

Voted NER System using Appropriate Unlabeled Data

Asif Ekbal

Dept. of Computer Science & Engg.,
Jadavpur University, Kolkata-700032,
India
asif.ekbal@gmail.com

Sivaji Bandyopadhyay

Dept. of Computer Science & Engg.,
Jadavpur University, Kolkata-700032,
India
sivaji_cse_ju@yahoo.com

Abstract

This paper reports a voted Named Entity Recognition (NER) system with the use of appropriate unlabeled data. The proposed method is based on the classifiers such as Maximum Entropy (ME), Conditional Random Field (CRF) and Support Vector Machine (SVM) and has been tested for Bengali. The system makes use of the language independent features in the form of different contextual and orthographic word level features along with the language dependent features extracted from the Part of Speech (POS) tagger and gazetteers. Context patterns generated from the unlabeled data using an active learning method have been used as the features in each of the classifiers. A semi-supervised method has been used to describe the measures to automatically select effective documents and sentences from unlabeled data. Finally, the models have been combined together into a final system by weighted voting technique. Experimental results show the effectiveness of the proposed approach with the overall *Recall*, *Precision*, and *F-Score* values of 93.81%, 92.18% and 92.98%, respectively. We have shown how the language dependent features can improve the system performance.

1 Introduction

Named Entity Recognition (NER) is an important tool in almost all Natural Language Processing (NLP) application areas. Machine learning (ML) approaches are more popularly used in NER because these are easily trainable, adoptable to different domains and languages as well as their maintenance are also less expensive. Some of the very effective ML approaches used in NER are ME (Borthwick, 1999), CRF (Lafferty et al., 2001) and SVM (Yamada et al., 2002). In the earlier work (Florian et al., 2003), it has been shown that combination of several ML

models yields better performance than any single ML model. One drawback of the ML techniques to NLP tasks is the requirement of a large amount of annotated data to achieve a reasonable performance.

Indian languages are resource-constrained and the manual preparation of NE annotated data is both time consuming and cost intensive. It is important to decide how the system should effectively select unlabeled data and how the size and relevance of data impact the performance. India is a multilingual country with great cultural diversities. Named Entity (NE) identification in Indian languages in general and Bengali in particular is difficult and challenging as:

1. Unlike English and most of the European languages, Bengali lacks capitalization information, which plays a very important role in identifying NEs.
2. Indian person names are generally found in the dictionary as common nouns with some specific meanings. For example, *kabita* [Kabita] is a person name and can also be found in the dictionary as a common noun with the meaning ‘poem’.
3. Bengali is an inflectional language providing one of the richest and most challenging sets of linguistic and statistical features resulting in long and complex wordforms. For example, the person name *sachin* [root] can appear as *sachiner* [inflection:-er], *sachinke* [inflection:-ke], *sachinbAbu* [inflection: -bAbu], *sachinda* [inflection:-dA] etc. The location name *kolkata* [root] can appear in different wordforms like *kolkatar* [inflection:-r], *kolkate* [inflection:-te], *kolkatei* [inflection:-i] etc.
4. Bengali is a relatively free phrase order language. Thus, NEs can appear in any position of the sentence making the NER task more difficult.
5. Bengali, like other Indian languages, is a resource-constrained language. The annotated corpus, name dictionaries, good morphological

analyzers, POS taggers etc. are not yet available in the required measure.

6. Although Indian languages have a very old and rich literary history, technological developments are of recent origin.

7. Web sources for name lists are available in English, but such lists are not available in Bengali. This necessitates the use of transliteration for creating such lists.

A HMM based NER system for Bengali has been reported in Ekbal et al. (2007b), where additional contextual information has been considered during emission probabilities and NE suffixes are used for handling the unknown words. More recently, the works in the area of Bengali NER can be found in Ekbal et al. (2008a), and Ekbal and Bandyopadhyay (2008b) with the CRF, and SVM approach, respectively. Other than Bengali, the works on Hindi can be found in Li and McCallum (2004) with CRF and Saha et al. (2008) with a hybrid feature set based ME approach. Various works of NER involving Indian languages are reported in IJCNLP-08 NER Shared Task on South and South East Asian Languages (NERSSEAL)¹ using various techniques.

