Maximum Entropy Classifier Ensembling using Genetic Algorithm for NER in Bengali

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

[1]Department of Computational Linguistics
University of Heidelberg
Heidelberg-69120, Germany
ekbal@cl.uni-heidelberg.de, asif.ekbal@gmail.com
[2]Interdisciplinary Center for Scientific Computing (IWR)
University of Heidelberg
Heidelberg-69120, Germany
sriparna.saha@iwr.uni-heidelberg.de, sriparna.saha@gmail.com

Abstract

In this paper, we propose classifier ensemble selection for Named Entity Recognition (NER) as a single objective optimization problem. Thereafter, we develop a method based on genetic algorithm (GA) to solve this problem. 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. Maximum Entropy (ME) framework is used to generate a number of classifiers by considering the various combinations of the available features. In the proposed approach, classifiers are encoded in the chromosomes. A single measure of classification quality, namely F-measure is used as the objective function. Evaluation results on a resource constrained language like Bengali yield the recall, precision and F-measure values of 71.14%, 84.07% and 77.11%, respectively. Experiments also show that the classifier ensemble identified by the proposed GA based approach attains higher performance than all the individual classifiers and two different conventional *baseline* ensembles.

1. Introduction

Named Entity Recognition (NER) has immense applications in almost all Natural Language Processing (NLP) application areas that include Information Retrieval, Information Extraction, Machine Translation, Question Answering and Automatic Summarization etc. The main goal of NER is to identify every word/term in a document and to classify them 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 existing approaches of 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. Generally, these systems consist of a set of patterns using grammatical (e.g., part of speech), syntactic (e.g., word precedence) and orthographic features (e.g., capitalization) in combination with dictionaries. Some of the typical systems are University Of Sheffield's LaSIE-II (Humphreys et al., 1998), ISOQuest's NetOwl (Aone et al., 1998) and University Of Edinburgh's LTG ((Mikheev et al., 1998), (Mikheev et al., 1999)) for English NER. These kinds of systems yield results for restricted domains and are capable of detecting complex entities that are difficult with machine learning models. However, rule based systems lack the ability of portability and robustness, and furthermore the high cost of the maintenance of rules increases even when the data is slightly changed.

In comparison, machine learning (ML) approaches have gained more attention to the researchers for NER because these are easily trainable, adaptable to different domains and languages as well as their maintenance are also less expensive. The ML techniques can be grouped into the following three categories, namely supervised ML technique, semi-supervised ML technique and unsupervised ML technique. The idea of supervised learning is to study the features of positive and negative examples of NE over a large collection of annotated documents and design rules that capture instances of a given type. The popularly used supervised ML approaches used in NER are Hidden Markov Model (HMM) ((Miller et al., 1998), (Bikel et al., 1999)), Maximum Entropy (ME)(Borthwick, 1999), Decision Tree (Sekine, 1998) and Conditional Random Field (CRF) (Lafferty et al., 2001). The main shortcoming of supervised learning is the requirement of a large annotated corpus in order to obtain the reasonable performance. But, this is often a great problem for working with the resource poor languages. The creation of large amount of annotated data is both cost sensitive and time consuming. The unavailability of such resources and the prohibitive cost of creating them lead to two alternative learning methods: semi-supervised learning and unsupervised learning. The term "semi-supervised" (or, "weakly supervised") is relatively recent and more useful, specifically for the resource poor languages. One commonly used technique for semi-supervised approach is "bootstrapping" (Riloff and Jones, 1999) that involves a small degree of supervision, such as a set of seeds, for starting the learning process. Clustering is a typical approach in unsupervised learning. For example, one can try to gather NEs from clustered groups based on the similarity of context. In hybrid systems (Srihari et al., 2002), the goal is to combine rule-based and ML-based methods, and develop new methods using strongest points from each method. Although, hybrid approaches can get better result than some other approaches, but weakness of

handcraft rule based NER surfaces when there is a need to change the domain of data.

Besides these, there are many other existing works in the area of NER. 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 at least 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 Bengali as well as in any Indian language is more difficult and challenging compared to others due to the following facts:

- Lack of capitalization information that acts as a good indicator for NE identification, especially in English.
- Indian names are more diverse and a lot of these appear in the dictionary as common nouns.
- Indian languages are relatively free word order in nature.
- Bengali, like any other Indian languages, is also resource constrained, i.e., corpus, annotated corpus, name dictionaries, morphological analyzers, part of speech (POS) taggers etc are not readily available.
- Indian languages are highly inflected and provide rich and challenging sets of linguistic and statistical features resulting in long and complex wordforms.

