Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition

Karthik Gali, Harshit Surana, Ashwini Vaidya, Praneeth Shishtla and Dipti Misra Sharma
Language Technologies Research Centre,
International Institute of Information Technology,
Hyderabad, India.
karthikg@students.iiit.ac.in, surana.h@gmail.com,
ashwini_vaidya@research.iiit.ac.in, praneethms@students.iiit.ac.in,
dipti@iiit.ac.in

Abstract

This paper, submitted as an entry for the NERSSEAL-2008 shared task, describes a system build for Named Entity Recognition for South and South East Asian Languages. Our paper combines machine learning techniques with language specific heuristics to model the problem of NER for Indian languages. The system has been tested on five languages: Telugu, Hindi, Bengali, Urdu and Oriya. It uses CRF (Conditional Random Fields) based machine learning, followed by post processing which involves using some heuristics or rules. The system is specifically tuned for Hindi and Telugu, we also report the results for the other four languages.

1 Introduction

Named Entity Recognition (NER) is a task that seeks to locate and classify entities ('atomic elements') in a text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, etc. It can be viewed as a two stage process:

1. Identification of entity boundaries
2. Classification into the correct category

For example, if "Mahatma Gandhi" is a named entity in the corpus, it is necessary to identify the beginning and the end of this entity in the sentence. Following this step, the entity must be classified into the predefined category, which is NEP (Named Entity Person) in this case.

This task is the precursor for many natural language processing applications. It has been used in Question Answering (Toral et al, 2005) as well as Machine Translation (Babych et al, 2004).

The NERSSEAL contest has used 12 categories of named entities to define a tagset. The data has been manually tagged for training and testing purposes for the contestants.

The task of building a named entity recognizer for South and South East Asian languages presents several problems related to their linguistic characteristics. We will first discuss some of these linguistic issues, followed by a description of the method used. Further, we show some of the heuristics used for post-processing and finally an analysis of the results obtained.

2 Previous Work

The linguistic methods generally 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 upto 92% f-measure accuracy for English (McDonald, 1996; Wakao et al., 1996).

Linguistic approach uses hand-crafted rules which need skilled linguistics. The chief disadvantage of these rule-based techniques is that they require huge experience and grammatical knowledge of the particular language or domain and these systems are not transferable to other languages or domains. However, given the closer nature of many Indian languages, the cost of adaptation of a re-

Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, pages 25–32,
Hyderabad, India, January 2008.

source from one language to another could be quite less (Singh and Surana, 2007).

Various machine learning techniques have also been successfully used for the NER task. Generally hidden markov model (Bikel et al.,1997), maximum entropy (Borthwick, 1999), conditional random field (Li and Mccallum, 2004) are more popular machine learning techniques used for the purpose of NER.

Hybrid systems have been generally more effective at the task of NER. Given lesser data and more complex NE classes which were present in NERSSEAL shared task, hybrid systems make more sense. Srihari et al. (2000) combines MaxEnt, hidden markov model (HMM) and handcrafted rules to build an NER system.

Though not much work has been done for other South Asian languages, some previous work focuses on NER for Hindi. It has been previously attempted 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 different languages. Among these the accuracy for Hindi was the worst. For Hindi the system achieved 42% f-value with a recall of 28% and about 85% precision. A result which highlights lack of good training data, and other various issues involved with linguistic handling of Indian languages.

Later approaches have resulted in better results for Hindi. Hindi NER system developed by Wei Li and Andrew Mccallum (2004) using conditional random fields (CRFs) with feature induction have achieved f-value of 71%. (Kumar and Bhattacharyya, 2006) used maximum entropy markov model to achieve f-value of upto 80%.

3 Some Linguistic Issues

3.1 Agglutinative Nature

Some of the SSEA languages have agglutinative properties. For example, a Dravidian language like Telugu has a number of postpositions attached to a stem to form a single word. An example is:

guruvAraMwo = guruvAraM + wo
up to Wednesday = Wednesday + up to

Most of the NERs are suffixed with several different postpositions, which increase the number of distinct words in the corpus. This in turn affects the machine learning process.

