A Hybrid Approach for Named Entity Recognition in Indian Languages

Abstract

In this paper we describe a hybrid system that applies maximum entropy model (MaxEnt), language specific rules and gazetteers to the task of named entity recognition (NER) in Indian languages designed for the IJCNLP NERSSEAL shared task. Starting with named entity (NE) annotated corpora and a set of features we first build a baseline NER system. Then some language specific rules are added to the system to recognize some specific NE classes. Also we have added some gazetteers and context patterns to the system to increase the performance. As identification of rules and context patterns requires language knowledge, we were able to prepare rules and identify context patterns for Hindi and Bengali only. For the other languages the system uses the MaxEnt model only. After preparing the one-level NER system, we have applid a set of rules to identify the nested entities. The system is able to recognize 12 classes of NEs with 65.13% f-value in Hindi, 65.96% f-value in Bengali and 44.65%, 18.74%, and 35.47% f-value in Oriya, Telugu and Urdu respectively.

1 Introduction

Named entity recognition involves locating and classifying the names in text. NER is an important task, having applications in information extraction (IE), question answering (QA), machine translation and in most other NLP applications.

This paper presents an Hybrid NER system for Indian languages which is designed for the IJCNLP NERSSEAL shared task competition, the goal of which is to perform NE recognition on 12 types of NEs - person, designation, title-Person, organization, abbreviation, brand, title-object, location, time, number, measure and term.

In this work we have identified suitable features for the Hindi NER task. Orthography features, suffix and prefix information, morphology information, part-of-speech information as well as information about the surrounding words and their tags are used to develop a MaxEnt based Hindi NER system. Then we realized that the recognition of some classes will be better if we apply class specific language rules instead of the MaxEnt model. We have defined rules for time, measure and number classes. We made gazetteers based identification for designation, title-person and some terms. Also we have used person and location gazetteers as features of MaxEnt for better identification of these classes. Finally we have build a module for semi-automatic extraction of context patterns and extracted context patterns for person, location, organization and tile-object classes and these are added to the baseline NER system.

The shared task was defined to build the NER systems for 5 Indian languages - Hindi, Bengali, Oriya, Telugu and Urdu for which training data was provided. Among these 5 languages only Bengali and Hindi are known to us but we have no knowledge for other 3 languages. So we are unable to build rules and extract context patterns for these languages. The NER systems for these 3 languages contain only the baseline system i.e. the MaxEnt system. Also

our baseline MaxEnt NER system uses morphological and parts-of-speech (POS) information as a feature. Due to unavailability of morphological analyzer and POS tagger for these 3 languages, these information are not added to the systems. Among the 3 languages, only for Oriya NER system we have used small gazetteers for person, location and designation extracted from the training data. For Bengali and Hindi the developed systems are complete hybrid systems containing rules, gazetteers, context patterns and the MaxEnt model.

The paper is organized as follows. A brief survey of different techniques used for the NER task in different languages and domains are presented in Section 2. Also a brief survey on nested NE recognition systems is presented here. A discussion on the training data is given in Section 3. The MaxEnt based NER system is described in Section 4. Various features used in NER are then discussed. Next we present the experimental results and related discussions in Section 8. Finally Section 9 concludes the paper.

2 Previous Work

A variety of techniques has been used for NER. The two major approaches to NER are:

1. Linguistic approaches.
2. Machine learning (ML) based approaches.

The linguistic approaches typically use rules manually written by linguists. There are several rule-based NER systems, containing mainly lexicalized grammar, gazetteer lists, and list of trigger words, which are capable of providing 88%-92% f-measure accuracy for English (Grishman, 1995; McDonald, 1996; Wakao et al., 1996).

The main disadvantages of these rule-based techniques are that these require huge experience and grammatical knowledge of the particular language or domain and these systems are not transferable to other languages or domains.

ML based techniques for NER make use of a large amount of NE annotated training data to acquire high level language knowledge. Several ML techniques have been successfully used for the NER task of which hidden markov model (Bikel et al., 1997), maximum entropy (Borthwick, 1999), conditional random field (Li and Mccallum, 2004) are most common. Combinations of different ML approaches are also used. Srihari et al. (2000) combines MaxEnt, hidden markov model (HMM) and handcrafted rules to build an NER system.

