two classifiers MaxEnt and CRF and Results show that CRF always gave better results in comparison to MaxEnt.

In the following sections, we will discuss about previous works, the issues in Hindi language & various approaches for NER task and examine our approach, design and implementation details, results and concluding discussion.

2. RELATED WORKS

NER has drawn more and more attention from NLP researchers since the last decade (Chinchor 1995, Chinchor 1998) [5] [18]. Two generally classified approaches to NER are Linguistic approach and Machine learning (ML) based approach. The Linguistics approach uses rule-based models manually written by linguists. ML based techniques make use of a large amount of annotated training data to acquire high-level language knowledge. Various ML techniques which are used for the NER task are Hidden Markov Model (HMM) [7], Maximum Entropy Model (MaxEnt) [6], Decision Tree [3], Support Vector Machines [4] and Conditional Random Fields (CRFs) [10]. Both the approaches may make use of gazetteer information to build system because it improves the accuracy.

Ralph Grishman in 1995 developed a rule-based NER system which uses some specialized name dictionaries including names of all countries, names of major cities, names of companies, common first names etc [15]. Another rule-based NER system is developed in 1996 which make use of several gazetteers like organization names, location names, person names, human titles etc [16]. But the main disadvantages of these rule based techniques are that these require huge experience and grammatical knowledge of particular languages or domains and these systems are not transferable to other languages.

Here we mention a few NER systems that have used ML techniques. 'Identifinder' is one of the first generation ML based NER systems which used Hidden Markov Model (HMM) [7]. By using mainly capital letter and digit information, this system achieved F-value of 87.6 on English. Borthwick used MaxEnt in his NER system with lexical information, section information and dictionary features [6]. He had also shown that ML approaches can be combined with hand-coded systems to achieve better performance. He was able to develop a 92% accurate English NER system. Mikheev et al. has also developed a hybrid system containing statistical and hand coded system that achieved F-value of 93.39 [17].

Other ML approaches like Support Vector Machine (SVM), Conditional Random Field (CRF), and Maximum Entropy Markov Model (MEMM) are also used in developing NER systems. Combinations of different ML approaches are also used. For example, we can mention a system developed by Srihari et al., which combined several modules, built by using MaxEnt, HMM and handcrafted rules, that achieved F-value of 93.5 [19].

The NER task for Hindi has been explored by Cucerzan and Yarowsky in their language independent NER which used morphological and contextual evidences [20]. They ran their experiments with 5 languages: Romanian, English, Greek, Turkish and Hindi. Among these, the accuracy for Hindi was the worst. A Recent Hindi NER system is developed by Li and McCallum using CRF with feature induction [21]. They automatically discovered relevant features by providing a large array of lexical tests and using feature induction to automatically construct the features that mostly increase conditional likelihood. However the performance of these systems is significantly hampered when the test corpus is not similar to the training corpus. Few studies (Guo et al., 2009), (Poibeau and Kosseim, 2001) have been performed towards genre/domain adaptation. But this still remains an open area. In IJCNLP-08 workshop on NER for South and South East Asian languages, held in 2008 at IIIT Hyderabad, was a major attempt in introducing NER for Indian languages that concentrated on five Indian languages- Hindi, Bengali, Oriya, Telugu and Urdu. As part of this shared task, [22] reported a CRF-based system followed by post-processing which involves using some heuristics or rules. Some efforts for Indian Language have also been made [23 [24]. A CRF-based system has been reported in [25], where it has been shown that the hybrid CRF based model can perform better than CRF. [26] presents a hybrid approach for identifying Hindi names, using knowledge infusion from multiple sources of evidence.

The authors, to the best of their knowledge and efforts have not encountered a work which demonstrates a comparative study between the two classifiers MaxEnt and CRF and uses a hybrid model based on MaxEnt, CRF and Rulebase for Hindi Named Entity Recognition.

