Word Clustering and Word Selection based Feature Reduction for MaxEnt based Hindi NER

Sujan Kumar Saha
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
sujan.kr.saha@gmail.com

Pabitra Mitra
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
pabitra@gmail.com

Sudeshna Sarkar
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
shudeshna@gmail.com

Abstract

Statistical machine learning methods are employed to train a Named Entity Recognizer from annotated data. Methods like Maximum Entropy and Conditional Random Fields make use of features for the training purpose. These methods tend to overfit when the available training corpus is limited especially if the number of features is large or the number of values for a feature is large. To overcome this we proposed two techniques for feature reduction based on word clustering and selection. A number of word similarity measures are proposed for clustering words for the Named Entity Recognition task. A few corpus based statistical measures are used for important word selection. The feature reduction techniques lead to a substantial performance improvement over baseline Maximum Entropy technique.

1 Introduction

Named Entity Recognition (NER) involves locating and classifying the names in a text. NER is an important task, having applications in information extraction, question answering, machine translation and in most other Natural Language Processing (NLP) applications. NER systems have been developed for English and few other languages with high accuracy. These belong to two main categories based on machine learning (Bikel et al., 1997; Borthwick, 1999; McCallum and Li, 2003) and language or domain specific rules (Grishman, 1995; Wakao et al., 1996).

In English, the names are usually capitalized which is an important clue for identifying a name. Absence of capitalization makes the Hindi NER task difficult. Also, person names are more diverse in Indian languages, many common words being used as names.

A pioneering work on Hindi NER is by Li and McCallum (2003) where they used Conditional Random Fields (CRF) and feature induction to automatically construct only the features that are important for recognition. In an effort to reduce overfitting, they use a combination of a Gaussian prior and early-stopping.

In their Maximum Entropy (MaxEnt) based approach for Hindi NER development, Saha et al. (2008) also observed that the performance of the MaxEnt based model often decreases when huge number of features are used in the model. This is due to overfitting which is a serious problem in most of the NLP tasks in resource poor languages where annotated data is scarce.

This paper is a study on effectiveness of word clustering and selection as feature reduction techniques for MaxEnt based NER. For clustering we use a number of word similarities like cosine similarity among words and co-occurrence, along with the k-means clustering algorithm. The clusters are then used as features instead of words. For important word selection we use corpus based statistical measurements to find the importance of the words in the NER task. A significant performance improvement over baseline MaxEnt was observed after using the above feature reduction techniques.

The paper is organized as follows. The MaxEnt

Proceedings of ACL-08: HLT, pages 488–495,
Columbus, Ohio, USA, June 2008.

based NER system is described in Section 2. Various approaches for word clustering are discussed in Section 3. Next section presents the procedure for selecting the important words. In Section 5 experimental results and related discussions are given. Finally Section 6 concludes the paper.

2 Maximum Entropy Based Model for Hindi NER

Maximum Entropy (MaxEnt) principle is a commonly used technique which provides 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, history can 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 predictions about the outcome. The features may be binary-valued or multi-valued. Given a set of features and a training corpus, the MaxEnt estimation process produces a model in which every feature f_i has a weight α_i. We can compute the conditional probability as (Berger et al., 1996):

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

$$Z(h) = \sum_o \prod_i {\alpha_i}^{f_i(h,o)} \quad (2)$$

The conditional probability of the outcome is the product of the weights of all active features, normalized over the products of all the features. For our development we have used a Java based open-nlp MaxEnt toolkit[1]. A beam search algorithm is used to get the most probable class from the probabilities.

2.1 Training Corpus

The training data for the Hindi NER task is composed of about $243K$ words which is collected from the popular daily Hindi newspaper "Dainik Jagaran". This corpus has been manually annotated and contains about 16,491 Named Entities (NEs). In this study we have considered 4 types of NEs, these are $Person$ (Per), $Location$ (Loc), $Organization$ (Org) and $Date$ (Dat). To recognize entity boundaries each name class N has 4 types of labels: N_Begin, $N\_Continue$, N_End and N_Unique. For example, ***Kharagpur*** is annotated as Loc_Unique and ***Atal Bihari Vajpeyi*** is annotated as $Per\_Begin\ Per\_Continue\ Per\_End$. Hence, there are a total of 17 classes including one class for not-name. The corpus contains 6298 person, 4696 location, 3652 organization and 1845 date entities.

