ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-783

Classifier Ensemble using Multiobjective Optimization for Named Entity Recognition

Asif Ekbal [1] and **Sriparna Saha** [2,3]

Abstract. In this paper, we report a multiobjective optimization (MOO) based classifier ensemble technique to solve the problem of Named Entity Recognition (NER). Our underlying assumption is that rather than searching for the best feature set for a particular classifier, ensembling of several classifiers which are trained using different feature representations could be a more fruitful approach. But, it is very crucial to select the appropriate classifiers that can participate in final ensembling. Here, we propose a new technique for classifier ensembling based on MOO that can simultaneously optimize several different classification measures. Maximum Entropy (ME) framework is used to generate a number of classifiers by considering the various combinations of the available features. The proposed technique is evaluated for two resource constrained languages, namely Bengali and Hindi. Evaluation results yield the recall, precision and F-measure values of 72.34%, 84.94% and 78.13%, respectively for Bengali, and 64.93%, 83.29% and 72.97%, respectively for Hindi. Experiments also show that the classifier ensemble identified by the proposed multiobjective based approach outperforms all the individual classifiers, two different *baseline* ensembles and a classifier ensemble identified by the single objective genetic algorithm (GA) based approach.

1 Introduction

Named Entity Recognition (NER) is a well-established task that has huge applications in many Natural Language Processing (NLP) areas including Information Retrieval, Information Extraction, Machine Translation, Question Answering and Automatic Summarization etc. The objective of NER is to identify and classify every word/term in a document into some predefined categories like person name, location name, organization name, miscellaneous name (date, time, percentage and monetary expressions etc.) and "none-of-the-above". The main approaches to NER can be grouped into three main categories, namely rule-based, machine learning based and hybrid approach. Rule based approaches focus on extracting names using a number of handcrafted rules that yield better results for restricted domains; and are capable of detecting complex entities that are difficult with learning models. These types of systems are often domain dependent, language specific and do not necessarily adapt well to new domains and languages. Nowadays, researchers are popularly using machine learning approaches for NER because these are easily trainable, adaptable to different domains and languages as well as their maintenance are also less expensive. The main shortcoming of machine learning approach (particularly, supervised systems) is the requirement of large annotated corpus in order to achieve reasonable performance. Thus, building NER systems using machine learning approaches for the resource constrained languages is a great problem. In hybrid systems, the goal is to combine rule-based and machine learning based techniques, and develop new methods using strongest points from each one. Although, hybrid approaches can attain better result than some other approaches, but the weakness of rule-based system still exists when there is a need to change the domain and/or language of data.

In the literature, a lot of works are available that use any of these techniques. But, the languages covered include English, most of the European languages and some of the Asian languages like Chinese, Japanese and Korean. India is a multilingual country with great linguistic and cultural diversities. People speak in 22 different official languages that are derived from almost all the dominant 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 is more difficult and challenging compared to others due to the lack of capitalization information, appearance of NEs in the dictionary as common nouns, relatively free word order nature of the languages, resource-constrained environment, i.e., non-availability of corpus, annotated corpus, name dictionaries, morphological analyzers, part of speech (POS) taggers etc. Some of the works related to Indian language NER can be found in [5, 11, 9, 10] for Bengali and in [14] for Hindi. For Bengali, the available works are based on unsupervised learning [5], supervised learning like Hidden Markov Model (HMM) [11], Conditional Random Field (CRF) [9] and Support Vector Machine (SVM) [6]. Various works of NER involving Indian languages using different approaches are reported in the proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages (NERSSEAL)[4].

1.1 Background of the Present Work

Classifier ensembling is a new direction of machine learning. In the literature there exists some works related to NER that use classifier combination techniques. For example, Florian et al. [12] reported a system by combining four diverse classifiers that exhibited best performance in CoNLL-2003 shared task [15]. In Indian languages, the classifier combination technique for NER has been reported in [10] for Bengali. But, these two works are based on the heterogenous classifiers and made use of more complex set of features, gazetteers, various post-processing techniques as well as the unlabeled data to im-

[1] Department of Computational Linguistics, Heidelberg University, Germany, email: ekbal@cl.uni-heidelberg.de, asif.ekbal@gmail.com
[2] Image Processing and Modeling IWR, Heidelberg University, Germany, email: sriparna.saha@iwr.uni-heidelberg.de, sriparna.saha@gmail.com
[3] Authors equally contributed to this work.