3. ISSUES WITH HINDI LANGUAGE

The task of building a named entity recognizer for Hindi language presents several issues related to their linguistic characteristics. There are some issues faced by Hindi and other Indian languages:

- No capitalization: Unlike English and most of the European languages, Indian languages lack the capitalization information that plays a very important role to identify NEs in those languages. Hence English NER systems can exploit the feature of capitalization to its advantage because all English names always start with capital letters while Hindi names don't have scripts with graphical cues like capitalization, which could act as an important indicator for NER.

- Ambiguous names: Hindi names are ambiguous and this issue makes the recognition a very difficult task. One of the features of the named entities in Hindi language is the high overlap between common nouns and proper nouns. Indian person names are more diverse compared to those of most other languages and a lot of them can be found in the dictionary as common nouns.

- Scarcity of resources and tools: Hindi, like other Indian languages, is also a resource poor language. Annotated corpora, name dictionaries, good morphological analyzers, POS taggers etc. are not yet available in the required quantity and quality.

- Lack of standardization and spelling: Another important language related issue is the variation in the spellings of proper names. This increases the number of tokens to be learnt by the machine and would perhaps also require a higher level task like co-occurrence resolution.

- Free word order language: Indian languages have relatively free word order.
- Web sources for name lists are available in English, but such lists are not available in Indian languages.
- Although Indian languages have a very old and rich literary history still technology development are recent.
- Indian languages are highly inflected and provide rich and challenging sets of linguistic and statistical features resulting in long and complex word forms.
- Lack of labeled data.
- Non-availability of large gazetteer:

4. VARIOUS APPROACHES FOR NER

There are three basic approaches to NER [1]. They are rule based approach, statistical or machine learning approach and hybrid approach.

4.1 Rule Based Approach

It uses linguistic grammar-based techniques to find named entity (NE) tags. It needs rich and expressive rules and gives good results. It requires great knowledge of grammar and other language related rules. Good experience is needed to come up with good rules and heuristics. It is not easily portable and has high acquisition cost. It is very specific to the target data.

4.2 Statistical Methods or Machine Learning Methods

The common machine learning models used for NER are:

- **HMM [14]:** HMM stands for Hidden Markov Model. HMM is a generative model. The model assigns the joint probability to paired observation and label sequence. Then the parameters are trained to maximize the joint likelihood of training sets.

  It is advantageous as its basic theory is elegant and easy to understand. Hence it is easier to implement and analyze. It uses only positive data, so they can be easily scaled.

  It has few disadvantages. 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.

- **MaxEnt [6]:** MaxEnt stands for Maximum Entropy Markov Model (MEMM). It is a conditional probabilistic sequence model. It can represent multiple features of a word and can also handle long term dependency. It is based on the principle of maximum entropy which states that the least biased model which considers all know facts is the one which maximizes entropy. Each source state has a exponential model that takes the observation feature as input and output a distribution over possible next state. Output labels are associated with states.

  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.

  It also has some disadvantages. It has Label Bias Problem. 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.

- **CRF [10]:** CRF stands for Conditional Random Field. It is a type of discriminative probabilistic model. It has all the advantages of MEMMs without the label bias problem. CRFs are undirected graphical models (also know as random field) which is used to calculate the conditional probability of values on assigned output nodes given the values assigned to other assigned input nodes..

4.3 Hybrid Models

Hybrid models are basically combination of rules based and statistical models. In Hybrid NER system, approach uses the combination of both rule-based and ML technique and makes new methods using strongest points from each method. It is making use of essential feature from ML approaches and uses the rules to make it more efficient.

5. OUR APPROACH

5.1 CRF Based Machine Learning

The basis idea of CRF is to construct a conditional probability $P(Y \mid X)$ from the label sequence Y (e.g. NE tags) and observation sequence X (e.g. words) after model is constructed, then testing can be done by ending the label that maximizes $P(Y \mid X)$ for the observed features.

Definition [10]: " Let $G = (V, E)$ be a graph such that $Y = (Y_v)v \in V$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X , the random variables Y_v obey the Markov Property with respect to the graph: $P(Y_v \mid X, Y_w, w \neq v) = P(Y_v \mid X, Y_w, w \sim v)$; where w ~ v means that w and v are neighbors in G."

