A Hybrid Named Entity Recognition System for South Asian Languages

Praveen Kumar P
Language Technologies Research Centre
International Institute of Information
Technology - Hyderabad
praveen_p@students.iiit.ac.in

Ravi Kiran V
Language Technologies Research Centre
International Institute of Information
Technology - Hyderabad
ravikiranv@students.iiit.ac.in

Abstract

This paper is submitted for the contest NERSSEAL-2008. Building a statistical based Named entity Recognition (NER) system requires huge data set. A rule based system needs linguistic analysis to formulate rules. Enriching the language specific rules can give better results than the statistical methods of named entity recognition. A Hybrid model proved to be better in identifying Named Entities (NE) in Indian Language where the task of identifying named entities is far more complicated compared to English because of variation in the lexical and grammatical features of Indian languages.

1 Introduction

Named Entities (NE) are phrases that contain person, organization, location, number, time, measure etc. Named Entity Recognition is the task of identifying and classifying the Named Entities into predefine categories such as person, organization, location, etc in the text.

NER has several applications. Some of them are Machine Translation (MT), Question-Answering System, Information Retrieval (IR), and Cross-lingual Information Retrieval.

The tag set used in the NER-SSEA contest has12 categories. This is 4 more than the CONLL-2003 shared task on NER tag-set. The use of finer tag-set aims at improving Machine Translation (MT). Annotated data for Hindi, Bengali, Oriya, Telugu and Urdu languages was provided to the contestants.

Significant work in the field of NER was done in English, European languages but not in Indian languages. There are many rule-based, HMM based; Conditional Random Fields (CRF) based NER systems. MEMM were used to identify the NE in Hindi (Kumar and Bhattacharyya, 2006). Many techniques were used in CoNLL-2002 shared task on NER which aimed at developing a language independent NER system.

2 Issues: Indian Languages

The task of NER in Indian Languages is a difficult task when compared to English. Some features that make the task difficult are

2.1 No Capitalization

Capitalization is an important feature used by the English NER systems to identify the NE. The absence of the lexical features such as capitalization in Indian languages scripts makes it difficult to identify the NE.

2.2 Agglutinative nature

Some of the Indian language such as Telugu is agglutinative in nature. Telugu allows polyagglutination, the unique feature to being able to add multiple suffixes to words to denote more complex words.
Ex: “hyderabadlonunci” = hyderabad+ lo + nunchi

2.3 Ambiguities

There can be ambiguity among the names of persons, locations and organizations such as *Washington* can be either a person name as well as location name.

2.4 Proper-noun & common noun Ambiguity

In India the common-nouns often occur as the person names. For instance *Akash* which can mean ‘*sky*’ is also name of a person.

Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, pages 83–88,
Hyderabad, India, January 2008. ©2008 Asian Federation of Natural Language Processing

2.5 Free-word order

Some of the Indian languages such as Telugu are free word order languages. The heuristics such as position of the word in the sentence can not be used as a feature to identify NE in these languages.

3 Approaches

A NER system can be either a Rule based or statistical or hybrid. A Rule-based system needs linguistic analysis to formulate the rules. A statistical NER system needs annotated corpus. A hybrid system is generally a rule based system on top of statistical system.

For the NER-SSEAL contest we developed CRF based and HMM based hybrid system.

3.1 Hidden Markov Model

We used a second order Markov model for Named entity tagging. The tags are represented by the states, words by the output. Transition probabilities depend on the states. Output probabilities depend on the most recent category. For a given sentence $w_1 \ldots w_T$ of length T. $t_1,t_2 .. t_T$ are elements of the tag-set. We calculate

$$\text{Argmax}_{t1\ldots tT} [\Pi_1^T P(t_i|t_{i-1},t_{i-2})P(w_i|t_i)](P(t_{T+1}|t_T)$$

This gives the tags for the words. We use linear interpolation of unigrams, bigrams and trigrams for transition probability smoothing and suffix trees for emission probability smoothing.

3.1.1 HMM based hybrid model

In the first phase HMM models are trained on the training corpus and are used to tag the test data.

The first layer is purely statistical method of solving and the second layer is pure rule based method of solving. In order to extend the tool for any other Indian language we need to formulate rules in the second layer. In the first layers HMM models are training from the annotated training corpus. The annotation follows as: Every word in the corpus if belongs to any Named entity class is marked with the corresponding class name. And the one's which don't fall into any of the named entity class fall into the class of words that are not named entities. The models obtained by training the annotated training corpus are used to tag the test data. In the first layer the class boundaries may not be identified correctly. This problem of correctly identifying the class boundaries and nesting is solved in the second layer.

