A Hidden Markov Model Based Named Entity Recognition System: Bengali and Hindi as Case Studies

Asif Ekbal and Sivaji Bandyopadhyay

Computer Science and Engineering Department, Jadavpur University, Kolkata, India
asif.ekbal@gmail.com, sivaji_cse_ju@yahoo.com

Abstract. Named Entity Recognition (NER) has an important role in almost all Natural Language Processing (NLP) application areas including information retrieval, machine translation, question-answering system, automatic summarization etc. This paper reports about the development of a statistical Hidden Markov Model (HMM) based NER system. The system is initially developed for Bengali using a tagged Bengali news corpus, developed from the archive of a leading Bengali newspaper available in the web. The system is trained with a training corpus of 150,000 wordforms, initially tagged with a HMM based part of speech (POS) tagger. Evaluation results of the 10-fold cross validation test yield an average Recall, Precision and F-Score values of 90.2%, 79.48% and 84.5%, respectively. This HMM based NER system is then trained and tested on the Hindi data to show its effectiveness towards the language independent abilities. Experimental results of the 10-fold cross validation test has demonstrated the average Recall, Precision and F-Score values of 82.5%, 74.6% and 78.35%, respectively with 27,151 Hindi wordforms.

Keywords: Named Entity (NE), Named Entity Recognition (NER), Hidden Markov Model (HMM), Named Entity Recognition in Bengali.

1 Introduction

Named Entity Recognition is an important tool in almost all NLP application areas. The objective of named entity recognition is to identify and classify every word/term in a document into some predefined categories like person name, location name, organization name, miscellaneous name (date, time, percentage and monetary expressions) and "none-of-the-above". The challenge in detection of named entities is that such expressions are hard to analyze using traditional NLP because they belong to the open class of expressions, i.e., there is an infinite variety and new expressions are constantly being invented.

During the last decade, NER has drawn more and more attention from the NE tasks [1] [2] in Message Understanding Conferences (MUCs) [MUC6; MUC7]. This reflects the importance of NER in information extraction. The problem of correct identification of named entities is specifically addressed and benchmarked

A. Ghosh, R.K. De, and S.K. Pal (Eds.): PReMI 2007, LNCS 4815, pp. 545–552, 2007.

by the developers of Information Extraction System, such as the GATE system [3]. NER also finds application in question-answering [4] systems and machine translation [5].

The current trend in NER is to use the machine learning (ML) approach, which is more attractive in that it is trainable and adoptable and the maintenance of a ML system is much cheaper than that of a rule-based one. 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 steep. The representative machine-learning approaches used in NER are HMM (BBN's IdentiFinder in [6]), Maximum Entropy (New York University's MENE in [7]) and Conditional Random Fields [8].

Among the machine learning approaches, the evaluation performance of the HMM is quite impressive. The main reason may be due to its better ability of capturing the locality of phenomena, which indicates names in text. Moreover, HMM seems more and more used in NER because of the efficiency of the Viterbi algorithm [9] used in decoding the NE-class state sequences. Zhou and Su [10] reported state of the art results on the MUC-6 and MUC-7 data using an HMM-based tagger. However, the performance of a ML system is always poorer than that of a rule-based one. This may be because current ML approaches capture important evidences behind NER problem much less effectively than human experts who handcraft the rules, although machine learning approaches always provide important statistical information that is not available to human experts.

All the works, carried out already in the area of NER, are in non-Indian languages. In Indian languages, particularly in Bengali, the works in the area of named entity recognition can be found in [11] [12]. Other than Bengali, the work on NER can be found in [13] for Hindi. Here, in this paper, an HMM based NER system has been reported that outperforms the systems developed in [11] [12].

The paper is organized as follows. The NER system has been described in Section 2. Experimental results with the 10-fold cross validation tests in terms of three evaluation parameters, Precision, Recall and F-Score, are reported in Section 3 for Bengali and Hindi. Finally, Section 4 concludes the paper.