"Lafferty et. al [10] define the probability of a particular label sequence Y given the observation sequence X to be a normalized product of potential functions each of the form,

$$\exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i) \right)$$

Where $t_j(y_{i-1}, y_i, x, i)$ is a transition feature function of the entire observation sequence and the labels at positions i and i-1 in the label sequence; $s_k(y_i, x, i)$ is a state feature function of the label at position i and the observation sequence; and λ_j and μ_k are parameters to be estimated from training data.

Final expression of probability of a label sequence Y given an observation sequence X is

$$p(y \mid x, \lambda) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{n} \sum_j \lambda_j f_i(y_{i-1}, y_i, x, i) \right)$$ Where $f_i(y_{i-1}, y_i, x, i)$ is either a state function $s(y_{i-1}, y_i, x, i)$ or a transition function $t(y_{i-1}, y_i, x, i)$." [13]

We are using mallet-0.4 [12] for training and testing. Mallet provides SimpleTagger program that takes input as a file in mallet format of Figure 1. After training the model is saved in a file. Then model file can be used for testing. When trained model is tested, it produces an output file that

contains the predicted tags of the word. The predicted tags are present in the same line number as the text file.

```
word feature_1 feature_2 .... feature_m NE_tag
sansaar noun firstWord none
vishnu noun <ne=NEP>
ki noun verb none
pooja noun none
karte none
hai verb none
, symbol none
narad_muni <ne=NEP>
ki noun verb none
nahi noun none
| symbol none
```

FIGURE 1: Data in mallet format

5.2 MaxEnt Based Machine Learning

It is based on the principle of maximum entropy which states that the least biased model which considers all know facts is the one which maximizes entropy.

Let H be the set of histories and T be the set of allowable tags.
The maximum entropy model is defined over $H \times T$.
The model's probability is defined as probability of history h with tag.

$$p(h,t) = \pi\mu\prod_{j}\alpha_j^{f_i(h,t)}$$

Where,
π is normalization constant
μ, α_j are model parameters
$f_i(h,t)$ feature function
Let $L(p)$ = likelihood of training data using distribution,

$$L(p) = \prod_{i=1}^{n} p(h_i, t_i)$$

The method is to choose the model parameters correctly with respect to maximum likelihood principle.

We are using mallet-0.4 MaxEnt implementation. For the purpose of training and testing using MaxEnt, we created file MaxEntTagger which converts the input file in format specified in Figure 1 into their internal data structure. The file is similar to SimpleTagger. Then the training and testing is done similar to CRF.

5.3 Rule Based Model

Following rules were used to get NE tags from words

- <ne=NEN>: For numbers written in Hindi font like ek, paanch etc, word matching with dictionary is used. The file contain Hindi number words are provided by Hindi Wordnet [11]. If the number contains only digits then it is NEN.

- <ne=NEL>: Use dictionary matching for common locations like Bharat(India), Kanpur. Also used suffix matching like words ending with "pur" are generally cities like Kanpur, Nagpur, Jodhpur etc.

- <ne=NEB>: Used dictionary matching.

- <ne=NETI>: Used regular expression matching e.g. 12-3-2008 format is NETI

- <ne=NEP>: Suffix matching is used with common surnames like Sharma, Agrawal, Kumar etc

- <ne=NED>: Prefix matching with common designation like doctor, raja, pradhanmantri etc.

5.4 Voting

In Voting we use the results of CRF, MaxEnt and Rule Based model to get a better model. We have NE tags including "none". For each word the weight of these tags is initialized 0. Now when the word is predicted as some NE tag by a model then the weight of that tag is increased. The final answer is the tag which has highest weight.

Some heuristics are used to improve the accuracy of model. Like weight of NEM tags predicted by rule based model is kept high as they generally predict correct NE tag. If two tags are same then the answer is that tag.

