A Semi-supervised Approach for Maximum Entropy Based Hindi Named Entity Recognition

Sujan Kumar Saha, Pabitra Mitra, and Sudeshna Sarkar

Indian Institute of Technology, Kharagpur, India - 721302
{sujan.kr.saha,shudeshna,pabitra}@gmail.com

Abstract. Scarcity of annotated data is a challenge in building high performance named entity recognition (NER) systems in resource poor languages. We use a semi-supervised approach which uses a small annotated corpus and a large raw corpus for the Hindi NER task using maximum entropy classifier. A novel statistical annotation confidence measure is proposed for the purpose. The confidence measure is used in selective sampling based semi-supervised NER. Also a prior modulation of maximum entropy classifier is used where the annotation confidence values are used as 'prior weight'. The superiority of the proposed technique over baseline classifier is demonstrated extensively through experiments.

1 Introduction

Machine learning based approaches are commonly used for the development of named entity recognition (NER) systems (Borthwick 1999, Li and McCallum 2004, Saha et al. 2008). In this approach a classifier is trained using the annotated data with a suitable set of features. The performance of such statistical classifier largely depends on the amount of annotated data. But in many languages and domains sufficient annotated data do not exist. Manual creation of a sufficiently large named entity (NE) annotated corpus is costly and time consuming. Semi-supervised learning (SSL) may be adopted in such cases (Collins and Singer 1999, Mohit and Hwa 2005). SSL techniques use a limited annotated corpus along with a large raw corpus.

In this paper we propose a semi-supervised approach to named entity recognition and applied it on the Hindi NER task. A maximum entropy (MaxEnt) based classifier (baseline) is trained using a training corpus and a set of suitable features. The amount of training data is not sufficient and the baseline classifier suffers from poor recall. In order to improve the recall of the classifier we have used a large raw corpus for SSL.

A novel statistical confidence measure specific to the NER task is proposed for the purpose. A large raw corpus is annotated using the baseline classifier and the confidence weight (between 0 to 1) of the baseline annotation (for each word) is computed. The high confident part of the corpus is then selected and added to the original training data. A new classifier is built using the 'extended corpus'. In addition, we have modified the classifier in such a way that it accepts a 'prior

S. Chaudhury et al. (Eds.): PReMI 2009, LNCS 5909, pp. 225–230, 2009.

confidence weight' corresponding to each training sample. Each annotated word in the manually created training corpus is assigned a prior confidence of *one* and the samples in the baseline system annotated corpus is assigned a confidence weight in [0-1] using the proposed algorithm. The merged corpus is given to the modified classifier along with the prior confidence weights of the samples. Experimental results show that the classifiers built using the SSL techniques achieve a considerable amount of performance improvement over the baseline.

2 Computing Confidence Weights for NER Task

We present here the procedure for computing the confidence weight of the output obtained from a NE classifier. This method assigns a weight, ranging between 0 to 1, to each word and its annotation in the corpus. The procedure uses word based statistics derived from the training data. The details are presented below.

The confidence weight is computed using a word window of length $p+q+1$, containing p previous words and q next words of a target word. For each of the $p+q+1$ positions, we compute a class specific weight corresponding to the words in the lexicon. For each word (w) in the corpus the class specific word weight, $Wt_{\{C,\ pos\}}(w)$, is defined as,

$$Wt_{\{C,\ pos\}}(w) = \frac{\#\ occurrence\ of\ \text{'w'}\ in\ position\ \text{'pos'}\ of\ a\ NE\ of\ class\ \text{'C'}}{\#\ total\ occurrence\ of\ \text{'w'}\ in\ the\ corpus} \tag{1}$$

where *pos* denotes a particular position of the $p+q+1$ window. Using this equation we calculate the word weights for each position in the window for the NE-class predicted by the classifier. The weighted sum of these word-wights can be used as confidence weight.

During the computation of the weighted sum, the current position (i.e. the target word) is given the highest weight. We assign 50% of the total weight of the word window to the current word. The rest 50% weight is equally divided to the p previous positions and q next positions. That is the p previous words share a total of 25% weight. Now this weight is distributed to the p positions in such a way that the weight becomes inversely proportional to the distance; i.e., (-1) position shares more weight and (-p) shares the minimum. Similar distribution is followed for the next positions also.