NER systems use gazetteer lists for identifying names. Both the linguistic approach (Grishman, 1995; Wakao et al., 1996) and the ML based approach (Borthwick, 1999; Srihari et al., 2000) use gazetteer lists.

Linguistic approach uses hand-crafted rules which needs skilled linguistics. Some recent approaches try to learn context patterns through ML which reduce amount of manual labour. Talukder et al.(2006) combined grammatical and statistical techniques to create high precision patterns specific for NE extraction. An approach to lexical pattern learning for Indian languages is described by Ekbal and Bandopadhyay (2007). They used seed data and annotated corpus to find the patterns for NER.

The NER task for Hindi has been explored by Cucerzan and Yarowsky in their language independent NER work which used morphological and contextual evidences (Cucerzan and Yarowsky, 1999). They ran their experiment with 5 languages - Romanian, English, Greek, Turkish and Hindi. Among these the accuracy for Hindi was the worst. For Hindi the system achieved 41.70% f-value with a very low recall of 27.84% and about 85% precision. A more successful Hindi NER system was developed by Wei Li and Andrew Mccallum (2004) using conditional random fields (CRFs) with feature induction. They were able to achieve 71.50% f-value using a training set of size 340k words. In Hindi the maximum accuracy is achieved by (Kumar and Bhattacharyya, 2006). Their maximum entropy markov model (MEMM) based model gives 79.7% f-value.

All the NER systems described above are able to detect one-level NEs. In recent years, the interest in detection of nested NEs has increased. Here we mention few attempts for nested NE detection. Zhou et al. (2004) described an approach to identify cascaded NEs from biomedical texts. They detected the innermost NEs first and then they derived rules to find the other NEs containing these as substrings. Another approach, described by McDonald

et al. (2005), uses structural multilevel classification to deal with overlapping and discontinuous entities. B. Gu (2006) has treated the task of identifying the nested NEs a binary classification problem and solved it using support vector machines. For each token in nested NEs, they used two schemes to set its class label: labeling as the outmost entity or the inner entity.

3 Training Data

The data used for the training of the systems was provided. The annotated data uses shakti standard format (SSF). For our development we have converted the SSF format data into the IOB formatted text in which a $B-XXX$ tag indicates the first word of an entity type XXX and $I-XXX$ is used for subsequent words of an entity. The tag O indicates the word is outside of a NE. The training data for Hindi contains more than 5 lakhs words, for Bengali about 160K words and about 93K, 64K and 36K words for Oriya, Telugu and Urdu respectively.

In time of development we have observed that the training data, provided by the organizers of the shared task, contains several types of errors in NE tagging. These errors in the training corpora affects badly to the machine learning (ML) based models. But we have not made corrections of the errors in the training corpora in time of our development. All the results shown in the paper are obtained using the provided corpora without any modification in NE annotation.

4 Maximum Entropy Based Model

We have used MaxEnt model to build the baseline NER system. MaxEnt is a flexible statistical model which assigns an outcome for each token based on its history and features. Given a set of features and a training corpus, the MaxEnt estimation process produces a model. For our development we have used a Java based open-nlp MaxEnt toolkit[1] to get the probability values of a word belonging to each class. That is, given a sequence of words, the probability of each class is obtained for each word. To find the most probable tag corresponding to each word of a sequence, we can choose the tag having the highest class conditional probability value. But this method is not good as it might result in an inadmissible assignment.

Some tag sequences should never happen. To eliminate these inadmissible sequences we have made some restrictions. Then we used a beam search algorithm with a beam of length 3 with these restrictions.