6. DESIGN & IMPLEMENTATION

6.1 Data and Tools

- **Dataset:** Named Entity Annotated Corpus for Hindi. The data is obtained from IJCNLP-08 website [8]. SSF format [9] is used for representing the annotated Hindi corpus. The annotation was performed manually by IIIT Hyderabad.

- **Dictionary Source:** We have used files containing common Hindi nouns, verbs, adjectives, adverbs for Parts-of-speech (POS) tagging. The files are obtained from Hindi Wordnet, IIT Mumbai [11].

- **Tools:** Mallet-0.4 [12] is used for training and testing machine learning based models CRF [10] and MaxEnt [6]. For CRF, a SimpleTagger is provided which takes input as a file containing word followed by word features (noun, verb, number etc) and Named Entity (NE) tag for training. A SimpleTagger program converts the file into suitable data structures used by CRF for training.

  e.g. Training file format:
  Word feaure_1 feature_2 ... feature_n NE_tag
  ek noun adj number <ne=NEN>
  adhik adj adv none

  Here word "ek" has 3 features namely noun, adj and number. Its NE tag is <ne=NEN>. Second word "adhik" has 2 features namely adj and adv and it has NE tag none.

  For testing the file format is same except it doesn't contain NE tags at last of each sentence i.e. it only contains words followed by its features

  For MaxEnt, we created MaxEntTagger.java to process the input file and use them to test and train MaxEnt model.

- **Tagset Used:** Table 1 [2] contains the list Named Entity tagset used in the corpus.

- Programming Language & utility: Java, bash script, awk, grep

Tags	Names	Description
<ne=NEP>	Person	Bob Dylan, Mohandas Gandhi
<ne=NED>	Designation	General Manager, Commissioner
<ne=NEO>	Organization	Municipal Corporation
<ne=NEA>	Abbreviation	NLP, B.J.P.
<ne=NEB>	Brand	Pepsi, Nike (ambiguous)
<ne=NETP>	Title Person	Mahatma, Dr., Mr.
<ne=NETO>	Title Object	Pride and Prejudice, Othello
<ne=NEL>	Location	New Delhi, Paris
<ne=NETI>	Time	3rd September, 1991(ambiguous)
<ne=NEN>	Number	3.14, 4,500
<ne=NEM>	Measure	Rs. 4,500, 5 kg
<ne=NETE>	Terms	Maximum Entropy, Archeology
None	Not a named entity	Rain, go, hai, ka, ke , ki

TABLE 1: The named entity tagset used for shared task

6.2 Design Schemes

- **Editing Data:** The first objective is to convert annotated Hindi corpus given in SSF format to new format that can be used by mallet-0.4 models CRF and MaxEnt for training and testing. SSF format like the example given in Figure 2 contains many things like line number, braces, <Sentence id=""> etc that are not present in mallet format (e.g. data format of Figure 3). NE tags are present in different line in SSF, which need to put after the word for mallet format. Also some words which represents a NE tag when combined like "narad muni" in Figure 2 needs to be concatenated. After writing each word in different line with their NE tags, we need to find features for each word.

- **Features:** Here we used mostly orthographic features like other researchers have been using. Features of words include

    - Symbol: If the word is symbol like "?", ",", ";", "." etc
    - Noun: If word is noun
    - Adj: The word is adjective
    - Adv: adverb
    - Verb: verb
    - First Word: If the word is first word of a sentence
    - Number: If the word is a number like ek, paanch, or 123,
    - Num Start: If the word starts with number line 123_kg

  Features of the words are added using some rule based matching (like for numbers) and from dictionary matching of words with the words which are obtained from Hindi wordnet, IIT Mumbai [11] (like noun, verb).