In the second layer, the chunk information of the test corpus is used to identify the correct boundaries of the named entities identified from the first layer. It's a type of validation of result from the first layer. Simultaneously, few rules for every class of named entities are used in order to identify nesting of named entities in the chunks and to identify the unidentified named entities from the first layer output. For Telugu these rules include suffixes with which Named Entities can be identified

3.2 Conditional Random Fields

Conditional Random Fields (CRFs) are undirected graphical models, a special case of which corresponds to conditionally-trained finite state machines. CRFs are used for labeling sequential data.

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 $o = (o, o_2, o_3, o_4, \ldots o_T)$ 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 $s = (s_1,s_2,s_3 ,s_4 ,\ldots s_T)$ 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 $?_k$ is a learned weight for each feature function. A feature function may, for example, be defined to have value 0 or 1. Higher ? 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,

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

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

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

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.

And that the number of state sequences is exponential in the input sequence length T. In arbitrarily structured CRF's calculating the normalization factor in closed form is intractable, but in liner-chain-structure CRFs, the probability that a particular transition was taken between two CRF states at a particular position in the input can be calculated by dynamic programming.

3.2.1 CRF based model

CRF models were used to perform the initial tagging. The features for the Hindi and Telugu models include the Root, number and gender of the word from the morphological analyzer. From our previous experiments it is observed that the system performs better with the suffix and the prefix as features. So the first 4, first 3, first 2 and the 1st letter of the word (prefix) and the last 4, 3, 2, 1 letters of the word (suffix) are used as features.

The word is a Named Entity depends on the POS tag. So the POS tag is used as a feature. The chunk information is important to identify the Named entities with more than one word. So the chunk information is also included in the feature list.

The resources for the rest of the three languages (Oriya, Urdu and Bengali) are limited. Since we couldn't find the morphological analyzer for these languages, the first 4,3,2,1 letters and the last 4,3,2,1 letters are used as features.

The word being classified as a named entity also depends on the previous and next words. So these are used as features for all the languages

4 Evaluation

Precision, Recall and F-measure are used as metric to evaluate the system. These are calculated for Nested (both nested and largest possible NE match), Maximal (largest possible NE match) and Lexicon matches

Nested matches (n): The largest possible as well as the nested NE

Maximal matches (m): The largest possible NE matched with reference data.

Lexical item (l): The lexical item inside the NE are matched

5 Results

P_m, P_n ,P_l are the precision of maximal, nested, lexical matches respectively. R_m, R_n, R_l are the recall of maximal, nested, lexical matches respectively. Similarly F_m, F_n, F_l are the F-measure of maximal, nested, lexical matches.

The precision, recall, F-measure of five languages for CRF system is given in Table1. Table 2 has the lexical F-measure for each category. Similarly Table3 and Table4 give the precision, recall and F-measure for the five languages and the lexical F-measure for each category of HMM based system.

The performance of the NER system for five languages using a CRF based system is shown in Table-1.

	Precision			Recall			F-Measure		
Language	Pm	Pn	Pl	Rm	Rn	Rl	Fm	Fn	Fl
Bengali	61.28	61.45	66.36	21.18	20.54	24.43	31.48	30.79	***35.71***
Hindi	69.45	72.53	73.30	30.38	29.12	27.97	***42.27***	***41.56***	***40.49***
Oriya	37.27	38.65	64.20	19.56	16.19	25.75	25.66	22.82	***36.76***
Telugu	33.50	36.18	61.98	15.90	11.13	36.10	21.56	17.02	***45.62***
Urdu	45.55	46.11	52.35	26.08	24.24	30.13	33.17	31.78	***38.25***
m: Maximal n: Nested l: lexical									

Table 1: Performance of NER system for five languages (CRF)