4.1 Features

MaxEnt makes use of different features for identifying the NEs. Orthographic features (like capitalization, decimal, digits), affixes, left and right context (like previous and next words), NE specific trigger words, gazetteer features, POS and morphological features etc. are generally used for NER. In English and some other languages, capitalization features play an important role as NEs are generally capitalized for these languages. Unfortunately this feature is not applicable for the Indian languages. Also Indian person names are more diverse, lots of common words having other meanings are also used as person names. Li and Mccallum (2004) used the entire word text, character n-grams (n = 2, 3, 4), word prefix and suffix of lengths 2, 3 and 4, and 24 Hindi gazetteer lists as atomic features in their Hindi NER. Kumar and Bhattacharyya (2006) used word features (suffixes, digits, special characters), context features, dictionary features, NE list features etc. in their MEMM based Hindi NER system. In the following we have discussed about the features we have identified and used to develop the Indian language NER systems.

Static Word Feature: The previous and next words of a particular word are used as features. The previous m words $(w_{i-m}...w_{i-1})$ to next n words $(w_{i+1}...w_{i+n})$ can be treated. During our experiment different combinations of previous 4 to next 4 words are used.

Context Lists: Context words are defined as the frequent words present in a word window for a particular class. We compiled a list of the most frequent words that occur within a window of $w_{i-3}...w_{i+3}$ of every NE class. For example, location context list contains the words like '$jAkara$[2]' (going to), '$desha$' (country), '$rAjadhAnI$' (capital) etc. and person context list contains '$kahA$' (say),

[1] www.maxent.sourceforge.net

[2] All Hindi words are written in italics using the 'Itrans' transliteration

'$prdhAnama.ntrI$' (prime minister) etc. For a given word, the value of this feature corresponding to a given NE type is set to 1 if the window $w_{i-3}...w_{i+3}$ around the w_i contains at last one word from this list.

Dynamic NE tag: Named Entity tags of the previous words $(t_{i-m}...t_{i-1})$ are used as features.

First Word: If the token is the first word of a sentence, then this feature is set to 1. Otherwise, it is set to 0.

Contains Digit: If a token 'w' contains digit(s) then the feature $ContainsDigit$ is set to 1.

Numerical Word: For a token 'w' if the word is a numerical word i.e. a word denoting a number (e.g. eka (one), do (two), $tina$ (three) etc.) then the feature $NumWord$ is set to 1.

Word Suffix: Word suffix information is helpful to identify the NEs. Two types of suffix features have been used. Firstly a fixed length word suffix of the current and surrounding words are used as features. Secondly we compiled lists of common suffixes of person and place names in Hindi. For example, '$pura$', '$bAda$', '$nagara$' etc. are location suffixes. We used binary features corresponding to the lists - whether a given word has a suffix from a particular list.

Word Prefix: Prefix information of a word may be also helpful in identifying whether it is a NE. A fixed length word prefix of current and surrounding words are treated as a features.

Root Information of Word: Indian languages are morphologically rich. Words are inflected in various forms depending on its number, tense, person, case etc. Identification of NEs becomes difficult for these inflections. The task becomes easier if instead of the inflected words, corresponding root words are checked whether these are NE or not. For the task we have used morphological analyzers for Hindi and Bengali which are developed at IIT kharagpur.

Parts-of-Speech (POS) Information: The POS of the current word and the surrounding words may be useful feature for NER. We have access to Hindi and Bengali POS taggers developed at IIT Kharagpur which has accuracy about 90%. The tagset of the tagger contains 28 tags. We have used the POS values of the current and surrounding tokens as features.

We realized that the detailed POS tagging is not very relevant. Since NEs are noun phrases, the noun tag is very relevant. Further the postposition following a name may give a clue to the NE type for Hindi. So we decided to use a coarse-grained tagset with only three tags - nominal (Nom), postposition (PSP) and other (O).

The POS information is also used by defining several binary features. An example is the $NomPSP$ binary feature. The value of this feature is defined to be 1 if the current token is nominal and the next token is a PSP.

5 Language Specific Rules

After building of the MaxEnt model we have observed that only a small set of rules are able to identify the classes like number, measure, time, more efficiently than the MaxEnt based model. Then we have tried to define the rules for these classes. The rule identification is done manually and requires language knowledge. We have defined the required rules for Bengali and Hindi but we are unable to do the same for other 3 languages as the languages are unknown to us. In the following we have mentioned some example rules which are defined and used in our system.