2 Named Entity Recognition in Bengali and Hindi

Bengali is one of the widely used languages all over the world. It is the seventh popular language in the world, second in India and the national language of Bangladesh. Hindi is the national language of India. Named Entity (NE) identification in Indian languages (ILs) in general is difficult and challenging as there is no concept of capitalization in ILs. A tagged Bengali news corpus, developed from the archive of a widely read Bengali news paper available in the web, has been used in this work for NER in Bengali. At present the corpus contains around 34 million wordforms in ISCII (Indian Standard Code for Information

Interchange) and UTF-8 format. The location, reporter, agency and different date tags in the tagged corpus help to identify some NEs that appear in the fixed places of the newspaper. The training corpus for Hindi was obtained from the SPSAL 20007 Contest [1]. An HMM based named entity tagger has been used in this work to identify named entities in Bengali/Hindi and classify them into person, location, organization and miscellaneous names. Miscellaneous names include date, time, percentage and monetary expressions. To apply HMM in named entity tagging task the NE tags, as shown in Table 1, are defined.

Table 1. Named Entity Tag Set

NE Tag	Meaning	Example
PER	Single-word person name	*sachin*/PER
LOC	Single-word location name	*jadavpur*/LOC
ORG	Single-word organization name	*infosys*/ORG
MISC	Single-word miscellaneous name	*100%*/MISC
B-PER	Beginning, Internal or the	*sachin*/B-PER
I-PER	End of a multi-word	*ramesh*/I-PER
E-PER	person name	*tendulkar*/E-PER
B-LOC	Beginning, Internal or the	*mahatma*/B-LOC
I-LOC	End of a multi-word	*gandhi*/I-LOC
E-LOC	location name	*road*/E-LOC
B-ORG	Beginning, Internal or the	*bhaba*/B-ORG
I-ORG	End of a multi-word	*atomic*/I-ORG *research*/I-ORG
E-ORG	orgnization name	*centre*/E-ORG
B-MISC	Beginning, Internal or the	*10 e*/B-MISC
I-MISC	End of a multi-word	*magh*/I-MISC
E-MISC	miscellaneous name	*1402*/E-MISC
NNE	Not Named Entity	*neta*/NNE, *bidhansabha*/NNE

2.1 Hidden Markov Model Based Named Entity Tagging

The goal of NER is to find a stochastic optimal tag sequence $T = t_1, t_2, t_3, \ldots, t_n$ for a given word sequence $W = w_1, w_2, w_3 \ldots, w_n$. Generally, the most probable tag sequence is assigned to each sentence following the Viterbi algorithm [9]. The tagging problem becomes equivalent to searching for $argmax_T P(T) * P(W|T)$, by the application of Bayes' law ($P(W)$ is constant).

The probability of the NE tag, i.e., $P(T)$ can be calculated by Markov assumption which states that the probability of a tag is dependent only on a small, fixed number of previous NE tags. Here, in this work, a trigram model has been used. So, the probability of a NE tag depends on two previous tags, and then we have,

$$P(T) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_1, t_2) \times P(t_4|t_2, t_3) \times \ldots \times P(t_n|t_{n-2}, t_{n-1})$$

[1] http://shiva.iiit.ac.in/SPSAL2007/check_login.php

An additional tag '$' (dummy tag) has been introduced in this work to represent the beginning of a sentence. So, the previous probability equation can be slightly modified as:

$$P(T) = P(t_1|\$) \times P(t_2|\$, t_1) \times P(t_3|t_1, t_2) \times P(t_4|t_2, t_3) \times \ldots \times P(t_n|t_{n-2}, t_{n-1})$$

Due to sparse data problem, the linear interpolation method has been used to smooth the trigram probabilities as follows: $P'(t_n|t_{n-2}, t_{n-1}) = \lambda_1 P(t_n) + \lambda_2 P(t_n|t_{n-1}) + \lambda_3 P(t_n|t_{n-2}, t_{n-1})$ such that the λs sum to 1. The values of λs have been calculated by the following method [14]:

1. set $\lambda_1 = \lambda_2 = \lambda_3 = 0$
2. for each tri-gram (t_1, t_2, t_3) with $freq(t_1, t_2, t_3) > 0$ depending on the maximum of the following three values:
   - case: $\frac{(freq(t_1,t_2,t_3)-1)}{(freq(t_1,t_2)-1)}$: increment λ_3 by $freq(t_1, t_2, t_3)$
   - case: $\frac{(freq(t_2,t_3)-1)}{(freq(t_2)-1)}$: increment λ_2 by $freq(t_1, t_2, t_3)$
   - case: $\frac{(freq(t_3)-1)}{(N-1)}$: increment λ_1 by $freq(t_1, t_2, t_3)$
3. normalize λ_1, λ_2, λ_3.

Here, N is the corpus size, i.e., the number of tokens present in the training corpus. If the denominator in one of the expression is 0, then the result of that expression is defined to be 0. The -1 in both the numerator and denominator has been considered for taking unseen data into account.

By making the simplifying assumption that the relation between a word and its tag is independent of context, $P(W|T)$ can be calculated as:
$P(W|T) \approx P(w_1|t_1) \times P(w_2|t_2) \times \ldots \times P(w_n|t_n)$.

The emission probabilities in the above equation can be calculated from the training set as, $P(w_i|t_i) = \frac{freq(w_i|t_i)}{freq(t_i)}$.

2.2 Context Dependency

To make the Markov model more powerful, additional context dependent features are introduced to the emission probability in this work. This specifies that the probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. Now, $P(W|T)$ is calculated by the equation:

$$P(W|T) \approx P(w_1|\$, t_1) \times P(w_2|t_1, t_2) \times \ldots \times P(w_n|t_{n-1}, t_n)$$

So, the emission probability can be calculated as:

$$P(w_i|t_{i-1}, t_i) = \frac{freq(t_{i-1}, t_i, w_i)}{freq(t_{i-1}, t_i)}.$$

Here, also the smoothing technique is applied rather than using the emission probability directly. The emission probability is calculated as:

$P'(w_i|t_{i-1}, t_i) = \theta_1 P(w_i|t_i) + \theta_2 P(w_i|t_{i-1}, t_i)$, where θ_1, θ_2 are two constants such that all θs sum to 1.

The values of θs should be different for different words. But the calculation of θs for every word takes a considerable time and hence θs are calculated for the entire training corpus. In general, the values of θs can be calculated by the same method that is adopted in calculating λs.

2.3 Viterbi Algorithm

The Viterbi algorithm [9] allows us to find the best T in linear time. The idea behind the algorithm is that of all the state sequences, only the most probable of these sequences need to be considered. The trigram model has been used in the present work. The pseudo code of the algorithm is shown bellow.

```
for i = 1to Number_of_Words_in_Sentence
  for each state c ∈Tag_Set
    for each state b ∈Tag_Set
      for each state a ∈ Tag_Set
        for the best state sequence ending in state
         a at time (i − 2), b at time (i − 1), compute
        the probability of that state sequence going
        to state c at time i.
      end
    end
  end
Determine the most-probable state sequence ending in state c at time i.
end
```

So if every word can have S possible tags, then the Viterbi algorithm runs in $O(S^3 \times |W|)$ time, or linear time with respect to the length of the sentence.

2.4 Handling the Unknown Words

Handling of unknown words is an important issue in NE tagging. Viterbi algorithm [9] attempts to assign a NE tag to the unknown words. Specifically, suffix features of the words and a lexicon are used to handle the unknown words in Bengali.