2 Named Entity Recognition in Bengali

We have used a Bengali news corpus (Ekbal and Bandyopadhyay, 2008c), developed from the web-archive of a widely read Bengali newspaper for NER. A portion of this corpus containing 200K wordforms has been manually annotated with the four NE tags namely, *Person*, *Location*, *Organization* and *Miscellaneous*. We have also used the NE annotated data of 122K wordforms, collected from the NERSSEAL shared task. The shared task data was originally annotated with a fine-grained NE tagset of twelve tags. We consider only those tags that represent person, location, organization, and miscellaneous names (NEN [number], NEM [Measurement] and NETI [Time]). Other tags have been mapped to the NNE tags that represent the “other-than-NE” category. In order to properly denote the boundaries of NEs, four NE tags are further divided into the following forms:

B-XXX: Beginning of a multiword NE, I-XXX: Internal of a multiword NE consisting of more than two words, E-XXX: End of a multiword NE, XXX→PER/LOC/ORG/MISC. For example, the name *sachin ramesh tendulkar* is

tagged as *sachin/B-PER ramesh/I-PER tendulkar/E-PER*. The single word NE is tagged as, PER: Person name, LOC: Location name, ORG: Organization name and MISC: Miscellaneous name. In the output, sixteen NE tags are replaced with the four NE tags.

2.1 Our Approaches

Initially, we started with the development of a NER system using an active learning method. This is used as the *baseline* model. Four supervised NER systems based on ME, CRF and SVM have been developed. Two different systems with the SVM model, one using **forward parsing** (SVM-F) that parses from left to right and other using **backward parsing** (SVM-B) that parses from right to left, have been developed. The SVM system has been developed based on (Valdimir, 1995), which perform classification by constructing an N-dimensional hyperplane that optimally separates data into two categories. We have used *YamCha* toolkit (<http://chasen.org/~taku/software/yamcha>), an SVM based tool for detecting classes in documents and formulating the NER task as a sequential labeling problem. Here, the *pairwise* multi-class decision method and *polynomial kernel function* have been used. The TinySVM-0.0² classifier has been used for classification. The C++ based CRF++ package (<http://crfpp.sourceforge.net>) and the C++ based ME package³ have been used for NER.

Performance of the supervised NER models is limited in part by the amount of labeled training data available. A part of the available unlabeled corpus (Ekbal and Bandyopadhyay, 2008c) has been used to address this problem. Based on the original training on the labeled corpus, there will be some tags in the unlabeled corpus that the taggers will be very sure about. We have proposed a semi-supervised learning technique that selects appropriate data from the available large unlabeled corpora and adds to the initial training set in order to improve the performance of the taggers. The models are retrained with this new training set and this process is repeated in a bootstrapped manner.

2.2 Named Entity Features

The main features for the NER task have been identified based on the different possible combinations of available word and tag contexts. In

¹ <http://ltrc.iiit.ac.in/ner-ssea-08/proc/index.html>

²<http://cl.aist-nara.ac.jp/~taku/ku/software/TinySVM>

³<http://homepages.inf.ed.ac.uk/s0450736/software/maxent-20061005.tar.bz2>

addition to these, various gazetteer lists have been developed for use in the NER tasks.

The set of features ‘F’ contains language independent as well as language dependent features. The set of language independent features includes the context words, fixed length prefixes and suffixes of all the words, dynamic NE information of the previous word(s), first word, length of the word, digit and infrequent word information. Language dependent features include the set of known suffixes that may appear with the various NEs, clue words that help in predicting the location and organization names, words that help to recognize measurement expressions, designation words that help to identify person names, various gazetteer lists that include the first names, middle names, last names, location names, organization names, function words, weekdays and month names. We have also used the part of speech (POS) information of the current and/or the surrounding word(s) as the features.

