Finding Appropriate Subset of Votes Per Classifier Using Multiobjective Optimization: Application to Named Entity Recognition

Asif Ekbal [1*], Sriparna Saha [1*] and Md. Hasanuzzaman[2]

[1] Heidelberg University, 69120 Heidelberg, Germany
Email:asif.ekbal@gmail.com, sriparna.saha@gmail.com
[2] West Bengal Industrial Development Corporation, Kolkata, India
Email: hasanuzzaman.im@gmail.com
[*]First two authors are the joint first authors.

Abstract. In this paper, we report a multiobjective optimization (MOO) based technique to select the appropriate subset of votes per classifier in an ensemble system. We hypothesize that the reliability of prediction of each classifier differs among the various output classes. Thus, it is necessary to find out the subset of classes for which any particular classifier is most suitable. Rather than optimizing a single measure of classification quality, we simultaneously optimize two different measures of classification quality using the search capability of MOO. We use our proposed technique to solve the problem of Named Entity Recognition (NER). Maximum Entropy (ME) model is used as a base to build a number of classifiers depending upon the various representations of the contextual, orthographic word-level and semantically motivated features. Evaluation results with a resource constrained language like Bengali yield the recall, precision and F-measure values of 87.98%, 93.00%, and 90.42%, respectively. Experimental results suggest that the use of semantic feature can significantly improve the overall system performance. Results also reveal that the classifier ensemble identified by the proposed MOO based approach performs better in comparison to the individual classifiers, two different *baseline* ensembles and the classifier ensemble identified by a single objective genetic algorithm (GA) based approach.

1 Introduction

Named Entity Recognition (NER) is an important pipelined module in many Natural Language Processing (NLP) application areas that include machine translation, information retrieval, information extraction, question-answering, automatic summarization etc. Machine learning approaches are popularly being used for NER due to their flexible adaptation to new domains and languages. Most of the existing works in NER cover the languages such as English, European languages and some of the Asian languages like Chinese, Japanese and Korean. India is a multilingual country with great linguistic and cultural diversities. In India, there are 22 official languages that are inherited from almost all the existing linguistic families in the world. However, the works related to NER in Indian languages have started to emerge only very recently. Named Entity (NE) identification in Indian languages in general and Bengali in particular is more difficult and challenging compared to others due to facts such as: (i). missing of capitalization information, (ii). appearance of NEs in the dictionary with some other specific meanings, (iii). free word order nature of the languages, (iv). resource-constrained environment, i.e. non-availability of corpora, annotated corpora, name dictionaries, good morphological analyzers, part of speech (POS) taggers etc. Some of the recent works related to Bengali NER can be found in (Ekbal and Bandyopadhyay, 2009b; Ekbal and Bandyopadhyay, 2009a; Ekbal and Bandyopadhyay, 2008b). Other works related to Indian language NER are reported in the proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages (NERSSEAL)[1].

[1] http://ltrc.iiit.ac.in/ner-ssea-08

The concept of combining classifiers is a very emerging topic in the area of machine learning. The primary goal of classifier ensemble [2] is to improve the performance of the individual classifiers. These classifiers could be based on a variety of classification methodologies, and could achieve different rate of correctly classified individuals. But, the appropriate classifier selection for constructing an ensemble remains a difficult problem. Moreover, all the classifiers are not equally good to detect all types of output classes. Thus, in a voted system, a particular classifier should only be allowed to vote for that output class for which it performs good. Therefore, selection of appropriate votes per classifier is a very crucial issue. Some single objective optimization techniques like genetic algorithm (GA) has been used to determine the appropriate vote combinations per classifier (Ekbal *et al.*, 2010). But, these single objective optimization techniques can only optimize a single quality measure, e.g. recall, precision or F-measure at a time. But sometimes, a single measure cannot capture the quality of a good ensembling reliably. A good voted classifier ensemble for NER should have its all the parameters optimized simultaneously. In order to achieve this, we use a multiobjective optimization (MOO) technique (Deb, 2001) that is capable of simultaneously optimizing more than one classification quality measures. Experimental results also justify our assumption that MOO can perform superior to the single objective approach for voting combination selection.