For words which have not been seen in the training set, $P(w_i|t_i)$ is estimated based on features of the unknown words, such as whether the word contains a particular suffix. There may be two different kinds of suffixes that could be helpful in predicting the NE classes of the unknown words. The corresponding lists have been prepared. The first category contains the suffixes that could usually appear at the end of different NEs and non-NEs. This list has 435 entries including a null suffix that has been kept for those words that have none of the suffixes in the list. The second category is the set of suffixes that may occur with person names (e.g., -*babu*, -*da*, -*di* etc.) and location names (e.g., -*land*, -*pur*, -*lia* etc.). The person and location lists contain 51 and 46 entries respectively. The probability distribution of a particular suffix with respect to specific tag

is generated from all words in the training set that share the same suffix. Two additional features that cover the numbers and symbols are also considered.

To handle the unknown words further, a lexicon [15], which was developed in an unsupervised way from the tagged Bengali news corpus, has been used. Lexicon contains the Bengali root words and their basic part of speech information such as: noun, verb, adjective, adverb, pronoun and indeclinable, excluding NEs. The lexicon has around 100,000 word entries. The heuristic is that '*if an unknown word is found to appear in the lexicon, then most likely it is not a named entity*'.

3 Experimental Results

A portion of the tagged (not NE tagged/POS tagged) news corpus, containing 150,000 wordforms, has been used to train the NER system. The training corpus is initially run through an HMM-based part of speech (POS) tagger [16] to tag the training corpus with the 26 different POS tags [2], defined for the Indian languages. This POS-tagged training set is then manually checked for the correctness. The POS tags representing NEs are replaced by the appropriate NE tags as defined in Table 1, and the rest are replaced by the NNE tags. The training set thus obtained is a corpus tagged with sixteen NE tags and one non-NE tag. In the output, sixteen NE tags are replaced appropriately by the four NE tags, viz., 'Person', 'Location', 'Organization' and 'Miscellaneous'.

The NER system has been evaluated in terms of Recall, Precision and F-Score as defined below:

$$\text{Recall (R)} = \frac{\text{(No. of tagged NEs)}}{\text{(Total no. of NEs present in the test set)}} \times 100\%$$

$$\text{Precision (P)} = \frac{\text{(No. of correctly tagged NEs)}}{\text{(No. of tagged NEs)}} \times 100\%$$

$$\text{F-Score (FS)} = \frac{(2 \times \text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \times 100\%$$

The training set is initially distributed into 10 subsets of equal size. In the cross validation test, one subset is withheld for testing while the remaining 9 subsets are used as the training sets. This process is repeated 10 times to yield an average result, which is called the 10-fold cross validation test. The experimental results of the 10-fold cross validation test are reported in Table 2. The NER system has demonstrated an average Recall, Precision and F-Score values of 90.2%, 79.48% and 84.5%, respectively. A close investigation to the experimental results reveals that the precision errors are mostly concerned with the organization names. The lack of robustness of the system to handle the unknown organization names properly might be the possible reason behind the fall in precision of organization names. Unlike person or location names, there is no list of suffixes that could be helpful in predicting the class of the unknown organization names. The other existing Bengali NER systems [11] [12] were also trained and tested with the

[2] http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

same dataset. Evaluation results of these systems demonstrated average F-Score values of 74.5% and 77.9% with the 10-fold cross validation test.

The HMM based NER system is also trained and tested with the Hindi data [3] to show its effectiveness for the language independent nature. The Hindi data was tagged with 26 POS tags. The POS tags representing NEs are replaced appropriately by the NE tags of Table 1, and the rest are replaced by the NNE tags. Evaluation results of the system for the 10-fold cross validation test yield an average Recall, Precision and F-Score values of 82.5%, 74.6% and 78.35%, respectively with a training corpus of 27,151 wordforms. The experimental results are presented in Table 3. The one possible reason behind the poorer performance of the system for Hindi might be the smaller amount of training data in comparison to Bengali. Another reason may be its inability to handle the unknown words as efficiently as Bengali. Unlike Bengali, there are no lists of suffixes or lexicon for Hindi in the system.