	Bengali	Hindi	Oriya	Telugu	Urdu
NEP	33.06	***42.31***	***51.50***	15.70	11.72
NED	00.00	***42.85***	01.32	00.00	04.76
NEO	11.94	34.83	12.52	02.94	20.92
NEA	00.00	36.36	00.00	00.00	00.00
NEB	NP	NP	00.00	00.00	00.00
NETP	29.62	00.00	18.03	00.00	00.00
NETO	28.96	08.13	03.33	00.00	00.00
NEL	34.41	***61.08***	***46.73***	12.26	***54.59***
NETI	***63.86***	***70.37***	35.22	***90.49***	***62.22***
NEN	***75.34***	***74.07***	21.03	26.32	13.44
NEM	***46.96***	***58.33***	14.19	***42.01***	***77.72***
NETE	12.54	13.85	NP	08.63	00.00
NP: Not present in reference data					

Table 2: Class specific F-Measure for nested lexical match (CRF)

Measure	Precision			Recall			F-Measure		
Language	Pm	Pn	Pl	Rm	Rn	Rl	Fm	Fn	Fl
Bengali	50.66	50.78	58.00	25.03	24.26	30.26	33.50	32.83	***39.77***
Hindi	69.89	73.37	73.59	36.90	35.75	34.34	***48.30***	***47.16***	***46.84***
Oriya	33.10	34.70	60.98	24.63	20.61	36.72	28.24	25.86	***45.84***
Telugu	15.61	49.67	62.00	11.64	24.00	37.30	13.33	32.37	***46.58***
Urdu	42.81	47.14	56.21	29.37	29.69	37.15	34.48	36.83	***44.73***
m: Maximal n: Nested l: lexical									

Table 3: Performance of NER system for five languages (HMM)

	Bengali	Hindi	Oriya	Telugu	Urdu
NEP	***38.10***	***53.19***	***63.04***	23.14	34.96
NED	00.00	***52.94***	08.75	06.18	***49.18***
NEO	05.05	***40.42***	28.52	04.28	31.53
NEA	00.00	25.00	10.00	00.00	04.00
NEB	NP	NP	00.00	00.00	00.00
NETP	36.25	00.00	19.92	00.00	09.09
NETO	07.44	16.39	09.09	05.85	00.00
NEL	***49.35***	***72.03***	***50.09***	29.26	***58.59***
NETI	***50.81***	***62.56***	***46.30***	***70.75***	***53.98***
NEN	***66.66***	***81.96***	30.43	***86.29***	23.63
NEM	***62.98***	***54.44***	20.68	35.44	***82.64***
NETE	12.56	17.43	NP	11.67	00.00
NP: Not present in reference data					

Table 4: Class specific F-measure for nested lexical match (HMM)

Table-2 shows the performance for specific classes of named entities. Table-3 presents the results for the HMM based system and Table-4 gives the class specific performance of the HMM based system.

6 Error Analysis

In both HMM, CRF based system the pos-tag and the chunk information are being used. NEs are generally the noun chunks. The pos-tagger and the chunker that we used had low accuracy. These errors in the POS-Tag contributed significantly to errors in NER.

In Telugu the F-measure for the maximal named entities is low for both the CRF, HMM models. This is because the test data had a large number of TIME named entities which are 5-6 words long. These entities further had nested named entities. Both the models are able to identify the nested named entities. We chose not to consider the Time entities as a maximal entity since it was not tagged as a maximal NE as in some places. Considering it as a maximal NE the F-measure of the system increased significantly to over 30 for both HMM and CRF based systems.

It is also observed that many NE's were retrieved correctly but were wrongly classified. Working with fewer tag-set will help to increase the performance of the system but this is not suggested.

7 Conclusion

The overall performance of the HMM model based hybrid system is better than the CRF model for all the languages. The performance of HMM based system is less that that of CRF. We obtained a decent Lexical F-measure of 39.77, 46.84, 45.84, 46.58, 44.73for Bengali, Hindi, Oriya, Telugu and Urdu using rules over HMM model. HMM based model has a better F-measure for NEP, NEL, NEO classes when compared to CRF model

References

CRF++: http://crfpp.sourceforge.net

P. Avinesh, G. Karthik. 2007. *Parts-of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based learning*. Proceedings of SPSAL workshop IJCNLP 07

Thorsen Brants. 2000. *TnT: a statistical Part-of-Speech Tagger*. Proceeding of sixth conference on Applied Natural Language Processing.

N. Kumar and Pushpak Bhattacharyya. 2006. *NER in Hindi using MEMM*.

J. Lafferty, A. McCullam, F. Pereira. 2001. *Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data.* 18th International Conference on Machine Learning

Wei Li and A. McCallum. 2003. *Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction*. Transactions on Asian Language Information Processing.