Language independent NE features can be applied for NER in any language without any prior knowledge of that language. The lists or gazetteers are basically language dependent at the lexical level and not at the morphology or syntax level. Also, we include the POS information in the set of language dependent features as the POS information depends on some language specific phenomenon such as person, number, tense, gender etc. Also, the particular POS tagger, used in this work, makes use of the several language specific resources such as lexicon, inflection lists and a NER system to improve its performance. Evaluation results have demonstrated that the use of language specific features is helpful to improve the performance of the NER system. In the resource-constrained Indian language environment, the non-availability of language specific resources acts as a stimulant for the development of such resources for use in NER systems. This leads to the necessity of apriori knowledge of the language. The features are described below very briefly.

- Context words: Such words include the preceding and succeeding words of the current word. This is based on the observation that the surrounding words carry effective information for the identification of NEs.

- Word suffix and prefix: Fixed length word suffixes and prefixes are helpful to identify NEs. In addition, variable length word suffixes are also used. Word suffixes and prefixes are the ef-

fective features and work well for the inflective Indian languages like Bengali.

- Named Entity Information: This is the only dynamic feature in the experiment. The previous word NE tag is very informative in deciding the current word NE tag.

- First word (binary valued): This feature checks whether the current token is the first word of the sentence or not. Though Bengali is a relatively free phrase order language, the first word of the sentence is most likely a NE as it appears most of the time in the subject position.

- Length of the word (binary valued): This feature checks whether the length of the token is less than three or not. We have observed that very short words are most probably not the NEs.

- Infrequent word (binary valued): A cut off frequency has been chosen in order to consider the infrequent words in the training corpus. This is based on the observation that the infrequent words are rarely NEs.

- Digit features: Several digit features have been considered depending upon the presence and/or the number of digit(s) in a token. These binary valued features are helpful in recognizing miscellaneous NEs such as time, monetary and date expressions, percentages, numerical numbers etc.

- Position of the word (binary valued): Position of the word (whether last word or not) in a sentence is a good indicator of NEs.

- Part of Speech (POS) Information: We have used an SVM-based POS tagger (Ekbal and Bandyopadhyay, 2008d) that was originally developed with 26 POS tags, defined for the Indian languages. For SVM models, we have used this POS tagger. However, for the ME and CRF models, we have considered a coarse-grained POS tagger that has the following tags: Nominal, PREP (Postpositions) and Other.

- Gazetteer Lists: Gazetteer lists, developed manually as well as semi-automatically from the news corpus (Ekbal and Bandyopadhyay, 2008c), have been used as the features in each of the classifiers. The set of gazetteers along with the number of entries are as follows:

- (1). Organization clue word (e.g., *ko.m* [Co.], *limited* [Limited] etc): 94, Person prefix words (e.g., *shrimAn* [Mr.], *shrImati* [Mrs.] etc.): 145, Middle names: 2,491, Surnames: 5,288, NE suffixes (e.g., *-bAbu* [-babu], *-dA* [-da], *-di* [-di] for person and *-lyAnd* [-land] *-pur*[-pur], *-liyA* [-lia] etc for location):115, Common location (e.g., *sarani* [Sarani], *roDa* [Road] etc.): 147, Action

verb (e.g., *balen* [says], *ballen* [told] etc.):141, Function words:743, Designation words (e.g., *netA*[leader], *sA.msad* [MP] etc.): 139, First names:72,206, Location names:7,870, Organization names:2,225, Month name (English and Bengali calendars):24, Weekdays (English and Bengali calendars):14

(2). Common word (521 entries): Most of the Indian language NEs appears in the dictionary with some meanings. For example, the word *ka-mol* may be the name of a person but also appears in the dictionary with another meaning *lotus*, the name of a flower; the word *dhar* may be a verb and also can be the part of a person name. We have manually created a list, containing the words that can be NEs as well as valid dictionary words.

3 Active Learning Method for Baseline NER System

We have used a portion, containing 35,143 news documents and approximately 10 million word-forms, of the Bengali news corpus (Ekbal and Bandyopadhyay, 2008c) for developing the *baseline* NER system.