Table 2. Results of 10-fold cross validation test for Bengali

Test Set	1	2	3	4	5	6	7	8	9	10	Average
Recall	90.80	90.75	90.63	90.49	90.31	90.25	90.12	89.81	89.72	90.12	90.30
Precision	80.40	80.30	79.15	79.87	79.75	79.52	79.39	78.12	78.27	80.40	79.52
F-Score	85.29	84.98	84.50	84.85	84.70	84.55	84.42	83.56	83.60	84.98	84.50

Table 3. Results of 10-fold cross validation test for Hindi

Test Set	1	2	3	4	5	6	7	8	9	10	Average
Recall	83.10	82.80	82.62	82.91	82.73	82.54	82.31	82.76	82.65	82.58	82.50
Precision	75.34	75.13	74.97	74.81	74.12	73.90	73.73	74.46	74.96	74.58	74.60
F-Score	79.03	78.78	78.61	78.65	78.19	77.98	77.78	77.94	78.16	78.38	78.35

4 Conclusion

In this paper, we have presented a named entity recognizer that uses HMM framework with more contextual information. The system has been evaluated with Bengali and Hindi data. The system uses a HMM based POS tagger for the preparation of training data for Bengali. The evaluation results of 10-fold cross validation test shows that such system has high Recall and good F-Score values for Bengali. The system has also shown good Recall and impressive F-Score values with a relatively smaller Hindi training set. Future works include investigating methods to boost precision of the NER system. Building NER

[3] http://shiva.iiit.ac.in/SPSAL2007/check_login.php

systems for Bengali using other statistical techniques like Maximum Entropy Markov Model (MEMM), Conditional Random Fields (CRFs) and analyzing the performance of these systems is another interesting task.

References

1. Chinchor, N.: MUC-6 Named Entity Task Definition (Version 2.1). In: MUC-6, Maryland (1995)
2. Chinchor, N.: MUC-7 Named Entity Task Definition (Version 3.5). In: MUC-7, Fairfax (1998)
3. Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities 36, 223–254 (2002)
4. Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A., Bolohan, O.: LCC Tools for Question Answering. In: Text REtrieval Conference (TREC 2002) (2002)
5. Babych, B., Hartley, A.: Improving Machine Translation Quality with Automatic Named Entity Recognition. In: Proceedings of EAMT/EACL 2003 Workshop on MT and other Language Technology Tools, pp. 1–8 (2003)
6. Bikel, D.M., Schwartz, R.L., Weischedel, R.M.: An Algorithm that Learns What's in a Name. Machine Learning 34(1-3), 211–231 (1999)
7. Borthwick, A.: Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University (1999)
8. McCallum, A., Li, W.: Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. In: Proceedings of CoNLL (2003)
9. Viterbi, A.J.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Transaction on Information Theory 13(2), 260–267 (1967)
10. Zhou, G., Su, J.: Named Entity Recognition using an HMM-based Chunk Tagger. In: Proceedings of ACL, Philadelphia, pp. 473–480 (2002)
11. Ekbal, A., Bandyopadhyay, S.: Pattern Based Bootstrapping Method for Named Entity Recognition. In: Proceedings of ICAPR-2007, Kolkata, India, pp. 349–355 (2007)
12. Ekbal, A., Bandyopadhyay, S.: Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In: Proceedings of 5th International Conference on Natural Language Processing (ICON), Hyderabad, India, pp. 123–128 (2007)
13. Li, W., McCallum, A.: Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction. ACM Transactions on Asian Language Information Processing (TALIP) 2(3), 290–294 (2003)
14. Brants, T.: TnT a Statistical Parts-of-Speech Tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing ANLP-2000, pp. 224–231 (2000)
15. Ekbal, A., Bandyopadhyay, S.: Lexicon Development and POS Tagging using a Tagged Bengali News Corpus. In: Proceedings of the 20th International Florida AI Research Society Conference (FLAIRS-2007), Florida, pp. 261–263 (2007)
16. Ekbal, A., Mondal, S., Bandyopadhyay, S.: POS Tagging using HMM and Rule-based Chunking. In: Proceedings of the IJCAI Workshop on Shallow Parsing for South Asian Languages, Hyderabad, India, pp. 31–34 (2007)