For Bengali, a few works are available that are based on unsupervised lexical pattern learning from the unlabeled data (Ekbal and Bandyopadhyay, 2007), HMM (Ekbal et al., 2007) that considers additional context information during emission probabilities, CRF (Ekbal and Bandyopadhyay, 2009a), SVM (Ekbal and Bandyopadhyay, 2008a) and voting (Ekbal and Bandyopadhyay, 2009b). Various systems on NER in Indian languages using different approaches can be found in the proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages (NERSSEAL)[1].

The performance of any classification technique depends on the features of training and test data sets. Feature selection, also known as variable selection, feature reduction, attribute selection or variable subset selection, is the technique, commonly used in machine learning, of selecting a subset of relevant features for building robust learning models. In ME based models, selection of appropriate features is a crucial problem and also a key issue to improve the recognition as well as classification performance. It does not provide a method for automatic feature selection and uses heuristics for this task in general. Rather than selecting the best-fitting feature set, ensembling several NER systems where each one is based on different feature representation can be considered as an alternative research direction. Ensembling of classifiers is done to increase the generalization accuracy that greatly depends on the diversity of each individual classifier as well as on their individual performance. But, it is a very crucial step to determine the particular subset of classifiers (from a set) that can participate in the process of an ensemble construction. In this paper, we formulate this classifier ensemble selection problem under the single objective optimization framework that uses genetic algorithm (GA) (Goldberg, 1989). Genetic algorithms (Goldberg, 1989) are randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. It performs search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. In GAs, the parameters of the search space are encoded in the form of strings called *chromosomes*. A collection of such strings is called a *population*. Initially, a random population is created, which represents different points in the search space. An *objective* or a *fitness* function are associated with each string that represents the degree of *goodness* of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these strings to yield a new generation of strings. The processes of selection, crossover and mutation continue for a fixed number of generations or till a termination condition is satisfied.

Depending on the various available feature combinations, different versions of the ME based classifier are made. One most interesting and important characteristics of these features is that these are language independent in nature, and can be easily derived for almost all the languages with a very little effort. Here, classifiers are encoded in the chromosomes. The average F-measure value of the 3-fold cross validation (on the training data) of the classifier ensemble encoded in a particular chromosome is used as it's fitness value. We use elitism to keep the best solution intact in a place outside the population. The proposed approach is evaluated for a resource-constrained language, namely Bengali. In terms of native speakers, Bengali ranks fifth in the world and second in India. Bengali is also the national language in Bangladesh. Evaluation results show the effectiveness of the proposed approach with the overall recall, precision and F-measure values of 71.14%, 84.07% and 77.11%, respectively. We also show that our technique performs superior to the two conventional *baseline* ensembles.

The remainder of the paper is organized as follows. Section 2 gives a very brief introduction about the main goal of this work. The ME framework for NER is introduced in Section 3. In Section 4, we describe the various language independent features used for our NER task. We elaborately describe our proposed GA based classifier ensemble technique in Section 5. Detailed evaluation results along with the necessary discussions are reported in Section 6. Finally, Section 7 concludes the paper with some future directions.

2. Goal of the Paper

The goal of the paper is to develop a single objective GA based classifier ensemble technique for NER in Bengali. The classifier ensemble problem is stated as follows.

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

Suppose, there are N number of classifiers available and these be denoted by $C_1, \ldots, C_N$. Let, $\mathcal{A} = \{C_i : i = 1; N\}$. The classifier ensemble selection problem is then stated as follows: Find a set of classifiers B which will optimize a function $F(B)$ such that: $B \subseteq A$. Here, F is a classification quality measure of the combined classifier. The particular kind of problem like NER has three different types of classification quality measures, namely recall, precision and F-measure. Thus, $F \in \{$ recall, precision, F-measure$\}$. Combination of the classifiers can be done by either majority voting or weighted voting. Several optimization techniques exist in the literature. In this paper we use GA (Goldberg, 1989), a very popular search technique, to solve the above mentioned classifier ensemble selection problem.

3. Maximum Entropy Framework for NER

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

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

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:

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

where, h_i is the context for the word w_i.

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

$$f_j(h,t) = 1 \text{ if word}(h) = \textit{sachIn} \text{ and } t = \text{I-PER} \quad (3)$$
$$= 0 \quad \text{otherwise} \quad (4)$$

We use the OpenNLP Java based ME package [2] 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. In the present work, we use the generalized iterative scaling (Darroch and Ratcliff, 1972) algorithm to estimate the MaxEnt parameters.