The proposed MOO based voting combination selection technique is applied to solve the problem of Named Entity Recognition (NER). We use Maximum Entropy (ME) as a base classifier. Depending on the various feature combinations, several different versions of this classifier are made. The features include contextual information of the words, orthographic word-level features, semantically motivated feature and the various features extracted from the gazetteers. Thereafter, a MOO technique based on a popular multiobjective evolutionary algorithm (MOEA), non-dominated sorting GA-II (NSGA-II) (Deb *et al.*, 2002), is used to search for the appropriate voting combination selection. The proposed MOO based approach searches for an appropriate subset of predictions per classifier which are considered to be relevant enough in the process of final output selection.

Our proposed technique is very general and can be applicable for any language and/or domain. Here, the technique is evaluated for a resource-constrained language, namely Bengali. In terms of native speakers, Bengali is the *fifth* popular language in the world, *second* in India and the *national* language in Bangladesh. Evaluation results show the effectiveness of the proposed approach with the recall, precision and F-measure values of 87.98%, 93.00%, and 90.42%, respectively. Results show the superiority of the proposed MOO based ensemble technique in comparison to the best individual classifier, two different *baseline* ensembles and a single objective GA based ensemble technique (Ekbal *et al.*, 2010). These results are also supported by the sufficient statistical analysis.

The remainder of the paper is organized as follows. The ME framework for NER is discussed briefly in Section 2. Section 3 describes in brief the definition of MOO and a popular way to solve this type of problem. The problem of vote based classifier ensemble is formulated under the MOO framework in Section 4. Section 5 describes different features that include contextual information of the words, several word-level orthographic features, semantic feature and various features extracted from the gazetteers. The proposed MOO based classifier ensemble selection approach is presented in Section 6. Section 7 reports the datasets, evaluation results and necessary discussions. Finally, Section 8 concludes the paper.

2 Maximum Entropy Framework for NER

The Maximum Entropy (ME) framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived

[2] Henceforth, we use ‘classifier ensemble’ and ‘ensemble classifier’ interchangeably

from the training data, expressing some relationships between features and outcome. The probability distribution that satisfies the above property is the one with the highest entropy. It is unique, agrees with the maximum likelihood distribution, and has the exponential form

$$P(t|h) = \frac{1}{Z(h)} exp(\sum_{j=1}^{n} \lambda_j f_j(h,t)) \tag{1}$$

where, t is the NE tag, h is the context (or history), $f_j(h,t)$ are the features with associated weight λ_j and $Z(h)$ is a normalization function.

The problem of NER can be formally stated as follows. Given a sequence of words $w_1, \ldots, w_n$, we want to find the corresponding sequence of NE tags $t_1, \ldots, t_n$, drawn from a set of tags T, which satisfies:

$$P(t_1, \ldots, t_n | w_1, \ldots, w_n) = \prod_{i=1,2\ldots,n} P(t_i|h_i) \tag{2}$$

where, h_i is the context for the word w_i.

The features are, in general, binary valued functions, which associate a NE tag with various elements of the context. For example:

$$\begin{aligned} f_j(h,t) &= 1 \text{ if word}(h) = \text{sachIn and } t = \text{I-PER} \\ &= 0 \quad \text{otherwise} \end{aligned}$$

We use the OpenNLP Java based MaxEnt package [3] for the computation of the values of the parameters λ_j. This allows to concentrate on selecting the features, which best characterize the problem instead of worrying about assigning the relative weights to the features. Selecting an optimal model subject to given constrains from the exponential (log-linear) family is not a trivial task. There are two popular iterative scaling algorithms specially designed to estimate parameters of ME models: Generalized Iterative Scaling (Darroch and Ratcliff, 1972) and Improved Iterative Scaling (Pietra *et al.*, 1997). In the present work, we use the Generalized Iterative Scaling (Darroch and Ratcliff, 1972) algorithm to estimate the MaxEnt parameters.