The frequently occurring words have been collected from the *reporter*, *location* and *agency* tags of the Bengali news corpus. The unlabeled corpus is tagged with the elements from the seed lists. In addition, various gazetteers have been used that include surname, middle name, person prefix words, NE suffixes, common location and designations for further tagging of the NEs in the training corpus. The following linguistic rules have been used to tag the training corpus:

(i). If there are two or more words in a sequence that represent the characters of Bengali or English alphabet, then such words are part of NEs. For example, *bi e* (B A), *ci em di e* (C M D A), *bi je pi* (B J P) are all NEs.

(ii). If at the end of a word, there are strings like *-era(-er)*, *-eraa(-eraa)*, *-ra(-ra)*, *-rA(-raa)*, *-ke(-ke)*, *-dera(-der)* then the word is likely to be a person name.

(iii). If a clue word like *saranI* (sarani), *ro.Da* (road), *lena* (lane) etc. is found after an unknown word then the unknown word along with the clue word may be a location name.

(iv). A few names or words in Bengali consist of the characters *chandrabinu* or *khanda ta*. So, if a particular word W is not identified as NE by any of the above rules but includes any of these two characters, then W may be a NE. For example *o.NrI* (*onry*) is a person name.

(v). The set of action verbs like *balen* (says), *ballen* (told), *ballo* (told), *shunla* (heard), *ha.Nslo* (*haslo*) etc. often determines the presence of person names. If an unknown word W appears in the sentence followed by the action verbs, then W is most likely a person name. Otherwise, W is not likely to be a NE.

(vi). If there is reduplication of a word W in a sentence then W is not likely to be a NE. This is so because rarely name words are reduplicated. In fact, reduplicated name words may signify something else. For example, *rAm rAm* (ram ram) is used to greet a person.

(vii). If at the end of any word W there are suffixes like *-gulo(-gulo)*, *-guli(guli)*, *-khAnA(-khana)* etc., then W is not a NE.

For each tag T inserted in the training corpus, the algorithm generates a *lexical pattern p* using a context window of maximum width 6 (excluding the tagged NE) around the left and the right tags, e.g.,

$$p = [l_{-3}l_{-2} l_{-1} \langle T \rangle \dots \langle /T \rangle l_{+1} l_{+2} l_{+3}],$$

where, $l_{\pm i}$ are the *context* of p. All these patterns, derived from the different tags of the labeled and unlabeled training corpora, are stored in a Pattern Table (or, set P), which has four different fields namely, pattern *id* (identifies any particular pattern), pattern *example* (pattern), pattern *type* (*Person/Location/Organization*) and *relative frequency* (indicates the number of times any pattern of a particular *type* appears in the entire training corpus relative to the total number of patterns generated of that *type*). This table has 20,967 distinct entries.

Every pattern p in the set P is matched against the same unlabeled corpus. In a place, where the context of p matches, p predicts the occurrence of the left or right boundary of name. POS information of the words as well as some linguistic rules and/or length of the entity have been used in detecting the other boundary. The extracted entity may fall in one of the following categories:

- *positive example*: The extracted entity is of the same NE *type* as that of the pattern.
- *negative example*: The extracted entity is of the different NE *type* as that of the pattern.
- *error example*: The extracted entity is not at all a NE.

The *type* of the extracted entity is determined by checking whether it appears in any of the seed lists; otherwise, its *type* is determined manually. The *positive* and *negative* examples are then added to the appropriate seed lists. The *accuracy* of the pattern is calculated as follows:

$$accuracy(p) = \frac{|positive(p)|}{(|positive(p)| + |negative(p)| + |error(p)|)}$$

A threshold value of *accuracy* has been chosen in order to discard the patterns below this threshold. A pattern is also discarded if its total *positive count* is less than a predetermined threshold value. The remaining patterns are ranked by their *relative frequency* values. The *n* top high frequent patterns are retained in the pattern set *P* and this set is denoted as *Accept Pattern*.

All the *positive* and *negative* examples extracted by a pattern *p* can be used to generate further patterns from the same training corpus. Each new *positive* or *negative* instance (not appearing in the seed lists) is used to further tag the training corpus. We repeat the previous steps for each new NE until no new patterns can be generated. A newly generated pattern may be identical to a pattern that is already in the set *P*. In such a case, the *type* and *relative frequency* fields in the set *P* are updated accordingly. Otherwise, the newly generated pattern is added to the set with the *type* and *relative frequency* fields set properly. The algorithm terminates after 13 iterations and there are 20,176 distinct entries in the set *P*.

