TRDDC @ FIRE 2013: System for NER in Indian Languages

Report for NER Track at FIRE 2013

Nitin Ramrakhiyani and Sachin Pawar

Tata Research Development and Design Centre, Tata Consultancy Services Ltd., Pune
{nitin.ramrakhiyani, sachin7.p}@tcs.com

Abstract. In this work, we describe a Named Entity Recognition system for English and Hindi. The system is based on Conditional Random Fields (CRF)[1] and uses a feature representation of text tokens to learn a classifier and test the developed model. We designed a diverse set of lexical, syntactic, semantic and corpus statistics based features and explored this feature space to determine the best set of features for both English and Hindi.

1 Introduction

This paper presents our experiments and results in the NER for Indian Languages Track at FIRE 2013. We submitted one run each for English and Hindi. This being a sequential labeling task, we use Conditional Random Fields (CRF)[1] on features of text tokens to achieve the desired tagging. Apart from a set of standard features, we use a feature derived from English WordNet and a feature based on the concept of Point-wise Mutual Information (PMI) for English NER. For Hindi, we have tested using standard features and the PMI feature. The paper in Section 2 delves details about our experiments.

2 Methodology

Named Entity Recognition (NER) is one of the most important information extraction techniques being developed in the NLP and IR communities. Considerable success has been achieved in English with extraction of multiple entities as per domain of interest. However, the area poses considerable challenges when tried in other languages and particularly Indian Languages. The NER Track for Indian Languages at FIRE 2013 is one such initiative, which invites systems performing NER in Indian Languages and English of an Indian context. We have developed a system to perform NER in English and Hindi and submitted the same.

We use the open-source software, CRF++[2] which is one of the popular implementations of Conditional Random Fields (CRF)[1] for training a model on the training data and then use the model to generate tags for the test data. The system comprises of the following methodology:

Adding more features to the given training data.

The training data provided contains tokens with their POS tags, shallow parse tags and three levels of named entity classification tags. We have used the first level of classification throughout our experiments. We augmented the training data tokens with more features. The description of the new features incorporated is as follows:

Table I : Description of various features used

Feature No.	Feature	Feature Description
1	Word Structure	This feature records whether the token is in capitals, has numbers and similar word structure information. The values generated were: Cc – for token with first letter in capitals bb – for token with all letters in small, AA – for token with all letters in capitals, 11 – for token with all digits a1 – for token with digits and letters
2	Known Lower	This feature checks whether an all-capitals or first-letter-capital token is seen in a complete lower case representation in the training data. The values generated were: KL – for the token which has been observed in lower case, NKL – otherwise, NA – for all other tokens
3, 4, 5	Prefixes	Features 3, 4 and 5 record the token prefixes of length 2, 3 and 4 respectively.
6, 7, 8	Suffixes	Features 6, 7 and 8 record the token prefixes of length 2, 3 and 4 respectively.
9	WordNet	This feature captured information about the type of the token as observed in the English WordNET. We used hierarchy of hypernyms in the WordNet to check whether any of the indicator senses (E.g. person, organization, location, living things, etc.) are one of the ancestors of each of the token.
10	PMI	This feature captures the target class with which the token shows highest affinity, if any. The affinity of a token with a particular named entity class is measured as follows: $PMI_k(Token, Class)$ $= log\left(\frac{\Pr\ (Token\ and\ Class\ occur\ within\ window\ of\ k\ tokens)}{\Pr\ (Classord)\Pr\ (Token)}\right)$ Value of k = 1 produced the best results
11	Next Verb	This feature highlights the next verb present in the sentence.
12	Previous Verb	This feature highlights the previous verb present in the sentence.
13	Context words	This feature captures a bag of words around the token in consideration. The window of (+2,-2) tokens around the token produced the best results.

The correlation quotient highlighted in [3] is used to generate a list of terms which are used to represent documents of the collection in a feature vector representation. Feature 10 (based on PMI) is devised similar to this correlation quotient. For feature 10, however, we select a set of tokens for each class, having high PMI values for that particular class.

Testing the performance of the developed model.

As no development set was provided as part of track data, we separated the training data in a 80:20 percent ratio. We used the 80% part to train the model and the rest 20% was used as a development set. Based on the performance obtained on the development set, we tuned the features and finalized on the final set of features. The finalized set of features for English included the features 1-10 and 13 as described in table I. For Hindi, the finalized set included features 3-8 and 11-13.

We then trained a new model using the finalized set of features, on 100% of the training data and used it for recording performance on the test data.

References

[1] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of ICML*, pp. 282–289, (2001)

[2] Kudo, Taku. "CRF++: Yet another CRF toolkit." *Software available at http://crfpp. sourceforge. net* (2005)

[3] Chieu, Hai Leong, and Hwee Tou Ng. "A maximum entropy approach to information extraction from semi-structured and free text." *AAAI/IAAI* 2002 (2002): 786-791.