3 Multiobjective Algorithms

The multiobjective optimization (MOO) can be formally stated as follows (Deb, 2001). Find the vectors $\overline{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of decision variables that simultaneously optimize the *M* objective values $\{f_1(\overline{x}), f_2(\overline{x}), \ldots, f_M(\overline{x})\}$, while satisfying the constraints, if any.

3.1 Nondominated Sorting Genetic Algorithm-II (NSGA-II)

Genetic algorithms are known to be more effective than classical methods such as weighted metrics, goal programming (Deb, 2001), for solving multiobjective problems primarily because of their population-based nature. NSGA-II (Deb *et al.*, 2002) is widely used in this regard, where initially a random parent population P_0 is created and the population is sorted based on the *partial order* defined by the non-domination relation. This results in a sequence of nondominated fronts. Each solution of the population is assigned a fitness which is equal to its non-domination level in the partial order. A child population Q_0 of size N is created from the parent population P_0 by using binary tournament selection, recombination, and mutation operators. According to this algorithm, in the t^{th} iteration, a combined population R_t = $P_t + Q_t$ is formed. The size of R_t is $2N$. All the solutions of R_t are sorted according to non-domination. If the total number of solutions belonging to the best nondominated set F_1 is smaller than N, then F_1 is totally included

[3] http://maxent.sourceforge.net/

in $P_{(t+1)}$. The remaining members of the population $P_{(t+1)}$ are chosen from subsequent nondominated fronts in the order of their ranking. To choose exactly N solutions, the solutions of the last included front are sorted using the crowded comparison operator (Deb *et al.*, 2002) and the best among them (i.e., those with lower crowding distance) are selected to fill in the available slots in $P_{(t+1)}$. The new population $P_{(t+1)}$ is then used for selection, crossover and mutation to create a population $Q_{(t+1)}$ of size N. The pseudocode of NSGA-II is provided in Figure 1.

NSGA-II

- Step 1: Combine parent and offspring populations and create $R_t = P_t \cup Q_t$. Perform a nondominated sort on R_t and identify different fronts: F_i, $i = 1, 2 \ldots$,etc.
- Step 2: Set new population $P_{t+1} = \emptyset$. Set a counter $i = 1$.
- Step 3: Perform the *Crowding-sort* procedure and include the most widely spread $(N - |P_{t+1}|)$ solutions by using the crowding distance values in the sorted F_i to P_{t+1}.
- Step 4: Create offspring population Q_{t+1} from P_{t+1} by using the crowded tournament selection, crossover and mutation operators.

Figure 1: Main steps of NSGA-II

4 Problem Formulation

In this section, we formulate the vote based classifier ensemble problem under the MOO framework.
Let, the N number of available classifiers be denoted by $C_1, \ldots, C_N$ and $\mathcal{A} = \{C_i : i = 1; N\}$. Suppose, there are M number of output classes. The vote based classifier ensemble selection problem is then stated as follows:

Find the combination of votes per classifier V such that:
$maximize\ \ [F_1(B), F_2(B)]$
$where,\ F_1, F_2 \in \{\text{recall}, \text{precision}, \text{F-measure}\}\ \ and\ \ F_1 \neq F_2$.
Here, V is a boolean array of size $N \times M$. $V(i,j)$ denotes the decision whether the i^{th} classifier is allowed to vote for j^{th} class. $V(i,j) = true/1$ denotes that the i^{th} classifier is allowed to vote for j^{th} class; else $V(i,j) = false/0$ denotes that the i^{th} classifier is not allowed to vote for j^{th} class. Here, F_1 and F_2 are some classification quality measures of the combined vote based classifier ensemble. The particular type of problem like NER has mainly three different kinds of classification quality measures, namely recall, precision and F-measure. Thus, $F \in \{\text{recall}, \text{precision}, \text{ F-measure}\}$. Combination of the classifiers can be done by either majority voting or weighted voting. Here, we choose $F_1 = $ recall and $F_2 = $ precision.