3.2 No Capitalization

All the five languages have scripts without graphical cues like capitalization, which could act as an important indicator for NER. For a language like English, the NER system can exploit this feature to its advantage.

3.3 Ambiguity

One of the properties of the named entities in these languages is the high overlap between common names and proper names. For instance *Kamal* (in Hindi) can mean 'lotus', which is not a named entity, but it can also be a person's name, in which case, it is a named entity.

Among the named entities themselves, there is ambiguity between a location name ***Bangalore*** *ek badzA shaher heI* (Bangalore is a big city) or a person's surname '*M.* ***Bangalore*** *shikshak heI*' (M. Bangalore is a teacher).

3.4 Low POS Tagging Accuracy for Nouns

For English, the available tools like POS (Part of Speech) tagger can be used to provide features for machine learning. This is not very helpful for SSEA languages because the accuracy for noun and proper noun tags is quite low (PVS and G., 2006) Hence, features based on POS tags cannot be used for NER for these languages.

To illustrate this difficulty, we conducted the following experiment. A POS tagger (described in PVS & G.,2006) was run on the Hindi test data. The data had 544 tokens with NEL, NEP, NEO tags. The POS tagger should have given the NNP (proper noun) tag for all those named entities. However the tagger was able to tag only 80 tokens accurately. This meant that only 14.7% of the named entities were correctly recognized.

3.5 Spelling Variation

One other important language related issue is the variation in the spellings of proper names. For instance the same name *Shri Ram Dixit* can be written as *Sri. Ram Dixit*, *Shree Ram Dixit*, *Sh. R. Dixit* and so on. This increases the number of tokens to be learnt by the machine and would perhaps also require a higher level task like co-reference resolution.

2.6 Pattern of suffixes

Named entities of Location (NEL) or Person (NEP) will share certain common suffixes, which can be exploited by the learning algorithm. For instance, in Hindi, ***-pur*** (*Rampur*, *Manipur*) or ***-giri*** (*Devgiri*) are suffixes that will appear in the named entities for Location. Similarly, there are suffixes like ***-swamy*** (*Ramaswamy*, *Krishnaswamy*) or ***-deva*** (*Vasudeva*, *Mahadeva*) which can be commonly found in named entities for person. These suffixes are cues for some of the named entities in the SSEA languages.

A NER system can be rule-based, statistical or hybrid. A rule-based NER system uses handwritten rules to tag a corpus with named entities. A statistical NER system learns the probabilities of named entities using training data, whereas hybrid systems use both.

Developing rule-based taggers for NER can be cumbersome as it is a language specific process. Statistical taggers require large amount of annotated data (the more the merrier) to train. Our system is a hybrid NER tagger which first uses Conditional Random Fields (CRF) as a machine learning technique followed by some rule based post-processing.

We treat the named entity recognition problem as a sequential token-based tagging problem.

According to Lafferty et. al. CRF outperforms other Machine Learning algorithms viz., Hidden Markov Models (HMM), Maximum Entropy Markov Model (MEMM) for sequence labeling tasks.

4 Training data

The training data given by the organizers was in SSF format[1]. For example in SSF format, the named entity 'Rabindranath Tagore' will be shown in the following way:

```
0       ((      SSF
1       ((      NP      <ne=NEP>
1.1     Rabindranath
1.2     Tagore
        ))
2       ne
3       kahaa
        ))
```

1 http://shiva.iiit.ac.in/SPSAL2007/ssf-analysis-representation.pdf

We have converted this format into the BIO format as described in Ramshaw et. al. For example, the above format will now be shown as:

Rabindranath	B-NEP
Tagore	I-NEP
ne	O
kahaa	O

The training data set contains (approximately) 400,000 Hindi, 50,000 Telugu, 35,000 Urdu, 93,000 Oriya and 120,000 Bengali words respectively.

5 Conditional Random Fields

Conditional Random Fields (CRFs) are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes.