4 Semi-supervised Approach for Unlabeled Document and Sentence Selection

A method for automatically selecting the appropriate unlabeled data from a large collection of unlabeled documents for NER has been described in Ekbal and Bandyopadhyay (2008e). This work reported the selection of unlabeled documents based on the overall *F-Score* value of the individual system. In this work, the unlabeled documents have been selected based on the *Recall*, *Precision* as well as the *F-Score* values of the participating systems. Also, we have considered only the SVM-F model trained with the language independent, language dependent and context features for selecting the appropriate sentences to be included into the initial training data. The use of single model makes the training faster compared to Ekbal and Bandyopadhyay (2008e). The SVM-F model has been considered as it produced the best results for the development set as well as during the 10-fold cross validation test. The unlabeled 35,143 news documents have been divided based on news sources/types in order to create segments of manageable size, separately evaluate the contribution of each segment using a

gold standard development test set and reject those that are not helpful and to apply the latest updated best model to each subsequent segment. It has been observed that incorporation of unlabeled data can only be effective if it is related to the target problem, i.e., the test set. Once the appropriate documents are selected, it is necessary to select the tagged sentences that are useful to improve both the *Recall* and *Precision* values of the system. Appropriate sentences are selected using the SVM-F model depending upon the structure and/or contents of the sentences.

4.1 Unlabeled Document Selection

The unlabeled data supports the acquisition of new names and contexts to provide new evidences to be incorporated in the models. Unlabeled data can degrade rather than improve the classifier's performance on the test set if it is irrelevant to the test document. So, it is necessary to measure the relevance of the unlabeled data to our target test set. We construct a set of key words from the test set *T* to check whether an unlabeled document *d* is useful or not.

- We do not use all words in the test set *T* as the key words since we are only concerned about the distribution of name candidates. So, each document is tested with the CRF model using the language independent features, language dependent features and the context features.
- We take all the name candidates in the top *N* best hypotheses (*N*=10) for each sentence of the test set *T* to construct a query set *Q*. Using this query set, we find all the relevant documents that include three (heuristically set) names belonging to the set *Q*. In addition, the documents are not considered if they contain fewer than seven (heuristic) names.

4.2 Sentence Selection

All the tagged sentences of a relevant document are not added to training corpus as incorrectly tagged or irrelevant sentences can lead to the degradation in model performance. Our main concern is on how much new information is extracted from each sentence of the unlabeled data compared to the training corpus that already we have in our hand.

The SVM-F model has been used to select the relevant sentences. All the relevant documents are tagged with the SVM-F model developed with the language independent, language de-

pendent and context features along with the class decomposition technique. If both *Recall* and *Precision* values of the SVM-F model increase then that sentence is selected to be added to the initial training corpus. A close investigation reveals the fact that this criterion often selects a number of sentences which are too short or do not include any name. These words may make the model worse if added to the training data. For example, the distribution of non-names may increase significantly that may lead to degradation of model performance. In this experiment, we have not included the sentences that include fewer than *five* words or do not include any names. The bootstrapping procedure is given as follows:

1. Select a relevant document *RelatedD* from a large corpus of unlabeled data with respect to the test set T using the document selection method described in Section 4.1.
2. Split *RelatedD* into n subsets and mark them C_1, C_2, \dots, C_n .
3. Call the development set *DevT*.
4. For I=1 to n
 - 4.1. Run SVM-F model, developed with the language independent features, language dependent feature and context features along with the class decomposition technique, on C_i .
 - 4.2. If the length of each tagged sentence S is less than five or it does not contain any name then discard S.
 - 4.3. Add C_i to the training data and retrain SVM-F model. This produces the updated model.
 - 4.4. Run the updated model on *DevT*; if the *Recall* and *Precision* values reduce then don't use C_i and use the old model.
5. Repeat steps 1-4 until *Recall* and *Precision* values of the SVM-F model either become equal or differ by some threshold values (set to 0.01) in consecutive two iterations.