[1] http://sourceforge.net/projects/maxent/

Type	Features
Word	w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}
NE Tag	t_{i-1}, t_{i-2}
Digit information	Contains digit, Only digit, Four digit, Numerical word
Affix information	Fixed length suffix, Suffix list, Fixed length prefix
POS information	POS of words, Coarse-grained POS, POS based binary features

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

2.2 Feature Description

We have identified a number of candidate features for the Hindi NER task. Several experiments were conducted with the identified features, individually and in combination. Some of the features are mentioned below. They are summarized in Table 1.

Static Word Feature: Recognition of NE is highly dependent on contexts. So the surrounding words of a particular word (w_i) are used as features. During our experiments different combinations of previous 3 words ($w_{i-3}...w_{i-1}$) to next 3 words ($w_{i+1}...w_{i+3}$) are treated as features. This is represented by L binary features where L is the size of lexicon.

Dynamic NE tag: NE tags of the previous words ($t_{i-m}...t_{i-1}$) are used as features. During decoding, the value of this feature for a word (w_i) is obtained only after the computation of the NE tag for the previous word (w_{i-1}).

Digit Information: If a word (w_i) contains digit(s) then the feature $ContainsDigit$ is set to 1. This feature is used with some modifications also. $OnlyDigit$, which is set to 1 if the word contains

Feature Id	Feature	Per	Loc	Org	Dat	Total
F1	w_i, w_{i-1}, w_{i+1}	61.36	68.29	52.12	88.9	67.26
F2	w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}	64.10	67.81	58	92.30	69.09
F3	w_i, w_{i-1}, w_{i-2}, w_{i-3}, w_{i+1}, w_{i+2}, w_{i+3}	60.42	67.81	51.48	90.18	66.84
F4	w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}, Suffix	66.67	73.36	58.58	89.09	71.2
F5	w_i, w_{i-1}, w_{i+1}, t_{i-1}, Suffix	69.65	75.8	59.31	89.09	73.42
F6	w_i, w_{i-1}, w_{i+1}, t_{i-1}, Prefix	66.67	71	58.58	87.8	70.02
F7	w_i, w_{i-1}, w_{i+1}, t_{i-1}, Prefix, Suffix	70.61	71	59.31	89.09	72.5
F8	w_i, w_{i-1}, w_{i+1}, t_{i-1}, Suffix, Digit	70.61	75.8	60.54	93.8	74.26
F9	w_i, w_{i-1}, w_{i+1}, t_{i-1}, POS (28 tags)	64.25	71	60.54	89.09	70.39
F10	w_i, w_{i-1}, w_{i+1}, t_{i-1}, POS (coarse grained)	69.65	75.8	59.31	92.82	74.16
F11	w_i, w_{i-1}, w_{i+1}, T_{i-1}, Suffix, Digit, NomPSP	72.26	78.6	61.36	92.82	75.6
F12	w_i, w_{i-1}, w_{i+1}, w_{i-2}, w_{i+2}, T_{i-1}, Prefix, Suffix, Digit, NomPSP	65.26	78.01	52.12	93.33	72.65

Table 2: F-values for different features in the MaxEnt based Hindi NER system

only digits, $4Digit$, which is set to 1 if the word contains only 4 digits, etc. are some modifications of the feature which are helpful.

Numerical Word: For a word (w_i) if it is a numerical word i.e. word denoting a number (e.g. eka[2] (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 (set of characters occurring at the end of the word) of the current and surrounding words are used as features. Secondly we compiled list of common suffixes of place names in Hindi. For example, $pura$, $bAda$, $nagara$ etc. are location suffixes. We used binary feature corresponding to the list - whether a given word has a suffix from the list.

Word Prefix: Prefix information of a word may be also helpful in identifying whether it is a NE. A fixed length word prefix (set of characters occurring at the beginning of the word) of current and surrounding words are treated as features. List of important prefixes, which are used frequently in the NEs, are also effective.