In the special case in which the output nodes of the graphical model are linked by edges in a linear chain, CRFs make a first-order Markov independence assumption, and thus can be understood as conditionally-trained finite state machines (FSMs). Let $o = (o,,o_2,o_3 ,o_4 ,... o_T)$ be some observed input data sequence, such as a sequence of words in text in a document,(the values on n input nodes of the graphical model). Let S be a set of FSM states, each of which is associated with a label, l € £.

Let $s = (s_1,s_2,s_3 ,s_4 ,... s_T)$ be some sequence of states, (the values on T output nodes). By the Hammersley-Clifford theorem, CRFs define the conditional probability of a state sequence given an input sequence to be:

$$P(s|o) = \frac{1}{Z_o} * exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k (s_{t-1}, s_t, o, t))$$

where Z_o is a normalization factor over all state sequences is an arbitrary feature function over its arguments, and λ_k is a learned weight for each feature function. A feature function may, for example, be defined to have value 0 or 1. Higher λ weights make their corresponding FSM transitions more likely. CRFs define the conditional probability of a label sequence based on the total probability over the state sequences,

$$P(l|o) = \sum_{s:l(s)=l} P(s|o)$$

	Precision			Recall			F-Measure		
	Pm	**Pn**	**Pl**	**Rm**	**Rn**	**Rl**	**Fm**	**Fn**	**Fl**
Bengali	53.34	49.28	58.27	26.77	25.88	31.19	35.65	33.94	40.63
Hindi	59.53	63.84	64.84	41.21	41.74	40.77	48.71	50.47	50.06
Oriya	39.16	40.38	63.70	23.39	19.24	28.15	29.29	26.06	39.04
Telugu	10.31	71.96	65.45	68.00	30.85	29.78	08.19	43.19	40.94
Urdu	43.63	44.76	48.96	36.69	34.56	39.07	39.86	39.01	43.46

Table 1: Evaluation of the NER System for Five Languages

where *l(s)* is the sequence of labels corresponding to the labels of the states in sequence s.

Note that the normalization factor, Z_o, (also known in statistical physics as the partition function) is the sum of the scores of all possible states.

$$Z_o = \sum_{s \in S^T} \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, o, t) \right),$$

And that the number of state sequences is exponential in the input sequence length, T. In arbitrarily-structured CRFs, calculating the partition function in closed form is intractable, and approximation methods such as Gibbs sampling or loopy belief propagation must be used. In linear-chain structured CRFs (in use here for sequence modeling), the partition function can be calculated efficiently by dynamic programming.

6 CRF Based Machine Learning

We used the CRF model to perform the initial tagging followed by post-processing.

6.1 Statistical Tagging

In the first phase, we have used language independent features to build the model using CRF. Orthographic features (like capitalization, decimals), affixes (suffixes and prefixes), context (previous words and following 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 as explained above this feature is not applicable for the Indian languages.

The exact set of features used are described below.

6.2 Window of the Words

Words preceding or following the target word may be useful for determining its category. Following a few trials we found that a suitable window size is five.

6.3 Suffixes

Statistical suffixes of length 1 to 4 have been considered. These can capture information for named entities having the NEL tag like *Hyderabad*, *Secunderabad*, *Ahmedabad* etc., all of which end in *-bad*. We have collected lists of such suffixes for NEP (Named Entity Person) and NEL (Named Entity Location) for Hindi. In the machine learning model, this resource can be used as a binary feature. A sample of these lists is as follows:

Type of NE	**Example suffixes (Hindi)**
NE- Location	-desa, -vana, -nagara, -garh, -rashtra, -giri
NE – Person	-raja, -natha, -lal, -bhai,-pathi, -krishnan

Table 2: Suffixes for Hindi NER

6.4 Prefixes

Statistical prefixes of length 1 to 4 have been considered. These can take care of the problems associated with a large number of distinct tokens. As mentioned earlier, agglutinative languages can have a number of postpositions. The use of prefixes will increase the probability of *Hyderabad* and *Hyderabadlo* (Telugu for 'in Hyderabad') being treated as the same token.