5 Evaluation Results and Discussions

Out of 200K wordforms, 150K wordforms along with the IJCNLP-08 shared task data has been used for training the models. Out of 200K wordforms, 50K wordforms have been used as the development data. The system has been tested with a gold standard test set of 35K wordforms. Each of the models has been evaluated in two different ways, being guided by language independent features (language independent system denoted as LI) and being guided by language

independent as well as language dependent features (language dependent system denoted as LD).

5.1 Language Independent Evaluation

A number of experiments have been carried out in order to identify the best-suited set of language independent features for NER in each of models. Evaluation results of the development set for the NER models are presented in Table 1 in terms of percentages of Recall (R), Precision (P) and F-Score (FS). The ME based system has demonstrated the F-Score value of 74.67% for the context word window of size three, i.e., previous one word, current word and the next word, prefixes and suffixes of length up to three characters of only the current word, dynamic NE tag of the previous word, first word, infrequent word, length and the various digit features. The CRF based system yielded the highest F-Score value of 76.97% for context window of size five, i.e., two preceding, current and two succeeding words along with the other set of features as in the ME model. Both the SVM based systems have demonstrated the best performance for the context window of size seven, i.e., three preceding, current and two succeeding words, dynamic NE information of the previous two words along with the other set of features as in the ME and CRF based systems. In SVM models, we have conducted experiments with the different polynomial kernel functions and observed the highest F-Score value with degree 2. It has been also observed that *pairwise* multiclass decision method performs better than the one vs rest method. For all the models, context words and prefixes and/or suffixes have been found to be the most effective features.

Model	R	P	FS
ME	76.82	72.64	74.67
CRF	78.17	75.81	76.97
SVM-F	79.14	77.26	78.19
SVM-B	79.09	77.15	78.11

Table 1. Results on the development set for the language independent supervised models

5.2 Language Dependent Evaluation

Evaluation results of the systems that include the POS information and other language dependent features are presented in the Table 2. During the experiments, it has been observed that all the language dependent features are not equally important. POS information is the most effective

followed by NE suffixes, person prefix words, designations, organization clue words and location clue words. Table 1 and Table 2 show that the language dependent features can improve the overall performance of the systems significantly.

Model	R	P	FS
ME	87.02	80.77	83.78
CRF	87.63	84.03	85.79
SVM-F	87.74	85.89	86.81
SVM-B	87.69	85.17	86.72

Table 2. Results on the development set for the language dependent supervised models

5.3 Use of Context Features as Features

Now, the high ranked patterns of the *Accept Pattern* set (Section 3) can be used as the features of the individual classifier. A feature ‘ContextInf’ is defined by observing the three preceding and succeeding words of the current word. Evaluation results are presented in Table 3. Clearly, it is evident from the results of Table 2 and Table 3 that context features are very effective to improve the *Precision* values in each of the models.

Model	R	P	FS
ME	88.22	83.71	85.91
CRF	89.51	85.94	87.69
SVM-F	89.67	86.49	88.05
SVM-B	89.61	86.47	88.01

Table 3. Results on the development set by including context features

5.4 Results on the Test Set

A gold standard test set of 35K wordforms has been used to report the evaluation results. The models have been trained with the language independent, language dependent and the context features. Results have been presented in Table 4 for the test set. In the *baseline* model, each pattern of the *Accept Pattern* set is matched against the test set. Results show that SVM-F model performs best for the test set.

Error analyses have been conducted with the help of confusion matrix. In order to improve the performance of the classifiers, we have used some post-processing techniques.

Output of the ME based system has been post-processed with a set of heuristics (Ekbal and Bandyopadhyay, 2009) to improve the performance further. The post-processing as described in Ekbal and Bandyopadhyay (2008e) tries to assign the correct tag according to the n-best re-

sults for every sentence of the test set in the CRF framework. In order to remove the unbalanced class distribution between names and non-names in the training set, we have considered the class decomposition technique (Ekbal and Bandyopadhyay, 2008e) for SVM. Evaluation results of the post-processed systems are presented in Table 5.