Selection of Objectives. Performance of MOO largely depends on the choice of the objective functions which should be as much contradictory as possible. In this work, we choose recall and precision as two objective functions. From the definitions, it is clear that while recall tries to increase the number of tagged entries as much as possible, precision tries to increase the number of correctly tagged entries. These two capture two different classification qualities. Often, there is an inverse relationship between recall and precision, where it is possible to increase one at the cost of reducing the other. For example, an information retrieval system (such as a search engine) can often increase its recall by retrieving more documents, at the cost of increasing number of irrelevant documents retrieved (i.e. decreasing precision). This is the underlying motivation of simultaneously optimizing these two objectives.

5 Named Entity Features

We use the following features for constructing the various classifiers based on the ME framework.

1. **Context words**: These are the preceding and succeeding words of the current word.
2. **Word suffix and prefix**: Fixed length (say, n) word suffixes and prefixes are very effective to identify NEs and work well for the highly inflective Indian language like Bengali. Actually, these are the fixed length character sequences stripped from either the rightmost or leftmost positions of the words.
3. **First word**: This is a binary valued feature that checks whether the current token is the first word of the sentence or not. We consider this feature with the observation that the first word of the sentence is most likely a NE, especially in a newspaper corpus.
4. **Length of the word**: This binary valued feature checks whether the length of the token is less than a predetermined threshold (set to 5) value and based on the observation that very short words are most probably not the NEs.
5. **Infrequent word**: A cut off frequency (set to 10) is chosen to consider the infrequent words in the training corpus with the observation that very frequent words are rarely NEs. A binary valued feature 'INFRQ' fires if the current word appears in this list.
6. **Part of Speech (POS) information**: POS information of the current and/or the surrounding word(s) are extracted using a SVM based POS tagger (Ekbal and Bandyopadhyay, 2008a). In the present work, we evaluate this POS tagger with a coarse-grained tagset of three tags, namely Nominal, PREP (Postpositions) and Other. The coarse-grained POS tagger is found to perform better compared to a fine-grained one.
7. **Position of the word**: This binary valued feature checks the position of the word in the sentence. Sometimes, position of the word in a sentence acts as a good indicator for NE identification.
8. **Digit features**: Several digit features (digitComma, digitPercentage etc.) are defined depending upon the presence and/or the number of digits and/or symbols in a token. This feature is useful for identifying miscellaneous NEs.
9. **Dynamic NE information**: The NE class information of the previous token is used as the feature. This is determined dynamically during run time.
10. **Semantic feature**: This feature is semantically motivated. We consider all unigrams in contexts $w_{i-3}^{i+3} = w_{i-3} \dots w_{i+3}$ of w_i (crossing sentence boundaries) for the entire training data. We convert tokens to lower case, remove stop-words, numbers and punctuation symbols. We define a feature vector of length 10 using the 10 most frequent content words. Given a classification instance, the feature corresponding to token t is set to 1 iff the context w_{i-3}^{i+3} of w_i contains t.
11. **Gazetteer based features**: Various features are extracted from the following gazetteer lists:
    (a). NE Suffix list (55 entries): A list of variable length NE suffixes is prepared. These are helpful to detect person (e.g., *-bAbU*, *-dA*, *-di* etc.) and location (e.g., *-lyAnDa*, *-pUra*, *-liYA* etc.) names.
    (b). Organization suffix word list (94 entries): This list contains the words that are helpful to identify organization names (e.g., *kO.m*[Co.], *limiteDa*[limited] etc.). These are also the part of organization names.
    (c). Person prefix word list (67 entries): This is useful for detecting person names (e.g., *shrImAna*[Mr.], *shrI*[Mr.], *shrImati*[Mrs.] etc.). Person name generally appears after these words.
    (d). Common location word list (147 entries): This list contains the words (e.g., *saranI*,