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

prove the performance. In contrast, our proposed algorithm is based on a small set of features that can be very easily obtained for many languages, and does not make use of any additional resources.

The main goal of ensemble is to achieve better generalization accuracy that greatly depends on the diversity of each individual classifier as well as on their individual performance. Thus, it is a very crucial step to determine the appropriate set of classifiers that can participate in classifier ensembling. Some optimization technique like genetic algorithm (GA) [13] may be used to determine these appropriate classifiers. But, these single objective optimization techniques can only optimize a single quality measure, e.g., recall, precision or F-measure at a time. In reality, sometimes a single measure like these cannot capture the quality of a good ensembling reliably. Any good ensemble should have it's recall, precision and F-measure parameters optimized simultaneously. In this paper, we use a multiobjective optimization (MOO) technique [3] in order to simultaneously optimize two different classification measures, namely recall and precision. It is thoroughly discussed in the initial chapters of [3] that weighted sum approach (here, F-measure is a weighted average of recall and precision) cannot identify all non-dominated solutions. Thus, it is indeed effective to solve the classifier ensemble [5] problem using a MOO technique.

MOO problem, typically has a rather different perspective. While in single objective optimization there is only one global optimum, in MOO there is a set of global optimum solutions called Pareto optimal set [3]. All these solutions have equal importance. A single objective approximation of multiple objectives, in form of a weighted sum, unfortunately often fails to capture the full Pareto front. Over the past decade, a number of multiobjective evolutionary algorithms have been suggested [2, 16]. The prime motivation for using evolutionary algorithms (EAs) to solve multiobjective problems is their population-based nature and ability to find multiple optima simultaneously. A simple EA can be easily extended to maintain a diverse set of solutions.

1.2 Overview of the Present Work

In the present work, we develop a MOO based classifier ensemble technique. Maximum Entropy (ME) is used to build a number of classifiers depending on the different combinations of the available features. These features are language independent and applicable for almost all the languages. Thereafter, a recently developed and widely used MOO technique, NSGA-II [4] is used to search for the appropriate combination of classifiers. The proposed approach is evaluated for two resource-poor (or, resource-constrained) languages, namely Bengali and Hindi. In terms of native speakers, Bengali is the *fifth* popular language in the world, *second* in India and the *national* language in Bangladesh. Hindi is the *third* popular language in the world and the *national* language of India. Evaluation results show the effectiveness of the proposed approach with the overall recall, precision, and F-measure values of 72.34%, 84.94% and 78.13%, respectively for Bengali, and 64.93%, 83.29% and 72.97%, respectively for Hindi. Evaluation results also show that the classifier ensemble identified by our proposed multiobjective based approach outperforms all the individual classifiers, two different *baseline* ensembles and the single objective GA based classifier ensemble for both the languages. The main contributions of our work are as follows:

1. MOO is used for selecting appropriate classifiers to form an ensemble. We tried to establish that such ensembling is capable to increase the classification quality by a reasonable margin compared to the conventional ensembling methods.
2. ME is used as a test classifier due to it's less computational overhead. However, the proposed method will work for any set of classifiers, i.e., either homogeneous or heterogeneous. The proposed technique is a very general approach and it's performance may further improve depending upon the choice and/or the number of classifiers as well as the use of more complex features.
3. The proposed technique is language independent that can be replicated for any resource-poor language very easily. Here, we evaluate the proposed algorithm for two resource-constrained languages, namely Bengali and Hindi.
4. The proposed framework is applicable for any type of classification problems like NER, POS-tagging, question-answering etc. To the best of our knowledge, use of MOO to select classifier ensemble is a novel contribution.
5. Note, that our work proposes a novel way of ensembling the available classifiers. Thus, the performance of the existing ensembling works (e.g., [10, 12] etc.) can be further improved with our framework.
6. Another important motivation of MOO based technique is to provide the users a set of alternative solutions. The alternatives could be the solutions with high precision values or solutions with high recall values or solutions with moderate recall and precision values. Depending upon the nature of the problems or the requirement of the users, appropriate solutions can be selected.