	Bengali	Hindi	Oriya	Telugu	Urdu
NEP	35.22	54.05	52.22	01.93	31.22
NED	NA	42.47	01.97	NA	21.27
NEO	11.59	45.63	14.50	NA	19.13
NEA	NA	61.53	NA	NA	NA
NEB	NA	NA	NA	NA	NA
NETP	42.30	NA	NA	NA	NA
NETO	33.33	13.77	NA	01.66	NA
NEL	45.27	62.66	48.72	01.49	57.85
NETI	55.85	79.09	40.91	71.35	63.47
NEN	62.67	80.69	24.94	83.17	13.75
NEM	60.51	43.75	19.00	26.66	84.10
NETE	19.17	31.52	NA	08.91	NA

Table 3: F-Measure (Lexical) for NE Tags

6.5 Start of a sentence

There is a possibility of confusing the NEN (Named Entity Number) in a sentence with the number that appears in a numbered list. The numbered list will always have numbers at the beginning of a sentence and hence a feature that checks for this property will resolve the ambiguity with an actual NEN.

6.6 Presence of digits

Usually, the presence of digits indicates that the token is a named entity. For example, the tokens *92, 10.1* will be identified as Named Entity Number based on the binary feature 'contains digits'.

6.7 Presence of four digits

If the token is a four digit number, it is likelier to be a NETI (Named Entity Time). For example, 1857, 2007 etc. are most probably years.

7 Heuristics Based Post Processing

Complex named entities like *fifty five kilograms* contain a Named Entity Number within a Named Entity Measure. We observed that these were not identified accurately enough in the machine learning based system. Hence, instead of applying machine learning to handle nested entities we make use of rule-based post processing.

7.1 Second Best Tag

It was observed that the recall of the CRF model is low. In order to improve recall, we have used the following rule: if the best tag given by the CRF model is O (not a named entity) and the confidence of the second best tag is greater than 0.15, then the second best tag is considered as the correct tag.

We observed an increase of 7% in recall and 3% decrease in precision. This resulted in a 4% increase in the F-measure, which is a significant increase in performance. The decrease in precision is expected as we are taking the second tag.

7.2 Nested Entities

One of the important tasks in the contest was to identify nested named entities. For example if we consider *eka kilo* (Hindi: one kilo) as NEM (Named Entity Measure), it contains a NEN (Named Entity Number) within it.

The CRF model tags *eka kilo* as NEM and in order to tag *eka* as NEN we have made use of other resources like a gazetteer for the list of numbers. We used such lists for four languages.

7.3 Gazetteers

For Hindi, we made use of three different kinds of gazetteers. These consisted of lists for measures (entities like kilogram, millimetre, lakh), numerals and quantifiers (one, first, second) and time expressions (January, minutes, hours) etc. Similar lists were used for all the other languages except Urdu. These gazetteers were effective in identifying this relatively closed class of named entities and showed good results for these languages.

8 Evaluation

The evaluation measures used for all the five languages are precision, recall and F-measure. These measures are calculated in three different ways:

1. **Maximal Matches**: The largest possible named entities are matched with the reference data.
2. **Nested Matches**: The largest possible as well as nested named entities are matched.
3. **Lexical Item Matches**: The lexical items inside largest possible named entities are matched.

9 Results

The results of evaluation as explained in the previous section are shown in the Table-1. The F-measures for nested lexical match are also shown individually for each named entity tag separately in Table-3

10 Unknown Words

Table 4 shows the number of unknown words present in the test data when compared with the training data.

First column shows the number of unique Named entity tags present in the test data for each language. Second column shows the number of unique known named entities present in the test data. Third column shows the percentage of unique unknown words present in the test data of different languages when compared to training data.

11 Error Analysis

We can observe from the results that the maximal F-measure for Telugu is very low when compared to lexical F-measure and nested F-measure. The reason is that the test data of Telugu contains a large number of long named entities (around 6 words), which in turn contain around 4 - 5 nested named entities. Our system was able to tag nested named entities correctly unlike maximal named entity.

We can also observe that the maximal F-measure for Telugu is very low when compared to other languages. This is because Telugu test data has very few known words.

Urdu results are comparatively low chiefly because gazetteers for numbers and measures were unavailable.