rOda, *lena* etc.) that are part of the multiword location names and usually appear at their end.
(e). Action verb list (53 entries): A set of action verbs like *balena*[told], *balalena*[told], *ballO*[says], *sUnllO*[hears], *h.AsalO*[smiles] etc. often determines the presence of person names. Person names generally appear before the action verbs.
(f). Designation words (62 entries): A list of common designation words (e.g., *netA*[leader], *sA.msada*[MP], *khelOYAra*[player] etc.) has been prepared. This helps to identify the position of person names.
(g). Name lists: Three different lists for person, location and organization are prepared that contain 72,206, 4,875 and 2,225 entries, respectively.
(h). Measurement expressions (24 entries): This contains the words that denote various measurement expressions like weight, distance etc.

6 Multiobjective GA for Vote based Classifier Ensemble

A multiobjective GA, along the lines of NSGA-II (Deb, 2001), is proposed for solving the voting combination selection problem. Note, that although the proposed approach has some similarity in steps with NSGA-II, any other existing multiobjective GAs could have been also used as the underlying MOO technique.

6.1 Chromosome Representation and Population Initialization

If the total number of available classifiers is M and total number of output tags (i.e., NE classes) is O, then the length of the chromosome is $M \times O$ (each chromosome encodes the votes for possible O tags for each classifier). As an example, the encoding of a particular chromosome is represented in Figure 2. Here, $M = 3$ and $O = 4$ (i.e., total 12 votes can be possible). The chromosome represents the following voting ensemble:
Classifier 1 is allowed to vote for classes 1 and 4;
Classifier 2 is allowed to vote for classes 1 and 2;
Classifier 3 is allowed to vote for classes 2, 3 and 4.

The entries of each chromosome are randomly initialized to either 0 or 1. Here, if the i^{th} position of a chromosome is 0 then it represents that $(i/4+1)^{th}$classifier is not allowed to vote for the $(i \bmod 4)^{th}$ class. Else, if it is 1 then it means that $(i/4+1)^{th}$ classifier is allowed to vote for the $(i \bmod 4)^{th}$ class. If the population size is P then all the P number of chromosomes of this population are initialized in the above way.

6.2 Fitness Computation

Initially, the F-measure values of all the ME based classifiers are calculated using 3-fold cross validation on the available training data. Thereafter, we execute the following steps to compute the objective values.

1. Suppose, there are total M number of classifiers. Let, the overall F-measure values of these M classifiers be F_i, $i = 1 \ldots M$.
2. Initially, the training data is divided into 3 parts. Each classifier is trained using 2/3 of the training set and tested with the remaining 1/3 part. We have M tags (each from a different classifier) for each word in the 1/3 training data. Now for the ensemble classifier, the output class label for each word in the 1/3 training data is determined using the weighted voting of these M classifiers' outputs. The weight of the output class (or, tag) provided by the i^{th} classifier is equal to F_i. The combined score of a particular class for a particular word w is:

$$f(c_i) = \sum F_m \times I(m,i),$$

$$\forall m = 1 \text{ to } M \text{ and } op(w,m) = c_i$$

Here, $I(m,i)$ is the entry of the chromosome corresponding to the m^{th} classifier and i^{th} class; and $op(w,m)$ denotes the output NE class provided by the classifier m for the word w. The class receiving the maximum combined score is selected as the joint decision.

3. The overall recall and precision values of the ensemble classifier for the 1/3 training data are calculated.
4. Steps 2 and 3 are repeated 3 times to perform 3-fold cross validation. The average recall and precision values of 3-fold cross validation of the ensemble classifier are used as the two objective functions of the proposed MOO technique. Thus, the objective functions corresponding to a particular chromosome are $f_1 = \text{recall}_{avg}$ and $f_2 = \text{precision}_{avg}$. The objective is to: $max[f_1, f_2]$. These two objective functions are simultaneously optimized using the search capability of NSGA-II.