2 NE Features for MaxEnt Model

In this work, we use MaxEnt model as a base classifier. The MaxEnt model produces a probability for each class label t (the NE class) of a classification instance, conditioned on its context of occurrence h. This probability is calculated by:

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

where, $f_j(h,t)$ is the j-th feature with associated weight λ_j and $Z(h)$ is a normalization constant to ensure a proper probability distribution. We use the following features for constructing the various classifiers based on this ME framework. These features are language independent in nature and can be very easily derived for many languages.

1. **Context words**: These are the local contexts surrounding the current word. Here, we consider context window of size five, i.e. previous two and next two words. We include this feature as the context words carry useful information for NE identification.
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 languages like Bengali and Hindi. Actually, these are the character sequences stripped from either the rightmost or leftmost positions of the words. For example, the suffixes of length upto 3 characters of the word "*ObAmA*" [Obama] are "*A*", "*mA*" and "*AmA*" whereas, it's prefixes of length up to 3 characters are "*ObAmA*" [Obama] are "*O*", "*Ob*" and "*ObA*".
3. **First word**: This is a binary valued feature that checks whether the current token is the first word of the sentence or not. Though Indian languages are relatively free word order in nature, NEs generally appear in the first position of the sentence, specifically in the newspaper corpus.

[5] We use 'classifier ensemble' and 'ensemble classifier' interchangeably throughout the paper

4. **Length of the word**: This binary valued feature is used to check whether the length of the token is less than a predetermined threshold (here, 3 characters) value and based on the observation that very short words are most probably not the NEs.
5. **Infrequent word**: A cut off frequency is chosen in order to consider the infrequent words in the training corpus with the observation that very frequent words are rarely NEs. In the present work, we set the threshold values to 7 and 10 for Bengali and Hindi, respectively. Then, a binary valued feature is defined that fires for those words, having less occurrences than the cut off frequency.
6. **Part of Speech (POS) information**: We use POS information of the current word as a feature. We have used a SVM based POS tagger [7] that was originally developed with a tagset of 27 tags, defined for the Indian languages. In this particular work, we evaluated this tagger with a coarse-grained tagset of only three tags, namely Nominal, PREP (Postpositions) and Other. The coarse-grained POS tagger has been found to perform better compared to a fine-grained one in case of ME based NER.
7. **Position of the word**: Sometimes, position of the word in a sentence acts as a good indicator for NE identification. In Indian languages, verbs generally appear in the last position of the sentence. We define a binary valued feature that fires if the current word appears in the last position of the sentence.
8. **Digit features**: Several digit features are defined depending upon the presence and/or the number of digits and/or symbols in a token. These features are digitComma (token contains digit and comma), digitPercentage (token contains digit and percentage), digitPeriod (token contains digit and period), digitSlash (token contains digit and slash), digitHyphen (token contains digit and hyphen) and digitFour (token consists of four digits only). These features are helpful to identify miscellaneous NEs.

3 Multiobjective Algorithms

The multiobjective optimization(MOO) can be formally stated as follows [3]. 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.

Nondominated Sorting Genetic Algorithm-II (NSGA-II). Genetic algorithms (GAs) are known to be more effective [3] for solving multiobjective problems primarily because of their population-based nature. NSGA-II [4] 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 relation yields 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 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 [4] 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.

4 Proposed Multiobjective GA for Classifier Ensemble Selection

A multiobjective GA, along the lines of NSGA-II, is now proposed to find an appropriate classifier ensemble for NER. Note, that although the proposed approach has some similarity in steps with NSGA-II, any other existing multiobjective GAs could have been used as the underlying MOO technique.