- **Training and Testing on Mallet:** The model is trained on 10, 50, 100, and 150 training files respectively. Then each trained model is tested on 10 files on which the model is not trained. The files on which the model is trained and tested are obtained randomly from the dataset. This process is done for 10 times. The average and good results of these tests are reported in the Results section. This is done for both CRF and MaxEnt model on the given data.

```
<Sentence id="">
0 (( SSF
1 sansaar
2 (( NP <ne=NFP>
2.1 vishnu
))
3 ki
4 pooja
5 karte
6 hai
7 ,
8 (( NP <ne=NEP>
8.1 narad
8.2 muni
))
9 ki
10 nahi
11 |
))
</Sentence>
```

FIGURE 2: Data in SSF format

```
word feature_1 feature_2 .... feature_m NE_tag
sansaar noun firstWord none
vishnu noun <ne=NEP>
ki noun verb none
pooja noun none
karte none
hai verb none
, symbol none
narad_muni <ne=NEP>
ki noun verb none
nahi noun none
| symbol none
```

FIGURE 3: Data in mallet format after conversion from SSF

- **Test Dataset using Rule Based Models:** test all datasets for Rule based models.

- **Improve Accuracy by Voting:** The output of each of the above method (CRF, MaxEnt, rule based) is file containing predicted tags for each word in the same line as the word. Voting algorithm uses trained CRF and MaxEnt model and rule based model's result and used the result of these to give better results. Voting is done on the results of these three models and the one with the most weight is the final tag.

7. RESULTS

7.1 Performance Evaluation Metric

The Evaluation measure for the data sets is precision, recall and F Measure.

- **Precision (P):** Precision is the fraction of the documents retrieved that are relevant to the user's information need.

$$\text{Precision(P)} = \frac{\text{correct answers}}{\text{answers produced}}$$

- **Recall (R):** Recall is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{Recall(R)} = \frac{\text{correct answers}}{\text{total possible correct answers}}$$

- **F-Measure:** The weighted harmonic mean of precision and recall, the traditional F-measure or balanced F-score is

$$F - Measure = \frac{(\beta^2 + 1)PR}{\beta^2 R + P}$$

β is the weighting between precision and recall typically $\beta = 1$.

When recall and precision are evenly weighted i.e. $\beta = 1$, F-measure is called F1-Measure.

$$F1 - Measure = \frac{2PR}{(P + R)}$$

There is a tradeoff between precision and recall in the performance metric.

7.2 Results Obtained

- **CRF Results:** The following table contains the results obtained from testing CRF models. The model is trained on 10, 50, 100 and 150 files and then tested on 10 files. This is done for 10 rounds i.e. for model trained on 100 files, 110 files are selected from the dataset and it is trained on 100 files and tested on 10 files(model trained on 10 files are tested on 5 files). Then again 110 files are chosen and training and testing is done. This is done for 10 times. Table 2 contains the results obtained from the above experiment.

Number of training files	Number of testing files	Precision	Recall	F-1 Measure
10	5	71.43	30.86	43.10
50	10	83.87	25.74	39.40
100	10	88.24	24.19	37.97
150	10	88.89	24.61	38.55

TABLE 2: CRF results for one best predicted tag

For the above experiments only one predicted tag of a word is considered. Since the number of NE tags are less compared to “none” tag, so the model learns mostly for “none” tag. So we considered using best of two of the predicted tags of a word to check the results. Here two best predicted tags are given by the model. The two tags can be either same or different. If first tag is a NE tag then that tag is considered correct. If first is none tag and second is NE tag then second tag is considered for the results. This experiment is also conducted in a similar manner as the above experiment.

The results obtained from the above experiment for CRF when two of the best predicted tags are taken into consideration is shown in the Table 3:

Number of training files	Number of testing files	Precision	Recall	F-1 Measure
10	5	70.0	34.57	46.28
50	10	89.28	49.5	63.69
100	10	83.33	33.9	48.19
150	10	74.28	33.37	46.43

TABLE 3: CRF results for best of two predicted tags

- **MaxEnt Results:** Following tables contain the results of training and testing of MaxEnt model. The model is trained on randomly chosen 10, 50, 100 and 150 files and then tested on 10 files on which it is not trained. Each of the training and testing is done for ten rounds. Similar to above these are also tested on different datasets. The results obtained is shown in the following table 4:

Number of training files	Number of testing files	Precision	Recall	F-1 Measure
10	5	76.92	19.8	31.49
50	10	70.40	16.68	26.39
100	10	69.21	18.14	28.19
150	10	69.46	16.57	26.06

TABLE 4: MaxEnt Results for one best predicted tag

MaxEnt results when two of the best predicted tags are taken into consideration are given in Table 5. This is done in similar way as done in CRF experiment.