Let us explain the word unigram based confidence computation method with an example. Assume in the Hindi sentence, "*Aja pradhAnama.ntrI manamohana ne kahA ...*[1] " (Today prime-minister Manmohan has said), '*manamohana*' is identified as person name by a classifier. To get the confidence of '*manamohana*' as a person name, here we consider its previous two words and next two words. So the confidence of the word as person is, $\lambda_{-2} \times Wt_{\{per,-2\}}(Aja) + \lambda_{-1} \times Wt_{\{per,-1\}}(pradhAnama.ntrI) + \lambda_0 \times Wt_{\{per,0\}}(manamohana) + \lambda_{+1} \times Wt_{\{per,+1\}}(ne) + \lambda_{+2} \times Wt_{\{per,+2\}}(kahA)$. The λ_i denotes the weight factor of position i. Using unigram word weights computed using Eqn. 1 and

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

corresponding λ_i values, the confidence weight of *manamohana* as person becomes 0.874.

During the confidence computation, few issues arise. To handle these issues the λ_i values are modified in such cases. Examples of such issues are, whether the current word is unknown (not present in the training corpus), whether any of the context words is unknown, whether any of the context words is postposition (these have very high occurrence in the corpus and as unigram these are not informative in identifying NEs) etc. Also in the NER task ambiguity between two classes is quite common. A NE belonging to a particular NE class, might occur as a NE or part of a NE of another class. Presence of some *clue words* often helps to resolve the ambiguous NEs. For example, presence of clue words like road, *palli* (locality), *setu* (bridge) etc. after a person name modifies it to a location NE. Such lists of clue words are collected for each pair of ambiguous NE classes. These words are given higher importance if they occur in the word window.

3 Confidence Prior Based MaxEnt

Maximum entropy (MaxEnt) principle is a commonly used learning technique which provides the probability of belongingness of a token to a class. MaxEnt computes the probability $p(o|h)$ for any o from the space of all possible outcomes O, and for every h from the space of all possible histories H. In NER task, history may be viewed as all information derivable from the training corpus relative to the current token. The computation of probability ($p(o|h)$) of an outcome for a token in MaxEnt depends on a set of features that are helpful in making the predictions about the outcome. Given a set of features and a training corpus, the MaxEnt estimation process produces a model in which every feature f_i (i - feature count index) has a weight α_i. We can compute the conditional probability as (Berger et al. 1996, Borthwick 1999):

$$p(o|h) = \frac{1}{Z(h)} \prod_i {\alpha_i}^{f_i(h,o)} \tag{2}$$

$$Z(h) = \sum_o \prod_i {\alpha_i}^{f_i(h,o)} \tag{3}$$

The conditional probability of the outcome is the product of the weights of all active features, normalized over the products of all the features. In the MaxEnt model the features are binary valued which are defined as,

$$f(h,o) = \begin{cases} 1, if\ (h,o) \in R; \\ 0, otherwise. \end{cases} \tag{4}$$

where R is an equivalent class over *(H,O)*. An example of MaxEnt feature is, if *previous word is professor* (h) = true and *NE_tag* (o) = person-begin then $f(h,o) = 1$. During the computation of $p(o|h)$ all such features which are active on the (h,o) pair, are considered.

Table 1. Features used in the MaxEnt based Hindi NER system

Type	Features
Word	current, previous two and next two words
NE tag	NE tag of previous two words
Digit information	Contains digit, digit & spl. char, numerical word
Affix information	Fixed length suffix, suffix list, fixed length prefix
POS information	POS of words, POS based binary features
Lists	Common locations, designations, person honorary terms, organization end terms

In simple supervised learning all the annotations in the training corpus are assumed to have same confidence values (say, 1 for each annotation). Now we introduce a confidence prior ($\rho_k \in [0, 1]$) for each annotation (a_k) in the training corpus. These values are multiplied with $f_i(h, o)$ in Equation 2 i.e. $\acute{f}_i(h, o)$ becomes $\rho_k \times f_i(h, o)$.

In MaxEnt training algorithm (Borthwick 1999) for a feature $f(h, o)$ the entire training corpus is scanned to determine the number of times ($\#f$) the feature is active. As the feature is binary, simple occurrence count represents the corresponding feature value over the corpus. The prior based MaxEnt does not simply counts the number of times the $f(h, o)$ fires. Here for each fire, the corresponding confidence prior is multiplied with the feature value (which is 1 in case of binary features), so the $\#f$ is now replaced by $\sum_k \ \rho_k \times f_i(h, o)$.

4 Experimental Result and Discussion

The experimental results are discussed in this section for the Hindi NER task.