Model	R	P	FS
Baseline	68.11	71.37	69.32
ME	86.04	84.98	85.51
CRF	87.94	87.12	87.53
SVM-F	89.91	85.97	87.89
SVM-B	89.82	85.93	87.83

Table 4. Results on the test set

Model	R	P	FS
ME	87.29	86.81	87.05
CRF	89.19	88.85	89.02
SVM-F	90.23	88.62	89.41
SVM-B	90.05	88.61	89.09

Table 5. Results of the post-processed models on the test set

Each of the models has been also evaluated for the 10-fold cross validation tests. Initially all the models have been developed with the language independent features along with the context features. Then, language dependent features have been included into the models. In each run of the 10 tests, the outputs have been post-processed with the several post-processing techniques as described earlier. Results are shown in Table 6.

	Model	R	P	FS
LI	ME	81.34	79.01	80.16
	CRF	82.66	80.75	81.69
	SVM-F	83.87	81.83	82.83
	SVM-B	83.87	81.77	82.62
LD	ME	87.54	87.97	87.11
	CRF	89.5	88.73	89.19
	SVM-F	89.97	88.61	89.29
	SVM-B	89.76	88.51	89.13

Table 6. Results of the 10-fold cross validation tests

Statistical ANOVA tests (Anderson and Scolve, 1978) demonstrated that the performance improvement in each of the language dependent model is statistically significant over the language independent model. We have also carried out the statistical tests to show that performance improvement in CRF over ME and SVM-F over CRF are statistically significant.

5.5 Impact of Unlabeled Data Selection

In order to investigate the contribution of document selection in bootstrapping, the post-processed models are run on 35,143 news documents. This yields the gradually improving performance for the SVM-F model as shown in Table 7. After selection of the appropriate unlabeled data, all the models have been retrained by including the unlabeled documents. Results have been presented in Table 8.

Iteration	Sentences added	R	P	FS
0	0	89.97	88.61	89.29
1	129	90.19	88.97	89.58
2	223	90.62	89.14	89.87
3	332	90.89	89.73	90.31
4	416	91.24	90.11	90.67
5	482	91.69	90.65	91.16
6	543	91.88	90.97	91.42
7	633	92.07	91.05	91.56
8	682	92.33	91.31	91.82
9	712	92.52	91.39	91.95
10	723	92.55	91.44	91.99
11	729	92.57	91.45	92.01
12	734	92.58	91.45	92.01

Table 7. Incremental improvement of performance

Model	R	P	FS
ME	90.7	89.78	90.24
CRF	92.02	91.66	91.84
SVM-B	92.34	91.42	91.88
SVM-F	92.58	91.45	92.01

Table 8. Results after unlabeled data selection

5.6 Voting Techniques

In order to obtain higher performance, we have applied weighted voting to the four models. We have used the following weighting methods:

(1). Uniform weights (Majority voting): All the models are assigned the same voting weight. The combined system selects the classifications, which are proposed by the majority of the models. In case of a tie, the output of the SVM-F model is selected. The output of the SVM-F model has been selected due to its highest performance among all the models.

(2). Cross validation *Precision* values: Two different types of weights have been defined depending on the 10-fold cross validation *Precision* on the training data as follows:

(a). Total *Precision*: In this method, the overall average *Precision* of any classifier is assigned as the weight for it.

(b). Tag *Precision*: In this method, the average *Precision* value of the individual tag is assigned as the weight for the corresponding model.

Experimental results of the voted system are presented in Table 9. Evaluation results show that the system achieves the highest performance for the voting scheme ‘Tag *Precision*’. Voting shows (Tables 8-9) an overall improvement of **2.74%** over the least performing ME based system and **0.97%** over the best performing SVM-F system. This also shows an improvement of 23.66% *F-Score* over the *baseline* model.

Voting	R	P	FS
Majority	92.59	91.47	92.03
Total <i>Precision</i>	93.08	91.79	92.43
Tag <i>Precision</i>	93.81	92.18	92.98

Table 9. Results of the voted system

6 Conclusion

In this paper, we have reported a voted system with the use of appropriate unlabeled data. We have also demonstrated how language dependent features can improve the system performance. It has been experimentally verified that effective measures to select relevant documents and useful labeled sentences are important. The system has demonstrated the overall *Recall*, *Precision*, and *F-Score* values of 93.81%, 92.18%, and 92.98%, respectively.