Number of training files	Number of testing files	Precision	Recall	F-1 Measure
10	5	90.47	29.23	44.18
50	10	89.28	21.36	34.48
100	10	87.5	22.58	35.89
150	10	96.15	25.25	39.99

T

TABLE 5: MaxEnt Results for best of two predicted tags

- **Rule Based Results:** Results driven from rule based model is given below in Table 6:

Number of testing files	Precision	Recall	F-1 Measure
1	65.93	77.92	71.43
2	88.0	60.27	71.54
3	96.05	86.90	91.25

TABLE 6: Rule based model's test results

- **Voting Algorithm:** For voting we used three classifiers crf trained on 50 files, MaxEnt trained on 50 files and rule based. Results from voting algorithm model is given in Table 7:

Number of testing files	Precision	Recall	F-1 measure
40	81.11	84.88	82.95
40	85.51	76.62	80.82

TABLE 7: Voting Algorithm's Results

8. CONCLUSION

Basically this paper presents a comparative study among different approaches like MaxEnt, CRF and Rulebase using POS & orthographic features. It also shows that voting mechanism gives the better results. On average CRF gives better result than MaxEnt. Rule based result has better recall and F-1 measure. On the given data the average precision is good. The main reason for the lower F-1 measure by CRF and MaxEnt is due to the presence of less NE tags in the original data compared to "none". For most file the percentage of NE tags is less that 2% of the total words present in a file. Because of that the classifier is learned more strongly for "none" rather than NE tags. Also data has tagging errors. e.g. "Gandhi" is classified as <ne=NEN>, <ne=NEP>, <ne=NED>,"none" in many files. Similarly "ek" is classified as <ne=NEN> or "none". These conflicting cases in the training set weaken the classifier. That's why more training doesn't give better results here. The classifier gives good precisions i.e. less tags are classified but they are classified correctly.

When we took best of two predicted tags for the results analysis F-1 measure and recall increases significantly. Since we have very few NE tags in data and also data is not very accurate, so most of the words are learned as "none", but when we consider best of two predicted tags, the result improves significantly. Rule based model gives better average result (F-1 measure, recall) for given data. Voting algorithm improves the F-1 measure of results.

9. FUTURE WORK

Dictionary matching of words is not very effective. In this experiment we used Othographic features like other researchers however POS tagger or morphological analyzer, semantic tags, parasargs (prepositions and postpositions) identification, lexicon database and co-occurrences may give the better results. Boosting may be done by containing 5 words above NE tags and 5 words below NE tags. Conflicting tags can be removed. Or we may try using another dataset. More features can be added to improve the models. Rule based model can be improved. We may experiment with other classifier like HMM.

10. ACKNOWLEDGMENT

I would like to thank Mr. Pankaj Srivastava, Ms. Agrima Srivastava and MS. Vertika Khanna who provide helpful analysis in model development.

11. REFERENCES:

[1] Sudeshna Sarkar, Sujan Saha and Prthasarthi Ghosh, "Named Entity Recognition for Hindi", In Microsoft Research India Summer School talk, p. 21-30, May 2007.

[2] Anil Kumar Singh, "Named Entity Recognition for South and South East Asian Languages: Taking Stock", p. 5-7, In IJCNLP 2008.

[3] Hideki Isozaki. 2001. “Japanese named entity recognition based on a simple rule generator and decision tree learning” in the proceedings of the Association for Computational Linguistics, pages 306-313. India.

[4] Takeuchi K. and Collier N. 2002. “Use of Support Vector Machines in extended named entity recognition” in the proceedings of the sixth Conference on Natural Language Learning (CoNLL-2002), Taipei, Taiwan, China.