- IF ((W_i is number or numeric word) AND (W_{i+1} is an unit))
  THEN (W_i W_{i+1}) bigram is a $measure$ NE.
- IF ((W_i is number or numeric word) AND (W_{i+1} is a month-name) AND (W_{i+2} is a 4 digit number))
  THEN (W_i W_{i+1} W_{i+2}) trigram is a $time$ NE.
- IF ((W_i denotes a day of a week) AND (W_{i+1} number or numeric word) AND (W_{i+2} is a month name))
  THEN (W_i W_{i+1} W_{i+2}) trigram is a $time$ NE.

We have defined 36 rules in total for time, measure and number classes. These rules use some lists which are built. These lists contain corresponding entries both in the target language and in English. For example the months names list contains the names according to the English calender and the names according to the Indian calender. In the following we have mentioned the lists we have prepared for the rule-based module.

- Names of months.
- Names of seasons.
- Days of a week.
- Names of units.
- Numerical words.

5.1 Semi-automatic Extraction of Context Patterns

Similar to the rules defined for time, measure and date classes, if efficient context patterns (CP) can be extracted for a particular class, these can help in identification of NEs of the corresponding class. But extraction of CP requires huge labour if done manually. We have developed a module for semi-automatically extraction of context patterns. This module makes use of the most frequent entities of a particular class as $seed$ for that class and finds the surrounding tokens of the $seed$ to extract effective patterns. We mark a pattern as 'effective' if the precision of the pattern is very high. Precision of a pattern is defined as the ratio of correct identification and the total identification when the pattern is used to identify NEs of a particular type from a text.

For our task we have extracted patterns for person, location, organization and title-object classes. These patterns are able to identify the NEs of a specific classes but detection of NE boundary is not done properly by the patterns. For boundary detection we have added some heuristics and used POS information of the surrounding words. The patterns for a particular class may identify the NEs of other classes also. For example the patterns for identifying person names may also identify the designation or title-persons. These needs to be handled carefully in time of using the patterns. In the following some example patterns are listed which are able to identify person names for Hindi.

- <PER> ne kahA ki
- <PER> kA kathana he.n
- mukhyama.ntrI <PER> Aja
- <PER> ne apane gra.ntha
- <PER> ke putra <PER>

6 Use of Gazetteer Lists

Lists of names of various types are helpful in name identification. Firstly we have prepared the lists using the training corpus. But these are not sufficient. Then we have compiled some specialized name lists from different web sources. But the names in these lists are in English, not in Indian languages. So we have transliterated these English name lists to make them useful for our NER task.

Using transliteration we have constructed several lists. Which are, month name and days of the week, list of common locations, location names list, first names list, middle names list, surnames list.

The lists can be used in name identification in various ways. One way is to check whether a token is in any list. But this approach is not good as it has some limitations. Some words may present in two or more gazetteer lists. Confusions arise to make decisions for these words. Some words are in gazetteer lists but sometimes these are used in text as not-name entity. We have used these gazetteer lists as features of MaxEnt. We have prepared several binary features which are defined as whether a given word is in a particular list.

7 Detection of Nested Entities

The training corpora used for the models, are not annotated as nested. The maximal entities are annotated in the training corpus. For detection of the nested NEs, we have derived some rules. For example, if a particular word is a number or numeric word and is a part of a NE type other than 'number', then we have made the nesting. Again, if any common location identifier word like, $jilA$ (district), $shahara$ (town) etc. is a part of a 'location' entity then we have nested there. During one-level NE identification, we have generated lists for all the identified location and person names. Then we have searched other NEs containing these as substring to make the nesting. After preparing the one-level NER system, we have applied the derived rules on it to identify the nested entities.

8 Evaluation

The accuracies of the system are measured in terms of the f-measure, which is the weighted harmonic mean of precision and recall. Nested, maximal and

lexical accuracies are calculated separately. The test data for all the five languages are provided. The size of the shared task test files are: Hindi - 38,704 words, Bengali - 32,796 words, Oriya - 26,988 words, Telugu - 7,076 words and Urdu - 12,805 words.