4. Named Entity Features

The main features for the NER task are identified based on the different possible combinations of available word and tag contexts. We use the following features for constructing the various classifiers based on the ME framework. These features are language independent in nature, and can be easily derived for almost all the languages.

1. Context words: These are the preceding and succeeding words of the current word. This feature is added with the observation that surrounding words carry effective information for the identification of NEs.

2. Word suffix: Fixed length (say, n) word suffixes are very effective to identify NEs and work well for the highly inflected language like Bengali. Actually, these are the fixed length character strings stripped from the rightmost position of the words. For example, the suffixes of length up to 3 characters of the word "*ObAmA*" [Obama] are "*A*", "*mA*" and "*AmA*". If the length of the corresponding word is less than or equal to $n-1$ then the feature values are "not defined" (denoted by ND). The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. This feature is included with the observation that NEs share some common suffixes.

3. Word prefix: Fixed length word prefixes are used as the features. These are the fixed length character strings stripped from the leftmost positions of the words. For example, the prefixes of length up to 3 characters of the word "*ObAmA*" [Obama] are "*O*", "*Ob*" and "*ObA*". This is also defined in the similar way as like the word suffixes.

4. Infrequent word: This is a binary valued feature that checks whether the current word appears in training set very frequently or not. We compile a list of most frequently occurring words from the training set by defining an appropriate threshold value. In the present work, we set this threshold value to 10. However, this threshold value does vary depending upon the size of the training set. A binary valued feature "INFRQ" is set to 1 if the word does not appear in this list, otherwise it is set to 0.

5. Part of Speech (POS) information: POS information of the current and/or the surrounding word(s) are effective for NE identification. We use a SVM based POS tagger (Ekbal and Bandyopadhyay, 2008b) that was originally developed with a tagset of 26 tags, defined for the Indian languages. In this particular work, we evaluate the SVM based POS tagger with a coarse-grained tagset that contains only three tags, namely Nominal, PREP (Postpositions) and Other. We consider postposition as it often occurs after the NEs. The coarse-grained POS tagger has been found to perform better compared to a fine-grained one in case of ME based NER.

6. Position of the word: This binary valued feature checks the position of the word in the sentence. We use this feature as the verbs generally appear in the last position of the sentence in Bengali.

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

Begin
1. $t = 0$
2. initialize population $P(t)$ /* $Popsize = |P|$ */
3. for $i = 1$ to $Popsize$
   compute fitness $P(t)$
4. $t = t + 1$
5. if termination criterion achieved go to step 10
6. select (P)
7. crossover (P)
8. mutate (P)
9. go to step 3
10. output best chromosome and stop

End

Figure 1: Basic Steps of GA

7. 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.

5. Proposed Approach

The proposed GA based classifier ensemble selection method is described below. The basic steps of our approach closely follow those of the conventional GA as shown in Figure 1.

5.1. String 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 2. Here, $M = 19$, i.e., total 19 different classifiers are built. The chromosome represents an ensemble of 7 classifiers (i.e., first, third, fourth, seventh, tenth, eleventh and twelfth classifiers). 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. If the population size is P then all the P number of chromosomes of this population are initialized in the above way.

5.2. Fitness Computation

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

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 $F_i, i = 1 \ldots N$.

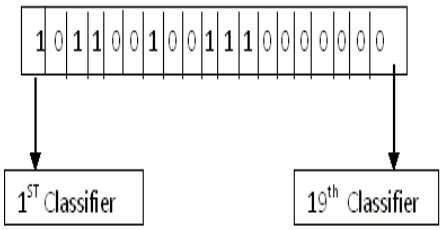

Figure 2: Chromosome Representation

2. Here, the training data is again divided into 3 parts. Each classifier is trained using 2/3 of the training data and tested with the remaining 1/3 part. Now for the ensemble classifier, the output NE tag for each word in the 1/3 training data is determined using the weighted voting of these N classifiers' outputs. The weight of the NE tag provided by the i^{th} classifier is equal to F_i.

3. The overall F-measure value of this ensemble classifier for the 1/3 training data is calculated.

4. Step 2 and 3 are repeated 3 times to perform 3-fold cross validation. The average F-measure value, obtained from the cross validation, of the ensemble classifier is used as the fitness value of that particular chromosome.

The objective is to maximize this fitness value (i.e., F-measure) using the search capability of GA.