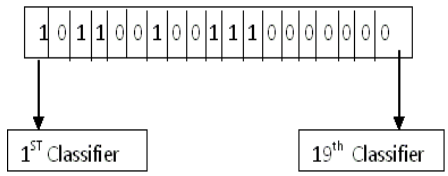


Figure 1. Chromosome Representation

4.1 Chromosome Representation and Population Initialization

If the total number of available classifiers is M, then the length of the chromosome is M. As an example, the encoding of a particular chromosome is represented in Figure 1. 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^{th}classifier does not participate in the classifier ensemble. Else, if it is 1 then the i^{th} classifier participates in the classifier ensemble. In the above example, 7 classifiers, i.e. first, third, fourth, seventh, tenth, eleventh and twelfth take part in constructing the ensemble. If the population size is P then all the P number of chromosomes of this population are initialized in the above way.

4.2 Fitness Computation

Initially, we calculate F-measure values of all the individual ME based classifiers using 3-fold cross validation on the available training data. Then, we calculate the fitness value by executing the following steps.

1. Suppose, there are N number of classifiers present in the ensemble represented in a particular chromosome (i.e., there are total N number of 1's in that chromosome). Let, the overall average F-measure values of the 3-fold cross validation on the training data for these N classifiers be F_i, $i = 1 \ldots N$.
2. Here, the training data is divided into 3 parts. Each classifier is trained using 2/3 of the training data and tested with the remaining 1/3 part. Now, we have N tags (each from a different classifier) for each word in the 1/3 training data. During ensembling, the appropriate output label for each word is determined using the weighted voting of these N classifiers' outputs. The weight of the output class provided by the i^{th} classifier is equal to F_i.
3. The overall recall, precision and F-measure values of this classifier ensemble 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 this classifier ensemble are used as the two objective functions of the proposed MOO technique.

Motivation for using recall and precision as two objective functions. The definitions of recall and precision suggest 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. Figure 2(a) shows, for example, the Pareto optimal front identified by the proposed MOO approach for Bengali NER. This again supports the contradictory nature of these two objective functions.

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.

4.3 Genetic Operators

We use crowded binary tournament selection as in NSGA-II, followed by conventional crossover and mutation for the MOO based classifier ensemble. The most characteristic part of NSGA-II is its elitism operation, where the non-dominated solutions [3] 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 classifier ensemble problem.

4.4 Selection of a Solution from the Final Pareto Optimal Front

In MOO, the algorithms produce a large number of non-dominated solutions [3] on the final Pareto optimal front. Each of these solutions provides a 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 overall average F-measure value of the classifier ensemble is computed from the 3-fold cross validation on the available training data. F-measure is the weighted harmonic mean of precision and recall. The solution with the maximum F-measure value is selected as the desired solution. Final results on the test data are reported using the classifier ensemble corresponding to this best solution. There can be many other different approaches of selecting a solution from the final Pareto optimal front.

5 Experimental Results and Discussions

We use the OpenNLP Java based ME package [6] for the MaxEnt experiments. 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. The parameters are selected after executing a detailed sensitivity study of parameters on the performance of the proposed algorithm. The source-code for NSGA-II is obtained from [7]. We define two different *baseline* classifier ensemble techniques as below:

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. If all the outputs differ then anyone is selected randomly.
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 on the training data.

5.1 Datasets for NER

Indian languages are resource-constrained in nature. For NER, we use a Bengali news corpus [8], developed from the archive of a leading Bengali newspaper available in the web. A portion, containing 250K wordforms, of this corpus has been manually annotated with a coarse-grained NE tagset of four tags namely, PER, LOC, ORG and MISC that denote person, location, organization and miscellaneous names, respectively. The miscellaneous name includes date, time, number, percentages, monetary expressions and measurement expressions. The data has been 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)[8] Shared Task data of around 100K wordforms that were originally tagged with a fine-grained tagset of twelve tags. This data is mostly from the *agriculture* and *scientific* domains. For Hindi, we use the dataset of approximately 502,913 wordforms obtained from the NERSSEAL shared task. An appropriate mapping is defined to convert the fine-grained NE annotated data to the desired forms, i.e., tagged with a coarse-grained tasget of four tags. In order to report the evaluation results, we randomly select a portion of each datset as the test set. Some statistics of the training and test sets are presented in Table 1. The percentages of unseen NEs in the Bengali and Hindi test sets are 35.1 and 40.3, respectively. 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 [15]. Other than NEs are denoted by 'O'.