Parts-of-Speech (POS) Information: The POS of the current word and the surrounding words are used as feature for NER. We have used a Hindi POS tagger developed at IIT Kharagpur, India which has an accuracy about 90%. We have used the POS values of the current and surrounding words 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. 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 word is nominal and the next

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

word is a PSP.

2.3 Performance of Hindi NER using MaxEnt Method

The performance of the MaxEnt based Hindi NER using the above mentioned features is reported here as a baseline. We have evaluated the system using a blind test corpus of 25K words. The test corpus contains 521 person, 728 location, 262 organization and 236 date entities. The accuracies are measured in terms of the f-measure, which is the weighted harmonic mean of precision and recall. Precision is the fraction of the correct annotations and recall is the fraction of the total NEs that are successfully annotated. The general formula for measuring the f-measure or f-value is, $F_\beta = (1+\beta^2)\,.\,(precision\,.\,recall) \setminus (\beta^2\,.\,precision + recall)$. Here the value of β is taken as 1. In Table 2 we have shown the accuracy values for few feature sets.

While experimenting with static word features, we have observed that a window of previous and next two words ($w_{i-2}...w_{i+2}$) gives best result (69.09) using the word features only. But when w_{i-3} and w_{i+3} are added with it, the f-value is reduced to 66.84. Again when w_{i-2} and w_{i+2} are deducted from the feature set (i.e. only w_{i-1} and w_{i+1} as feature), the f-value is reduced to 67.26. This demonstrates that w_{i-2} and w_{i+2} are helpful features in NE identification.

When suffix, prefix and digit information are added to the feature set, the f-value is increased upto 74.26. The value is obtained using the feature set F8 [w_i, w_{i-1}, w_{i+1}, t_{i-1}, Suffix, Digit]. It is observed that when w_{i-2} and w_{i+2} are added with the feature, the accuracy decreases by 2%. It contradicts the results using the word features only. Another interesting observation is that prefix information are helpful features in NE identification as these increase accuracy when separately added with the word features (F6). Similarly the suffix information helps in increasing the accuracy. But when both the suffix and prefix information are used in combination along with the word features, the f-value is decreased. From Table 2, a f-value of 73.42 is obtained using F5 [w_i, w_{i-1}, w_{i+1}, t_{i-1}, Suffix] but when prefix information are added with it (F7), the f-value is reduced to 72.5.

POS information are important features in NER. In general it is observed that coarse grained POS information performs better than the finer grained POS information. The best accuracy (75.6 f-value) of the baseline system is obtained using the binary NomPSP feature along with word feature (w_{i-1}, w_{i+1}), suffix and digit information. It is noted that when w_{i-2}, w_{i+2} and prefix information are added with the best feature, the f-value is reduced to 72.65.

From the above discussion it is clear that the system suffers from overfitting if a large number of features are used to train the system. Note that the surrounding word (w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2} etc.) features can take any value from the lexicon and hence are of high dimensionality. These cause the degradation of performance of the system. However it is obvious that few words in the lexicon are important in identification of NEs.

To solve the problem of high dimensionality we use clustering to group the words present in the corpus into much smaller number of clusters. Then the word clusters are used as features instead of the word features (for surrounding words). For example, our Hindi corpus contains 17,456 different words, which are grouped into N (say 100) clusters. Then for a particular word, it is assigned to a cluster and the corresponding cluster-id is used as feature. Hence the number of features is reduced to 100 instead of 17,456.

Similarly, selection of important words can also solve the problem of high dimensionality. As some of the words in the lexicon play important role in the NE identification process, we aim to select these particular words. Only these important words are used in NE identification instead of all words in the corpus.

3 Word Clustering

Clustering is the process of grouping together objects based on their similarity. The measure of similarity is critical for good quality clustering. We have experimented with some approaches to compute word-word similarity. These are described in details in the following section.

3.1 Cosine Similarity based on Sentence Level Co-occurrence

A word is represented by a binary vector of dimension same as the number of sentences in the corpus. A component of the vector is 1 if the word occurs in the corresponding sentence and zero otherwise. Then we measure cosine similarity between the word vectors. The cosine similarity between two word vectors ($\vec{A}$ and $\vec{B}$) with dimension d is measured as:

$$CosSim(\vec{A}, \vec{B}) = \frac{\sum_d A_d B_d}{(\sum_d A_d^2)^{\frac{1}{2}} \times (\sum_d B_d^2)^{\frac{1}{2}}} \quad (3)$$

This measures the number of co-occurring sentences.