4.1 Training and Test Data

The training data for the Hindi NER task is composed of about $200K$ words which is collected from the popular daily Hindi newspaper "Dainik Jagaran". Here three types of NEs are considered; namely *Person*, *Location* and *Organization*. The corpus has been manually annotated and contains $5,429$ person, $4,390$ location and $2,678$ organization entities. The Hindi test corpus contains $25K$ words, which is distinct from the training corpus. The test corpus contains 678 person, 480 location and 322 organization entities.

4.2 Features

A list of candidate features are selected for the Hindi NER task. Several experiments are conducted with the features, individually and in combination to find the best feature set for the task. The features are mentioned in Table 1.

Table 2. Semi-supervised Hindi NER accuracies

Confidence Wt.	Pre	Rec	F-val
>1 (baseline)	85.92	68.37	76.15
0.9	85.07	71.86	77.91
0.8	84.82	72.64	77.8
0.7	84.93	70.47	77.03
0.5	84.3	69.6	76.25
0 (all data)	83.35	69.4	75.74

4.3 Baseline System

Here we present the baseline system accuracy. In NER task the accuracies are measured in terms of *f-measure* or *f-value*, which is the weighted harmonic mean of precision and recall. *Precision* is the percentage of correct annotations and *recall* is the percentage of the total NEs that are successfully annotated.

The baseline NE classifier is trained only using the available annotated data and the above mentioned features. The highest baseline accuracy, which is a f-value of 76.15, is obtained using words, NE tag of the previous word, suffix, prefix, digit information, POS information and list based features. The precision of the system is 85.92% and the recall is 68.37%. For this the class specific f-values are, person - 73.13, location - 81.05 and organization - 75.55. The accuracy is achieved when the word window of length three (previous one word to next one word) is used. Use of wider window reduces the accuracy. From the baseline accuracy we observe that the baseline system suffers from poor recall. Now we investigate whether we can improve the result by making use of additional raw corpus which is often easily available.

4.4 Semi-supervised NER: Selecting High-Confidence Samples

To perform semi-supervised experiments we have used a raw corpus containing $2000K$ words. This corpus is annotated by the baseline classifier. The confidence of the annotation is measured using the proposed approach (Section 2) and the the sentences with high confidence are selected and added to the training data. We have experimented with different confidence threshold values. The details of the experiments are given in Table 2.

It can be observed from Table 2 that the semi-supervised classifier performs better than the baseline if a suitable confidence threshold is chosen. The highest accuracy is achieved when the threshold is 0.9, which is a f-value of 77.91. Here amount of corpus selected and added with the training corpus is 179K words. Comparing with the baseline accuracy here we have achieved higher recall with a small decrease in precision. The recall is increased to 71.86 from 68.37. Use of lower threshold i.e. selecting more *additional samples* reduces the accuracy.

Table 3. Performance of semi-supervised approaches in Hindi NER

System	Pre	Rec	F-val
Baseline classifier	85.92	68.37	76.15
Semi-supervised high confident	85.07	71.86	77.91
Semi-supervised prior based	85.73	72.63	78.64

4.5 Semi-supervised NER: Confidence Prior in Classifier

Now we present the performance of the classifier which uses the annotation confidence value as a prior weight during training as discussed in Section 3. Here we have used all the baseline classifier annotated data along with their confidence weights. The MaxEnt classifier with prior based semi-supervised approach yields a f-value of 78.64. The prior based method performs better than baseline classifier as well as the high-confident portion selection based SSL approach (see Table 3.

5 Conclusion

We propose a semi-supervised learning framework for named entity recognition. The approach is based on a novel statistical confidence measure computed using NER specific cues. We use a sample selection approach as well as a prior modulation approach to enhance the performance of the NE classifier using the confidence measure. The approach is particularly useful for resource poor languages like Hindi where annotated data is scarce. Experimental results on a Hindi corpus consisting of a small annotated set and a much larger raw data set demonstrate that semi-supervised learning improves performance significantly over baseline classifier.

References

Berger, A., Pietra, S., Pietra, V.: A maximum entropy approach to natural language processing. Computational Linguistic 22(1), 39–71 (1996)

Borthwick, A.: A maximum entropy approach to named entity recognition. Ph.D. thesis, Computer Science Department, New York University (1999)

Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999)

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 (2004)

Mohit, B., Hwa, R.: Syntax-based semi-supervised named entity tagging. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions, pp. 57–60. Association for Computational Linguistics, Ann Arbor (2005)

Saha, S., Sarkar, S., Mitra, P.: A hybrid feature set based maximum entropy Hindi named entity recognition. In: Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP), pp. 343–349 (2008)