Table 1. Statistics of the datasets

Language	# words in training	#NEs in training	#words in test	#NEs in test
Bengali	312,947	37,009	37,053	4,413
Hindi	444,231	43,021	58,682	3,005

5.2 Results and Discussions

We build a number of different ME models by considering the various combinations of the available NE features. Being language independent in nature, these features can be derived for almost all the

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

[7] http://www.iitk.ac.in/kangal/codes.shtml

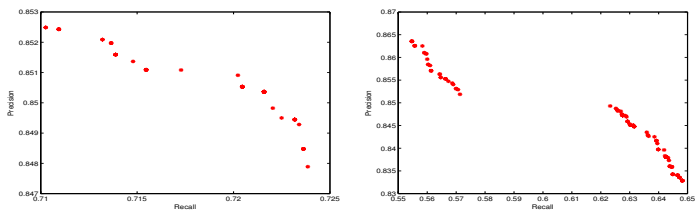
[8] http://ltrc.iiit.ac.in/ner-ssea-08

Table 2. Evaluation results with various feature types for Bengali. Here, the following abbreviations are used: 'CW':Context words, 'PRE-SIZE': Size of the prefix, 'SUF-SIZE': Size of the suffix, 'WL': Word length, 'IW': Infrequent word, 'PW': Position of the word, 'FW':First word, DI: 'Digit-Information', X: Denotes the presence of the corresponding feature (we report percentages)

Classifier	CW	FW	PRE-SIZE	SUF-SIZE	WL	IW	PW	DI	POS	recall	precision	F-measure
M_1	X	X						X	X	35.59	62.74	45.42
M_2	X	X	3					X	X	63.12	78.61	70.02
M_3	X	X	3	3				X	X	68.81	81.34	74.55
M_4	X	X	3	3	X			X	X	68.65	81.57	74.55
M_5	X	X	3	3	X	X		X	X	69.35	81.37	74.88
M_6	X	X	3	3	X	X	X	X	X	69.15	81.53	74.83
M_7	X	X	4					X	X	65.45	79.43	71.76
M_8	X	X	4	3				X	X	68.42	81.58	74.42
M_9	X	X	3	4				X	X	69.39	81.66	75.03
M_{10}	X	X	4	4				X	X	68.65	81.13	74.37
M_{11}	X	X	4	3	X			X	X	67.81	81.53	74.04
M_{12}	X	X	3	4	X			X	X	69.39	82.02	75.18
M_{13}	X	X	4	4	X			X	X	68.01	81.00	73.94
M_{14}	X	X	4	3	X	X		X	X	68.69	81.46	74.53
M_{15}	X	X	3	4	X	X		X	X	**69.76**	**81.75**	**75.28**
M_{16}	X	X	4	4	X	X		X	X	68.87	80.89	74.40
M_{17}	X	X	4	3	X	X	X	X	X	68.58	81.64	74.54
M_{18}	X	X	3	4	X	X	X	X	X	69.67	81.85	75.27
M_{19}	X	X	4	4	X	X	X	X	X	68.51	81.01	74.24

Table 3. Overall results for Bengali (we report percentages)

Classification Scheme	recall	precision	F-measure
Best individual classifier	69.76	81.75	75.28
Baseline 1	69.83	82.90	75.81
Baseline 2	70.25	82.97	76.08
GA based ensemble	71.14	84.07	77.11
MOO based ensemble	**72.34**	**84.94**	**78.13**



(a) (b)

Figure 2. Pareto optimal front for (a) Bengali (b) Hindi

languages. In this particular work, we construct the classifiers from the following set of features:
context of size five (previous two words, current word and next two 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, and several digit features.

We construct 19 different classifiers as shown in Table 2 for Bengali. The best individual classifier shows the recall, precision and F-measure values of 69.76%, 81.75% and 75.28%, respectively. Overall performance of the best individual classifier, two different *baseline* ensembles, the classifier ensemble identified by a single objective optimization based technique and the classifier ensemble identified by our proposed MOO based technique are presented in Table 3. In single objective GA based ensemble technique, we optimize only F-measure which is a combination of both recall and precision. Thus, optimization of F-measure may not always lead to optimization of both recall and precision together. Results show that the proposed MOO based classifier ensemble outperforms all the other models. We observe the improvement in recall, precision and F-measure values by 2.58%, 3.19% and 2.85%, respectively over the best individual classifier, 2.51%, 2.04% and 2.32% over *Baseline 1* and 2.09%, 1.97% and 2.05% over *Baseline 2*. The proposed technique also performs superior over the single objective GA based ensemble with more than 1.02% F-measure. The best solution of the proposed MOO based classifier selection approach selects the following classifiers for ensembling: M_2, M_3, M_6, M_{10}, M_{12}, M_{14}, M_{15}, M_{17} and M_{19}.