3.2 Cosine Similarity based on Proximal Words

In this measure a word is represented by a vector having dimension same as the lexicon size. For ease of implementation we have taken a dimension of 2×200, where each component of the vector corresponds to one of the 200 most frequent preceding and following words of a token word. $List\_Prev$ containing the most frequent (top 200) previous words (w_{i-1} or w_{i-2} if w_i is the first word of a NE) and $List\_Next$ contains 200 most frequent next words (w_{i+1} or w_{i+2} if w_i is the last word of a NE). A particular word w_k may occur several times (say n) in the corpus. For each occurrence of w_k find if its previous word (w_{k-1} or w_{k-2}) matches any element of $List\_Prev$. If matches, then set 1 to the corresponding position of the vector and set zero to all other positions related to $List\_Prev$. Similarly check the next word (w_{k+1} or w_{k+2}) in the $List\_Next$ and find the values of the corresponding positions. The final word vector $\vec{W_k}$ is obtained by taking the average of all occurrences of w_k. Then the cosine similarity is measured between the word vectors. This measures the similarity of the contexts of the occurrences of the word in terms of the proximal words.

3.3 Similarity based on Proximity to NE Categories

Here, for each word (w_i) in the corpus four binary vectors are defined corresponding to two preceding and two following positions (i-1, i-2, i+1, i+2). Each binary vector is of dimension five corresponding to four NE classes (C_j) and one for the not-name class. For a particular word w_k, find all the words occur in a particular position (say, +1). Measure the fraction ($P_j(w_k)$) of these words belonging to a class C_j. The component of the word vector $\vec{W_k}$ for the position corresponding to C_j is $P_j(w_k)$.

$$P_j(w_k) = \frac{No.\ of\ times\ w_{k+1}\ is\ a\ NE\ of\ class\ C_j}{Total\ occurrence\ of\ w_k\ in\ corpus}$$

The Euclidean distance is used to find the similarity between the above word vectors as a similarity measure. Some of the word vectors for the +1 position are given in Table 3. In this table we have given the word vectors for a few Hindi words, which are, *sthita* (located), *shahara* (city), *jAkara* (go), *nagara* (township), *gA.nva* (village), *nivAsI* (resident), *mishrA* (a surname) and *limiTeDa* (ltd.). From the table we observe that the word vectors are close for *sthita* [0 0.478 0 0 0.522], *shahara* [0 0.585 0.001 0.024 0.39], *nagara* [0 0.507 0.019 0 0.474] and *gA.nva* [0 0.551 0 0 0.449]. So these words are considered as close.

Word	Per	Loc	Org	Dat	Not
sthita	0	0.478	0	0	0.522
shahara	0	0.585	0.001	0.024	0.39
jAkara	0	0.22	0	0	0.88
nagara	0	0.507	0.019	0	0.474
gA.nva	0	0.551	0	0	0.449
nivAsI	0.108	0.622	0	0	0.27
mishrA	0.889	0	0	0	0.111
limiTeDa	0	0	1	0	0

Table 3: Example of some word vectors for next (+1) position (see text for glosses)

3.4 K-means Clustering

Using the above similarity measures we have used the k-means algorithm. The seeds were randomly selected. The value of k (number of clusters) was varied till the best result is obtained.

4 Important Word Selection

It is noted that not all words are equally important in determining the NE category. Some of the words

in the lexicon are typically associated with a particular NE category and hence have important role to play in the classification process. We describe below a few statistical techniques that has been used to identify the important words.