Future works include the development of NER system using other machine learning techniques such as decision tree, AdaBoost etc. We would like to apply the proposed voted technique for the development of NER systems in other Indian languages. Future direction of the work will be to investigate an appropriate clustering technique that can be very effective for the development of NER systems in the resource-constrained Indian language environment. Instead of the words, the cluster of words can be used as the features of the classifiers. It may reduce the cost of training as well as may be helpful to improve the performance. We would like to explore other voting techniques.

References

- Anderson, T. W. and Scolve, S. Introduction to the Statistical Analysis of Data. *Houghton Mifflin*, 1978.
- Bikel, Daniel M., R. Schwartz, Ralph M. Weischedel. 1999. An Algorithm that Learns What's in Name. *Machine Learning (Special Issue on NLP)*, 1-20.
- Bothwick, Andrew. 1999. A Maximum Entropy Approach to Named Entity Recognition. *Ph.D. Thesis*, NYU.
- Ekbal, Asif, Naskar, Sudip and S. Bandyopadhyay. 2007b. Named Entity Recognition and Transliteration in Bengali. *Named Entities: Recognition, Classification and Use, Special Issue of Linguisticae Investigationes Journal*, 30:1 (2007), 95-114.
- Ekbal, Asif, Haque, R and S. Bandyopadhyay. 2008a. Named Entity Recognition in Bengali: A Conditional Random Field Approach. In *Proceedings of 3rd International Joint Conference on Natural Language Processing (IJCNLP-08)*, 589-594.
- Ekbal, Asif, and S. Bandyopadhyay. 2008b. Bengali Named Entity Recognition using Support Vector Machine. In *Proceedings of the Workshop on Named Entity Recognition on South and South East Asian Languages (NERSSEAL)*, IJCNLP-08, 51-58.
- Ekbal, Asif, and S. Bandyopadhyay. 2008c. A Web-based Bengali News Corpus for Named Entity Recognition. *Language Resources and Evaluation Journal*, Volume (40), 173-182.
- Ekbal, Asif and S. Bandyopadhyay. 2008d. Web-based Bengali News Corpus for Lexicon Development and POS Tagging. In *POLIBITS, an International Journal*, Volume (37), 20-29, ISSN: 1870-9044.
- Ekbal, Asif and S. Bandyopadhyay. 2008e. Appropriate Unlabeled Data, Post-processing and Voting Can Improve the Performance of NER System. In *Proceedings of the 6th International Conference on Natural Language Processing (ICON-08)*, 234-239, India.
- Ekbal, Asif and S. Bandyopadhyay. 2009. Improving the Performance of a NER System by Post-processing, Context Patterns and Voting. In *W. Li and D. Molla-Aliod (Eds): ICCPOL 2009, Lecture Notes in Artificial Intelligence (LNAI), Springer Berlin/Heidelberg, Volume (5459)*, 45-56.
- Florian, Radu, Ittycheriah, A., Jing, H. and Zhang, T. 2003. Named Entity Recognition through Classifier Combination. In *Proceedings of CoNLL-2003*.
- Lafferty, J., McCallum, A., and Pereira, F. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of 18th International Conference on Machine Learning (ICML)*, 282-289.
- Li, Wei and Andrew McCallum. 2003. Rapid Development of Hindi Named Entity Recognition Using Conditional Random Fields and Feature Inductions. *ACM TALIP*, 2(3), (2003), 290-294.
- Saha, Sujan, Sarkar, S and Mitra, P. 2008. A Hybrid Feature Set based Maximum Entropy Hindi Named Entity Recognition. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP-08)*, 343-349.
- Valdimir N., Vapnik 1995. The Nature of Statistical Learning Theory. *Springer*.
- Yamada, Hiroyasu, Taku Kudo and Yuji Matsumoto. 2002. Japanese Named Entity Extraction using Support Vector Machine. In *Transactions of IPSJ*, Vol. 43 No. 1, 44-53.