[5] Charles L. Wayne. 1991., “A snapshot of two DARPA speech and Natural Language Programs” in the proceedings of workshop on Speech and Natural Languages, pages 103-404, Pacific Grove, California. Association for Computational Linguistics.

[6] A. Borthwick, "A Maximum Entropy Approach to Named Entity Recognition", In NY University, p. 1-4, 18-24, PHD Thesis, September 1999

[7] Daniel M. Bikel, Scott Miller, Richard Schwartz and Ralph Weischedel. 1997 “Nymble: a high performance learning name-finder” in the proceedings of the fifth conference on Applied natural language processing, pages 194-201, San Francisco, CA, USA Morgan Kaufmann Publishers Inc.

[8] IJCNLP-08 Workshop data set, Source: http://ltrc.iiit.net/ner-ssea-08/index.cgi?topic=5

[9] Akshar Bharti, Rajeev Sangal and Dipti M Sharma, "Shakti Analyzer: SSF Representation", IIIT Hyderabad, p. 3-5, 2006

[10] Lafferty, J., McCallum, A., Pereira, F., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, p. 1-5, 2001

[11] Hindi Wordnet, Source: http://www.cfilt.iitb.ac.in/wordnet/webhwn/

[12] McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

[13] Hanna M. Wallach, "Conditional Random Fields: An Introduction”, Technical Report, University of Pennsylvania. 4-5, 2004.

[14] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", In Proceedings of the IEEE, 77 (2), p. 257-286,February 1989

[15] R. Grishman. 1995. “The NYU system for MUC-6 or Where’s the Syntax” in the proceedings of Sixth Message Understanding Conference (MUC-6) , pages 167-195, Fairfax, Virginia.

[16] Wakao T., Gaizauskas R. and Wilks Y. 1996. “Evaluation of an algorithm for the Recognition and Classification of Proper Names”, in the proceedings of COLING-96.

[17] Mikheev A, Grover C. and Moens M. 1998. Description of the LTG system used for MUC-7. In Proceedings of the Seventh Message Understanding Conference.

[18] R. Grishman, Beth Sundheim. 1996. “Message Understanding Conference-6: A Brief History” in the proceedings of the 16th International Conference on Computational Linguistics (COLING), pages 466-471, Center for Sprogteknologi, Copenhagen, Denmark.

[19] Srihari R., Niu C. and Li W. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. In: Proceedings of the sixth conference on applied natural language processing.

[20] Cucerzan S. and Yarowsky D. 1999. Language independent named entity recognition combining morphological and contextual evidence. In: Proceedings of the Joint SIGDAT Conference on EMNLP and VLC 1999, pp. 90-99.

[21] Li W. and McCallum A. 2003. Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction. In: ACM Transactions on Asian Language Information Processing (TALIP), 2(3): 290–294.

[22] Gali, K., Sharma, H., Vaidya, A., Shisthla, P., Sharma, D.M.: Aggregrating Machine Learning and Rule-based Heuristics for Named Entity Recognition. In: Proceedings of the IJCNLP-08Workshop on NER for South and South East Asian Languages. (2008) 25–32

[23] Asif Ekbal et. al. "Language Independent Named Entity Recognition in Indian Languages". IJCNLP, 2008.

[24] Prasad Pingli et al. "A Hybrid Approach for Named Entity Recognition in Indian Languages". IJCNLP, 2008.

[25] Shilpi Srivastava, Siby Abraham, Mukund Sanglikar: "Hybrid Approach for Recognizing Hindi Named Entity", Proceedings of the International Conference on Managing Next Generation Software Applications - 2008 (MNGSA 2008), Coimbatore, India, 5th- 6th December 2008.

[26] Shilpi Srivastava, Siby Abraham, Mukund Sanglikar, D C Kothari: "Role of Ensemble Learning in Identifying Hindi Names", International Journal of Computer Science and Applications, ISSN No. 0974-0767.