4.1 Class Independent Important Word Selection

We define context words as those which occur in proximity of a NE. In other words, context words are the words present in the w_{i-2}, w_{i-1}, w_{i+1} or w_{i+2} position if w_i is a NE. Note that only a subset of the lexicon are context words. For all the context words, its N_weight is calculated as the ratio between the occurrence of the word as a context word and its total number of occurrence in the corpus. The context words having the higher N_weight are considered as important words for NER. For our experiments we have considered top 500 words as important words.

$$N\_weight(w_i) = \frac{Occurrence\ of\ w_i\ as\ context\ word}{Total\ occurrence\ of\ w_i\ in\ corpus}$$

4.2 Important Words for Each Class

Similar to the class independent important word selection from the contexts, important words are selected for individual classes also. This is an extension of the previous context word considering only NEs of a particular class. For person, location, organization and date classes we have considered top 150, 120, 50 and 50 words respectively as important words. Four binary features are also defined for these four classes. These are defined as having value 1 if any of the context words belongs to the important words list for a particular class.

4.3 Important Words for Each Position

Position based important words are also selected from the corpus. Here instead of context, particular positions are considered. Four lists are compiled for two preceding and two following positions (-2, -1, +1 and +2).

5 Evaluation of NE Recognition

The following subsections contain the experimental results using word clustering and important word selection. The results demonstrate the effectiveness of word clustering and important word selection over the baseline MaxEnt model.

k	Per	Loc	Org	Dat	Total
20	66.33	74.57	43.64	91.30	69.54
50	64.13	76.35	52	93.62	71.7
80	66.33	74.57	53.85	93.62	72.08
100	70.1	73.1	57.7	96.62	**72.78**
120	66.15	73.43	54.9	93.62	71.52
150	66.88	74.94	53.06	95.65	72.33
200	66.09	73.82	52	92	71.13

Table 4: Variation of MaxEnt based system accuracy depending on number of clusters (k)

5.1 Using Word Clusters

To evaluate the effectiveness of the clustering approaches in Hindi NER, we have used cluster features instead of word features. For the surrounding words, corresponding cluster-ids are used as feature.

Choice of k : We have already mentioned that, for k-means clustering number of classes (k) should be determined initially. To find suitable k we had conducted the following experiments. We have selected a feature F1 (mentioned in Table 2) and applied the clusters with different k as features replacing the word features. In Table 4 we have summarized the experimental results, in order to find a suitable k for clustering, the word vectors obtained using the procedure described in Section 3.3. From the table we observe that the best result is obtained when k is 100. We have used $k = 100$ for the subsequent experiments for comparing the effectiveness of the features. Similarly when we deal with all the words in the corpus (17,465 words), we got best results when the words are clustered into 1100 clusters. $\diamondsuit$

The details of the comparison between the baseline word features and the reduced features obtained using clustering are given in Table 5. In general it is observed that clustering has improved the performance over baseline features. Using only cluster features the system provides a maximum f-value of 74.26 where the corresponding word features give f-value of 69.09.

Among the various similarity measures of clustering, improved results are obtained using the clus-

Feature	Using Word Features	Using Clusters (C1)	Using Clusters (C2)	Using Clusters (C3)
w_i, window(-1, +1)	67.26	69.67	72.05	72.78
w_i, window(-2, +2)	69.09	71.52	72.65	74.26
w_i, window(-1, +1), Suffix	73.42	74.24	75.44	75.84
w_i, window(-1, +1), Prefix, Suffix	72.5	74.76	75.7	76.33
w_i, window(-1, +1), Prefix, Suffix, Digit	74.26	75.09	75.91	76.41
w_i, window(-1, +1), Prefix, Suffix, Digit, NomPSP	75.6	77.2	77.39	77.61
w_i, window(-2, +2), Prefix, Suffix, Digit, NomPSP	72.65	77.86	78.61	**79.03**

Table 5: F-values for different features in a MaxEnt based Hindi NER with clustering based feature reduction [$window(-m, +n)$ refers to the cluster or word features corresponding to previous m positions and next n positions; C1 is the clusters which use sentence level co-occurrence based cosine similarity (3.1), C2 denotes the clusters which use proximal word based cosine similarity (3.2), C3 denotes the clusters for each positions related to NE (3.3)]

ters which uses the similarity measurement based on proximity of the words to NE categories (defined in Section 3.3).

Using clustering features the best f-value (79.03) is obtained using clusters for previous two and next two words along with the suffix, prefix, digit and POS information.