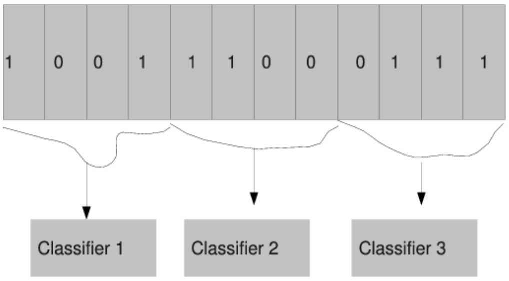

Figure 2: Chromosome Representation

6.3 Other Operators

We use crowded binary tournament selection as in NSGA-II, followed by conventional crossover and mutation. The most characteristic part of NSGA-II is its elitism operation, where the non-dominated solutions (Deb, 2001) among the parent and child populations are propagated to the next generation. The near-Pareto-optimal strings of the last generation provide the different solutions to the vote based classifier ensemble problem.

6.4 Selection of a Solution from the Final Pareto Optimal Front

In MOO, the algorithms produce a large number of non-dominated solutions (Deb, 2001) on the final Pareto optimal front. Each of these solutions provides a vote based classifier ensemble. All the solutions are equally important from the algorithmic point of view. But, sometimes the user may require only a single solution. Consequently, in this paper a method of selecting a single solution from the set of solutions is now developed. For every solution on the final Pareto optimal front, the average F-measure value of the classifier ensemble is computed from the 3-fold cross validation on the training data. The solution with the maximum F-measure value is selected as the best solution. Note, that there can be many other different approaches of selecting a solution from the final Pareto optimal front.

7 Datasets, Results and Discussions

For NER, we use a Bengali news corpus (Ekbal and Bandyopadhyay, 2008b), developed from the archive of a leading Bengali newspaper available in the web. We set the following parameter values for NSGA-II:
population size=100, number of generations=50, probability of mutation=0.2 and probability of crossover=0.9. Following two *baseline* classifier ensemble techniques are defined:

1. *Baseline 1*: In this *baseline* model, all the individual classifiers are combined together into a final system based on the majority voting of the output class labels.
2. *Baseline 2*: This is a weighted voting approach. In each classifier, weights are calculated based on the average F-measure value of the 3-fold cross validation test on the training data.

7.1 Datasets for NER

A portion of the corpus (Ekbal and Bandyopadhyay, 2008b) containing approximately 250K wordforms is manually annotated with a coarse-grained NE tagset of four tags namely, PER (*Person name*), LOC (*Location name*), ORG (*Organization name*) and MISC (*Miscellaneous name*). The miscellaneous name includes date, time, number, percentages, monetary and measurement expressions. The data is collected mostly from the *National*, *States*, *Sports* domains and the various sub-domains of *District* of the particular newspaper. This annotation was carried out by one of the authors and verified by an expert. We also use the IJCNLP-08 NER on South and South East Asian Languages (NERSSEAL)[4] Shared Task data of around 100K wordforms that were originally annotated with a fine-grained tagset of twelve tags. This data is mostly collected from the *agriculture* and *scientific* domains. For evaluation, we randomly partition the dataset into training and test sets. During experiment, a portion of the training set is used as the development set. Some statistics of training and test sets are presented below:
Total number of wordforms in training set: 312,947, Total number of NEs in training set: 37,009, Total number of wordforms in test set: 37,053, Total number of NEs in test set: 4,413, Unknown NEs in test set : 35.1%.

In order to properly denote the boundaries of NEs, four basic NE tags are further divided into the format, I-TYPE (TYPE→PER/LOC/ORG/MISC), which means that the word is inside a NE of type TYPE. Only if two NEs of the same type immediately follow each other, the first word of the second NE will have tag B-TYPE to show that it starts a new NE. This is the standard IOB format that was followed in the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003). Other than NEs are denoted by ’O’.