We have already mentioned that after preparing a one-level NER system, the rule-based module is used to modify it to a nested one. A number of experiments are conducted considering various combinations of features to identify the best feature set for Indian language NER task. It is very difficult and time consuming to conduct experiments for all the languages. During the development we have conducted all the experiments on Hindi and Bengali. We have prepared a development test data composed of 24,265 words for Hindi and 10,902 word for Bengali and accuracies of the system are tested on the development data. The details of the experiments on hindi data for the best feature selection is described in the following section.

8.1 Best Feature Set Selection

The performance of the system on the Hindi data using various features are presented in Table 1. They are summarized below. While experimenting with static word features, we have observed that a window of previous two words to next two words ($W_{i-2}...W_{i+2}$) gives best results. But when several other features are combined then smaller window ($W_{i-1}...W_{i+1}$) performs better. Similarly we have experimented with suffixes of different lengths and observed that the suffixes of length ≤ 2 gives the best result for the Hindi NER task. In using POS information, we have observed that the coarse-grained POS tagger information is more effective than the finer-grained POS values. The most interesting fact we have observed that more complex features do not guarantee to achieve better results. For example, a feature set combined with current and surrounding words, previous NE tag and fixed length suffix information, gives a f-value 64.17%. But when prefix information are added the f-value decreased to 63.73%. Again when the context lists are added to the feature set containing words, previous tags, suffix information, digit information and the NomPSP binary feature, the accuracy has decreased to 67.33% from 68.0%.

Feature	**Overall F-value**
Word, NE Tag	58.92
Word, NE Tag, Suffix (≤ 2)	64.17
Word, NE Tag, Suffix (≤ 2), Prefix	63.73
Word, NE Tag, Digit, Suffix	66.61
Word, NE Tag, Context List	63.57
Word, NE Tag, POS (full)	61.28
Word, NE Tag, Suffix (≤ 2), Digit, NomPSP	68.60
Word, NE Tag, Suffix (≤ 2), Digit, Context List, NomPSP	67.33
Word, NE Tag, Suffix (≤ 2), Digit, NomPSP, Linguistic Rules	73.40
Word, NE Tag, Suffix (≤ 2), Digit, NomPSP, Gazetteers	72.08
Word, NE Tag, Suffix (≤ 2), Digit, NomPSP, Linguistic Rules, Gazetteers	74.53

Table 1: Hindi development set f-values for different features

The feature set containing words, previous tags, suffix information, digit information and the NomPSP binary feature is the identified best feature set without linguistic rules and gazetteer information. Then we have added the linguistic rules, patterns and gazetteer information to the system and the changes in accuracies are shown in the table.

8.2 Results on the Test Data

The best identified feature set is used for the development of the NER systems for all the five languages. We have already mentioned that for only for Hindi and Bengali we have added linguistic rules and gazetteer lists in the MaxEnt based NER systems. The accuracy of the system on the shared task test data for all the languages are shown in Table 2.

The accuracies of Oriya, Urdu and Telugu languages are poor compared to the other two languages. The reasons are POS information, morphological information, language specific rules and gazetteers are not used for these languages. Also the size of training data for these languages are smaller.

Language	Type	Precision	Recall	F-measure
Hindi	Maximal	75.19	58.94	66.08
	Nested	79.58	58.61	67.50
	Lexical	82.76	53.69	65.13
Bengali	Maximal	52.92	68.07	59.54
	Nested	55.02	68.43	60.99
	Lexical	62.30	70.07	65.96
Oriya	Maximal	21.17	26.92	23.70
	Nested	27.73	28.13	27.93
	Lexical	51.51	39.40	44.65
Urdu	Maximal	26.12	29.69	27.79
	Nested	27.99	29.21	28.59
	Lexical	37.58	33.58	35.47
Telugu	Maximal	10.47	9.64	10.04
	Nested	22.05	13.16	16.48
	Lexical	25.23	14.91	18.74

Table 2: Accuracy of the system for all languages

Class	Hindi		Bengali	
	Maximal	Nested	Maximal	Nested
Person	70.87	71.00	77.45	79.09
Designation	48.98	59.81	26.32	26.32
Organization	47.22	47.22	41.43	71.43
Abbreviation	-	72.73	51.61	51.61
Brand	-	-	-	-
Title-person	-	60.00	5.19	47.61
Title-object	41.32	40.98	72.97	72.97
Location	86.02	87.02	76.27	76.27
Time	67.42	67.42	56.30	56.30
Number	84.59	85.13	40.65	40.65
Measure	59.26	55.17	62.50	62.50
Term	48.91	50.51	43.67	43.67

Table 3: Comparison of maximal and nested f-values for different classes of Hindi and Bengali

To mention, for Urdu, size of the training data is only about 36K words which is very small to train a MaxEnt model.