The amount of annotated corpus available for Hindi was substantially more. This should have ideally resulted in better results for Hindi with the machine learning approach. But, the results were only marginally better than other languages. A major reason for this was that a very high percentage (44%) of tags in Hindi were NETE. The tagset gives examples like 'Horticulture', 'Conditional Random Fields' for the tag NETE. It has also been mentioned that even manual annotation is harder for NETE as it is domain specific. This affected the overall results for Hindi because the performance for NETE was low (Table 3).

	Num of NE tokens	***Num of known NE***	***% of unknown NE***
Bengali	1185	277	23.37
Hindi	1120	417	37.23
Oriya	1310	563	42.97
Telugu	1150	145	12.60
Urdu	631	179	28.36

Table 4: Unknown Word

Also, the F-measures of NEN, NETI, and NEM could have been higher because they are relatively closed classes. However, certain NEN can be ambiguous (Example: *eka* is a NEN for 'one' in Hindi, but in a different context it can be a non-number. For instance *eka-doosra* is Hindi for 'each other').

In a language like Telugu, NENs will appear as inflected words. For example 2*001lo, guruvaaramto.*

10 Conclusion and Further Work

In this paper we have presented the results of using a two stage hybrid approach for the task of named entity recognition for South and South East Asian Languages. We have achieved decent Lexical F-measures of 40.63, 50.06, 39.04, 40.94, and 43.46 for Bengali, Hindi, Oriya, Telugu and Urdu respectively without using many language specific resources.

We plan to extend our work by applying our method to other South Asian languages, and by using more language specific constraints and resources. We also plan to incorporate semi-supervised extraction of rules for NEs (Saha et. al,

2008) and use transliteration techniques to produce Indian language gazetteers (Surana and Singh, 2008). Use of character models for increasing the lower recalls (Shishtla et. al, 2008) is also underway. We also plan to enrich the Indian dependency tree bank (Begum et. al, 2008) by use of our NER system.

11 Acknowledgments

We would like to thank the organizer Mr. Anil Kumar Singh deeply for his continuous support during the shared task.

References

B. Babych, and A. Hartley, Improving Machine translation Quality with Automatic Named Entity Recognition. www.mt-archive.info/EAMT-2003- Babych.pdf

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for Indian languages. In *Proceedings of IJCNLP-2008*, Hyderabad, India.

M. Bikel Daniel, 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*.

S. Cucerzan, and D. Yarowsky, 1999. Language independent named entity recognition combining morphological and contextual evidence. *Proceedings of the Joint SIGDAT Conference on EMNLP and VLC.*

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

John Lafferty, Andrew McCallum and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc. 18th International Conf. on Machine Learning*.

D. McDonald 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.*

Avinesh PVS and Karthik G. Part-Of-Speech Tagging and Chunking Using Conditional Random Fields and Transformation Based Learning. *Proceedings of the SPSAL workshop during IJCAI'07*.

Lance Ramshaw and Mitch Marcus. Text Chunking Using Transformation-Based Learning. Proceedings of the Third Workshop on Very Large Corpora.

S.K. Saha , S. Chatterji , S. Dandapat , S. Sarkar and P. Mitra 2008. A Hybrid Approach for Named Entity Recognition in Indian Languages. *In Proceedings of IJCNLP Workshop on NER for South and South East Asian Languages.*

Fei Sha and Fernando Pereira. 2003. Shallow Parsing with Conditional Random Fields. *In the Proceedings of HLT-NAACL*.

P. Shishtla, P. Pingali , V. Varma 2008. A Character n-gram Based Approach for Improved Recall in Indian Language NER. *In Proceedings of IJCNLP Workshop on NER for South and South East Asian Languages.*

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.*

A. K. Singh and H. Surana Can Corpus Based Measures be Used for Comparative Study of Languages? *In Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*. ACL. 2007.

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

H. Surana and A. K. Singh 2008. A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages. *In Proceedings of the Third International Joint Conference on Natural Language Processing.*

Charles Sutton, *An Introduction to Conditional Random Fields for Relational Learning*.

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

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

CRF++:.Yet another Toolkit.
http://crfpp.sourceforge.net/