7.2 Results and Discussions

We build a number of different ME models by considering the various combinations of the available NE features. In this particular work, we construct the classifiers from the following set of features:
various context window within the preceding three and succeeding three words, word suffixes and prefixes of length upto three (3+3 different features) or four (4+4 different features) characters, POS information of the current word, first word, length, infrequent word, position of the word in the sentence, several digit features, semantic feature, gazetteers, and dynamic NE information.

We generate 152 different classifiers varying the different available features. Some of these classifiers are shown in Table 1. Initially, the system is tuned on the development set and blind evaluation is performed on the test set. Here, we report all the results only on the test set. The best individual classifier shows the recall, precision and F-measure values of 86.82%, 90.28% and 88.52%, respectively. Thereafter, we apply our proposed MOO based approach to determine the appropriate classifier ensemble. Overall evaluation results of this ensemble along with the best individual classifier, two different *baseline* ensembles, and the single objective based approach (Ekbal *et al.*, 2010) are reported in Table 2. Results show that the proposed approach performs the best. We observe the improvement of 1.90%, 1.64% and 1.58% F-measures over the best individual classifier, *Baseline 1* and *Baseline 2*, respectively. The proposed approach also performs superior to the single objective based approach with an increment of 1.25 percentage F-measure points.

[4] http://ltrc.iiit.ac.in/ner-ssea-08

Table 1: Evaluation results with various feature types. Here, the following abbreviations are used: ’CW’:Context words, ’PS’: Size of the prefix, ’SS’: Size of the suffix, ’WL’: Word length, ’IW’: Infrequent word, ’PW’: Position of the word, ’FW’:First word, ‘DI’: Digit-Information, ‘NE’: Dynamic NE information, ‘Sem’: Semantic feature, ‘Gaz.’: Gazzetters, ‘R’: recall,‘P’: precision, ‘F’: F-measure, -i,j: Denotes the words spanning from the i^{th} left position the j^{th} right position with the current word being at 0^{th} position, X: Denotes the presence of the corresponding feature (we report percentages)

Classifier	CW	FW	PS	SS	WL	IW	PW	DI	POS	NE	Sem	Gaz.	R	P	F
M_9	-2,2	X	3	-	-	-	-	X	X	X	X	X	85.69	89.92	87.75
M_{10}	-2,1	X	3	-	-	-	-	X	X	X	X	X	85.92	89.87	87.85
M_{12}	-1,1	X	3	-	-	-	-	X	X	X	X	X	86.05	88.96	87.48
M_{13}	-1,2	X	3	-	-	-	-	X	X	X	X	X	86.03	89.70	87.83
M_{17}	-2,2	X	3	3	-	-	-	X	X	X	X	X	86.87	90.09	88.45
M_{18}	-2,1	X	3	3				X	X	X	X	X	86.82	90.28	88.52
M_{19}	-2,0	X	3	3	-	-	-	X	X	X	X	X	85.92	89.36	87.60
M_{19}	-2,0	X	3	3	-	-	-	X	X	X	X	X	85.92	89.36	87.60
M_{20}	-1,1	X	3	3	-	-	-	X	X	X	X	X	86.48	88.63	87.54
M_{21}	-1,2	X	3	3	-	-	-	X	X	X	X	X	87.12	89.52	88.30
M_{22}	0,2	X	3	3	-	-	-	X	X	X	X	X	86.76	88.69	87.71
M_{24}	-3,3	X	3	3	-	-	-	X	X	X	X	X	86.12	90.10	88.07
M_{57}	-2,2	X	4	3	-	-	-	X	X	X	X	X	85.44	90.23	87.77
M_{58}	-2,1	X	4	3	-	-	-	X	X	X	X	X	85.62	90.15	87.83
M_{60}	-1,1	X	4	3	-	-	-	X	X	X	X	X	85.71	89.09	87.37
M_{61}	-1,2	X	4	3	-	-	-	X	X	X	X	X	85.71	89.84	87.73
M_{65}	-2,2	X	3	4	-	-	-	X	X	X	X	X	85.80	89.89	87.80
M_{66}	-2,1	X	3	4	-	-	-	X	X	X	X	X	86.21	89.87	88.00
M_{67}	-2,0	X	3	4	-	-	-	X	X	X	X	X	85.46	89.05	87.22
M_{68}	-1,1	X	3	4	-	-	-	X	X	X	X	X	85.78	88.56	87.15
M_{69}	-1,2	X	3	4	-	-	-	X	X	X	X	X	86.17	89.52	87.81
M_{72}	-3,3	X	3	4	-	-	-	X	X	X	X	X	85.19	89.74	87.41