It is mentioned that we have prepared a set of rules which are capable of identifying the nested NEs. Once the one-level NER system has built, we have applied the rules on it. In Table 3 we have shown the f-values of each class after addition of the nested rules. The detailed results for all languages are not shown. In the table we have shown only the results of Hindi and Bengali.

For both the languages 'title-person' and 'designation' classes are suffering from poor accuracies. The reason is, in the training data and also in the annotated test data, these classes contains many annotation errors. Also the classes being closely related to each other, the system fails to distinguish them properly. The detection of the 'term' class is very difficult. In the test files amount of 'term' entity is large, for Bengali - 434 and for Hindi - 1080, so the poor accuracy of the class affects badly to the overall accuracy. We have made rule-based identification for 'number', 'measure' and 'time' classes; the accuracies of these classes proves that the rules need to be modified to achieve better accuracy for these classes. Also the accuracy of the 'organization' class is not high, because amount of organization entities is not sufficient in the training corpus. We have achieved good results for other two main classes - 'person' and 'location'.

9 Conclusion

We have prepared a MaxEnt based system for the NER task in Indian languages. We have also added rules and gazetteers for Bengali and Hindi. Also our derived rules need to be modified for improvement of the system. We have not make use of rules and gazetteers for Oriya, Telugu and Urdu. As the size of training data is not much for these 3 languages, rules and gazetteers would be effective. We have experimented with MaxEnt model only, other ML methods like HMM, CRF or MEMM may be able to give better accuracy. We have not worked much on the detection of nested NEs. Proper detection of nested entities may lead to further improvement of performance and is under investigation.

References

Bikel Daniel M., Miller Scott, Schwartz Richard and Weischedel Ralph. 1997. Nymble: A High Perfor-

mance Learning Name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 194–201.

Borthwick Andrew. 1999. A Maximum Entropy Approach to Named Entity Recognition. *Ph.D. thesis, Computer Science Department, New York University*.

Cucerzan Silviu and Yarowsky David. 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Proceedings of the Joint SIGDAT Conference on EMNLP and VLC 1999*, 90–99.

Ekbal A. and Bandyopadhyay S. 2007. Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In *Proceedings of International Conference on Natural Language Processing (ICON), 2007.*

Grishman Ralph. 1995. The New York University System MUC-6 or Where's the syntax? In *Proceedings of the Sixth Message Understanding Conference*.

Gu B. 2006. Recognizing Nested Named Entities in GENIA corpus. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06*, pages 112-113.

Kumar N. and Bhattacharyya Pushpak. 2006. Named Entity Recognition in Hindi using MEMM. In *Technical Report, IIT Bombay, India.*.

Li Wei and McCallum Andrew. 2004. Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction (Short Paper). In *ACM Transactions on Computational Logic*.

McDonald D. 1996. Internal and external evidence in the identification and semantic categorization of proper names. In *B. Boguraev and J. Pustejovsky, editors, Corpus Processing for Lexical Acquisition*, 21–39.

McDonald R., Crammer K. and Pereira F. 2005. Flexible text segmentation with structured multilabel classification. In *Proceedings of EMNLP05*.

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*.

Talukdar Pratim P., Brants T., Liberman M., and Pereira F. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*.

Wakao T., Gaizauskas R. and Wilks Y. 1996. Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of COLING-96*.

Zhou G., Zhang J., Su J., Shen D. and Tan C. 2004. Recognizing Names in Biomedical Texts: a Machine Learning Approach. *Bioinformatics*, 20(7):1178-1190.