Statistical analysis of variance, (ANOVA) [1], is performed in order to examine whether the MOO based ensemble technique really outperforms the best individual classifier, two *baseline* ensembles and GA based ensemble. 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 the cases. Figure 2(a) shows the final Pareto optimal front identified by the proposed technique for Bengali. This figure shows that there indeed exists a large number of alternative solutions with different recall and precision values.

Thereafter, the proposed system is evaluated with Hindi data. Evaluation results with various classifier combinations are reported in Table 4. Each of the classifiers is trained with the same set of features as Bengali. Experimental results are reported in Table 5. It shows the overall recall, precision and F-measure values for the proposed technique as 64.93%, 83.29% and 72.97%, respectively. This is the improvement of 3.15%, 2.02%, 1.40% and 0.90% F-measure values over the best performing individual classifier, *Baseline 1*, *Baseline 2* and the single objective GA based ensemble technique, respectively. ANOVA tests for Hindi again show that the differences in mean recall, precision and F-measure values of the proposed technique with respect to the individual classifier and three ensemble methods are statistically significant (p value is less than 0.05 in each case). The best solution of the proposed MOO based classifier ensemble technique selects the following classifiers for ensembling: M_7, M_8, M_9, M_{10}, M_{11}, M_{14}, M_{16} and M_{19}. Figure 2(b) represents the final Pareto optimal front as identified by the proposed technique for Hindi. Similar to Bengali, the proposed MOO based technique provides the users with a number of alternative solutions having different recall and precision values. Depending upon the particular requirement, user can choose one or more solution(s) from these.

Comparison between the results reveals that the proposed approach performs better for Bengali in comparison to Hindi. The possible reasons may be (i) higher unbalanced class distribution (i.e., ratio between non-NEs and NEs) in Hindi training data (9.33:1) than Bengali (7.46:1) and (ii). presence of more unknown NEs in the Hindi test data than Bengali (see Table 1). Error analysis shows that most of the errors in our proposed algorithm are concerned with the confusions: O vs. I-ORG, I-LOC vs. O, I-PER vs. O etc, i.e., the system suffers from the low recall values for both the languages. The system performs best for the miscellaneous NEs followed by person, location and organization classes.

Table 4. Evaluation results with various feature types for Hindi. Here, the abbreviations are same as Bengali (we report percentages)

Classifier	CW	FW	PRE-SIZE	SUF-SIZE	WL	IW	PW	POS	DI	recall	precision	F-measure
M_1	X	X						X	X	29.36	69.30	41.25
M_2	X	X	3					X	X	58.79	79.68	67.66
M_3	X	X	3	3				X	X	62.09	79.57	69.75
M_4	X	X	3	3	X			X	X	61.85	79.63	69.63
M_5	X	X	3	3	X	X		X	X	**62.16**	79.52	69.78
M_6	X	X	3	3	X	X	X	X	X	62.12	79.68	**69.82**
M_7	X	X	4					X	X	50.54	78.83	61.59
M_8	X	X	4	3				X	X	54.08	**80.04**	64.55
M_9	X	X	3	4				X	X	60.10	79.05	68.29
M_{10}	X	X	4	4				X	X	52.33	79.18	63.01
M_{11}	X	X	4	3	X			X	X	59.73	79.04	68.04
M_{12}	X	X	3	4	X			X	X	53.57	79.75	64.09
M_{13}	X	X	4	4	X			X	X	51.62	79.65	62.64
M_{14}	X	X	4	3	X	X		X	X	59.97	78.70	68.07
M_{15}	X	X	3	4	X	X		X	X	54.18	79.61	64.48
M_{16}	X	X	4	4	X	X		X	X	52.49	79.36	63.19
M_{17}	X	X	4	3	X	X	X	X	X	59.94	78.97	68.15
M_{18}	X	X	3	4	X	X	X	X	X	54.01	79.83	64.43
M_{19}	X	X	4	4	X	X	X	X	X	52.29	79.73	63.16