It is observed that the prefix information increases the accuracy if applied along with suffix information when cluster features are used. More interestingly, addition of cluster features for positions -2 and $+2$ over the feature [window(-1, +1), Suffix, Prefix, Digit, NomPSP] increase the f-value from 77.61 to 79.03. But in the baseline system addition of word features (w_{i-2} and w_{i+2}) over the same feature decrease the f-value from 75.6 to 72.65.

5.2 Using Important Word Selection

The details of the comparison between the word feature and the reduced features based on important word selection are given in Table 6. For the surrounding word features, find whether the particular word (e.g. at position -1, -2 etc.) presents in the important words list (corresponding to the particular position if position based important words are considered). If the word occurs in the list then the word is used as features. In general it is observed that word selection also improves performance over baseline features. Among the different approaches, the best result is obtained when important words for two preceding and two following positions (defined in Section 4.3) are selected. Using important word based features, the highest f-value of 79.85 is obtained by using the important words for previous two and next two positions along with the suffix, prefix, digit and POS information.

5.3 Relative Effectiveness of Clustering and Word Selection

In most of the cases clustering based features perform better then the important word based feature reduction. But the best f-value (79.85) of the system (using the clustering based and important word based features separately) is obtained by using important word based features.

Next we have made an experiment by considering both the clusters and important words combined. We have defined the combined feature as, if the word (w_i) is in the corresponding important word list then the word is used as feature otherwise the corresponding cluster-id (in which w_i belongs to) is considered as feature. Using the combined feature, we have achieved further improvement. Here we are able to achieve the highest f-value of 80.01.

6 Conclusion

A hierarchical word clustering technique, where clusters are driven automatically from large unan-

Feature	Using Word Features	Using Words (I1)	Using Words (I2)	Using Words (I3)
w_i, window(-1, +1)	67.26	66.31	67.53	66.8
w_i, window(-2, +2)	69.09	72.04	72.9	73.34
w_i, window(-1, +1), Suffix	73.42	73.85	73.12	74.61
w_i, window(-1, +1), Prefix, Suffix	72.5	73.52	73.94	74.87
w_i, window(-1, +1), Prefix, Suffix, Digit	74.26	73.97	74.13	74.7
w_i, window(-1, +1), Prefix, Suffix, Digit, NomPSP	75.6	75.84	76.6	77.22
w_i, window(-2, +2), Prefix, Suffix, Digit, NomPSP	72.65	76.69	77.42	**79.85**

Table 6: F-values for different features in a MaxEnt based Hindi NER with important word based feature reduction [$window(-m, +n)$ refers to the important word or baseline word features corresponding to previous m positions and next n positions; I1 is the class independent important words (4.1), I2 denotes the important words for each class (4.2), I3 denotes the important words for each positions (4.3)]

notated corpus, is used by Miller et al. (2004) for augmenting annotated training data. Note that our clustering approach is different, where the clusters are obtained using some statistics derived from the annotated corpus, and also the purpose is different as we have used the clusters for feature reduction.

In this paper we propose two feature reduction techniques for Hindi NER based on word clustering and word selection. A number of word similarity measures are used for clustering. A few statistical approaches are used for the selection of important words. It is observed that significant enhancement of accuracy over the baseline system which use word features is obtained. This is probably due to reduction of overfitting. This is more important for a resource poor languages like Hindi where there is scarcity in annotated training data and other NER resources (like, gazetteer lists).

7 Acknowledgement

The work is partially funded by Microsoft Research India.

References

Berger A L, Pietra S D and Pietra V D 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistic*, 22(1):39–71.

Bikel D M, Miller S, Schwartz R and W Ralph. 1997. Nymble: A High Performance Learning Name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201.

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

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

Li W and McCallum A. 2003. 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.

McCallum A and Li W. 2003. Early Results for Named Entity Recognition with Conditional Random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL.*

Miller S, Guinness J and Zamanian A. 2004. Name Tagging with Word Clusters and Discriminative Training. In *Proceedings of the HLT-NAACL 2004*, pages 337–342.

Saha S K, Sarkar S and Mitra P. 2008. A Hybrid Feature Set based Maximum Entropy Hindi Named Entity Recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-08)*, pages 343–349.

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*.