Table 2: Overall results for Bengali

Classification Scheme	recall (in %)	precision (in %)	F-measure (in %)
Best individual classifier	86.82	90.28	88.52
Baseline 1	85.78	92.00	88.78
Baseline 2	85.89	92.00	88.84
GA based approach	86.42	92.11	89.17
MOO based approach	87.98	93.00	90.42

Statistical analysis of variance, (ANOVA) (Anderson and Scolve, 1978), is performed in order to examine whether MOO really outperforms the best individual classifier and other ensembles. Here, all the classifiers, GA based ensemble (Ekbal *et al.*, 2010) and the proposed MOO based ensemble are executed 10 times. Thereafter, ANOVA analysis is carried out on these outputs. ANOVA tests show that the differences in mean recall, precision and F-measure are statistically significant as p value is less than 0.05 in each of these cases.

8 Conclusion

In this paper, we have posed the problem of finding suitable vote based classifier ensemble for NER under the MOO framework that simultaneously optimizes more than one objective functions. We hypothesized that instead of eliminating some classifiers completely, it is better to allow each classifier to vote for only those classes for which it is more reliable. We have used ME as the base classifier. The proposed technique is evaluated for a resource poor language, namely Bengali. Evaluation results show that the proposed technique outperforms the best individual classifier, two *baseline* ensembles and the classifier ensemble identified by a single objective based ensemble technique.

Future works include investigating appropriate way of ensembling with the heterogenous classifiers like ME, Conditional Random Field and Support Vector Machine.

References

Anderson, T. W. and S.L. Scolve. 1978. *Introduction to the Statistical Analysis of Data*. Houghton Mifflin.

Darroch, J. and D Ratcliff. 1972. Generalized Iterative Scaling for Log-linear Models. *Ann. Math.Statistics*, 43, 1470–1480.

Deb, Kalyanmoy. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Ltd, England.

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 181–197.

Ekbal, A. and S. Bandyopadhyay. 2008a. Web-based Bengali News Corpus for Lexicon Development and POS Tagging. *POLIBITS, ISSN 1870-9044*, 37, 20–29.

Ekbal, A. and S. Bandyopadhyay. 2008b. A Web-based Bengali News Corpus for Named Entity Recognition. *Language Resources and Evaluation Journal*, 42(2), 173–182.

Ekbal, A. and S. Bandyopadhyay. 2009a. A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi. *Linguistic Issues in Language Technology (LiLT)*, 2(1), 1–44.

Ekbal, A. and S. Bandyopadhyay. 2009b. Voted NER System using Appropriate Unlabeled Data. *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009), ACL-IJCNLP 2009*, pp. 202–210.

Ekbal, Asif, Sriparna Saha, and Christoph S. Garbe. 2010. Named Entity Recognition: A Genetic Algorithm based Classifier Ensemble Selection Approach. In *Proceedings of 2010 International Conference on Artificial Intelligence (ICAI 2010)*, USA.

Pietra, Della, Vincent Stephen, and John Lafferty. 1997. Inducing Features of Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 380–393.

Tjong Kim Sang, Erik F. and Fien De Meulder. 2003. Introduction to the Conll-2003 Shared Task: Language Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147.