Table 5. Overall results for Hindi (we report percentages)

Classification Scheme	recall	precision	F-measure
Best individual classifier	62.12	79.68	69.82
Baseline 1	63.57	80.28	70.95
Baseline 2	64.12	80.97	71.57
GA based ensemble	64.85	81.10	72.07
MOO based ensemble	**64.93**	**83.29**	**72.97**

Comparisons to Other Works. It will not be fair to compare the performance of our proposed system with that of the previous proposals for Bengali [9, 6, 10] and Hindi [14] as these works use (i). different experimental set up, (ii). different data sets, (iii). more complex set of features and (iv). domain dependent knowledge and/or resources. In contrast, our proposed algorithm is based on a relatively small set of features that can be easily obtained for almost all the languages, does not make use of any domain dependent information and hence can be replicated for any resource-poor language very easily. Though we use the IJCNLP-08 NERSSEAL shared task data, we convert these fine-grained annotated data to the coarse-grained forms. Thus, comparing our proposed system with that of the shared task papers [9] is also out-of-scope.

6 Conclusion and Future Works

In this paper, we have proposed a MOO based classifier ensemble technique for NER by simultaneously optimizing two different classification measures, namely recall and precision. We have assumed and shown experimentally that instead of searching for the best-fitting feature set heuristically, it could be more effective to find out an appropriate ensemble technique to combine the different classifiers, where each one is based on distinct feature representation. We built a number of different classifiers by considering the various combinations of the available features using ME framework. One most interesting and important characteristic of our system is that it makes use of only language independent features that can be easily derived for almost all the languages. The proposed technique is evaluated with two resource poor languages, namely Bengali and Hindi. Experiments show that the overall performance attained by our proposed algorithm outperforms all the individual classifiers, two different *baseline* ensembles and a single objective GA based ensemble technique.

In future, we would like to construct more classifiers by incorporating some important language independent features such as dynamic NE information etc. as well as by considering more variations of the existing features. We would also extract language dependent features from our various existing in-house resources and tools. In this work, we have only considered ME as the underlying classification technique. Future work also includes the development of classifier ensemble using other well known statistical classifiers, namely CRF and SVM.

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

REFERENCES

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

[2] Carlos A Coello Coello, 'A Comprehensive Survey of Evolutionary-based Multiobjective Optimization Techniques', *Knowledge and Information Systems*, **1**(3), 269–308, (August 1999).

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

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, 'A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, **6**(2), 181–197, (2002).

[5] A. Ekbal and S. Bandyopadhyay, 'Lexical Pattern Learning from Corpus Data for Named Entity Recognition', in *Proceedings of the 5th International Conference on Natural Language Processing (ICON)*, pp. 123–128, India, (2007).

[6] A. Ekbal and S. Bandyopadhyay, 'Bengali Named Entity Recognition using Support Vector Machine', in *Proceedings of Workshop on NER for South and South East Asian Languages, 3rd International Joint Conference on Natural Languge Processing (IJCNLP)*, pp. 51–58, India, (2008).

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

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

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

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

[11] A. Ekbal, S.K. Naskar, and S. Bandyopadhyay, 'Named Entity Recognition and Transliteration in Bengali', *Named Entities: Recognition, Classification and Use, Special Issue of Lingvisticae Investigationes Journal*, **30**(1), 95–114, (2007).

[12] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang, 'Named Entity Recognition through Classifier Combination', in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, (2003).

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.

[14] Wei Li and Andrew McCallum, 'Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction', *ACM Transactions on Asian Languages Information Processing*, **2**(3), 290–294, (2004).

[15] Tjong Kim Sang, Erik F., and Fien De Meulder, '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, (2003).

[16] D. Van Veldhuizen and G. Lamont, 'Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-art', *Evolutionary Computations*, **2**, 125–1473, (2000).