5.3. Selection

During each successive generation, a proportion of the existing population is selected to create a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions.
In this paper, we use Roulett wheel selection. Here, the fitness function invoked with each chromosome is used to associate a probability of selection with each individual chromosome. If f_i is the fitness of individual i in the population, its probability of being selected is

$$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j},$$

where N is the number of individuals in the population.
This selection process has resemblance to a Roulette wheel in a casino. Usually, a proportion of the wheel is assigned to each of the possible selections based on their fitness values. This could be achieved by dividing the fitness of a selection by the total fitness of all the selections, thereby normalizing them to 1. Then, a random selection is made similar to how the roulette wheel is rotated. Thus in case of Roulett wheel

selection, chromosomes with a higher fitness are less likely to be eliminated but there is still a chance that they may be.

5.4. Crossover

Here, we use the normal single point crossover (Holland, 1975). Suppose, there are 19 classifiers. The two chromosomes look like:
$P_1 = 0\,1\,1\,0\,0\,0\,1\,1$
$P_2 = 1\,1\,1\,0\,0\,0\,1\,0$

Now, if we consider the crossover point is at 4 then after single point crossover the new chromosomes will look like:
$O_1 = 0\,1\,1\,0\,0\,0\,1\,0$
$O_2 = 1\,1\,1\,0\,0\,0\,1\,1.$
Crossover probability is selected adaptively as in (Srinivas and Patnaik, 1994). The expressions for crossover probabilities are computed as follows. Let, f_{max} be the maximum fitness value of the current population, $\overline{f}$ be the average fitness value of the population and $f^{'}$ be the larger of the fitness values of the solutions to be crossed. Then the probability of crossover, μ_c, is calculated as:

$$\mu_c = k_1 \times \frac{(f_{max} - f^{'})}{(f_{max} - \overline{f})}, \quad \text{if} \quad f^{'} > \overline{f}, \tag{5}$$

$$= k_3, \quad \text{if} \quad f^{'} \leq \overline{f}. \tag{6}$$

Here, as in (Srinivas and Patnaik, 1994), the values of k_1 and k_3 are kept equal to 1.0. Note that, when f_{max}=$\overline{f}$, then $f^{'} = f_{max}$ and μ_c will be equal to k_3. The aim behind this adaptation is to achieve a trade-off between exploration and exploitation in a different manner. The value of μ_c is increased when the better of the two chromosomes to be crossed is itself quite poor. In contrast when it is a good solution, μ_c is low so as to reduce the likelihood of disrupting a good solution by crossover.

5.5. Mutation

Each chromosome undergoes mutation with a probability μ_m. The mutation probability is also selected adaptively for each chromosome as in (Srinivas and Patnaik, 1994). The expression for mutation probability, μ_m, is given below:

$$\mu_m = k_2 \times \frac{(f_{max} - f)}{(f_{max} - \overline{f})} \quad \text{if} \quad f > \overline{f}, \tag{7}$$

$$= k_4 \quad \text{if} \quad f \leq \overline{f}. \tag{8}$$

Here, values of k_2 and k_4 are kept equal to 0.5. This adaptive mutation helps GA to come out of local optimum. When GA converges to a local optimum, i.e., when $f_{max} - \overline{f}$ decreases, μ_c and μ_m both will be increased. As a result GA will come out of local optimum. It will also happen for the global optimum that may result in disruption of the near-optimal solutions. As a result GA will never converge to the global optimum. The μ_c and μ_m will get lower values for high fitness solutions and higher values for low fitness solutions. While the high fitness solutions aid in the convergence of GA, the low fitness solutions prevent the GA from getting stuck at a local optimum. The use of elitism will also keep the best solution intact. The values of μ_c and μ_m are both set to 0 for the solution with the maximum fitness value. The best solution in a population is transferred undisrupted into the next generation. Together with the selection mechanism, this may lead to an exponential growth of the solution in the population and may cause premature convergence. To overcome the above stated problem, a default mutation rate (of 0.02) is kept for every solution in the population. Here, we apply mutation operator to each entry of the chromosome where the entry is randomly replaced by either 0 or 1.

5.6. Termination Condition

In this approach, the processes of fitness computation, selection, crossover, and mutation are executed for a maximum number of generations. The best string seen upto the last generation provides the solution to the above classifier ensemble problem. Elitism is implemented at each generation by preserving the best string seen upto that generation in a location outside the population. Thus on termination, this location contains the best classifier ensemble.

6. Experimental Results and Discussions

We set the following parameter values for GA: population size=100, number of generations=50, probabilities of mutation and crossover are selected adaptively. The system is evaluated in terms of recall, precision and F-measure as defined in CoNLL-2003 shared task (Sang et al., 2003). We define two different *baseline* ensemble systems as below:

- *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.

- *Baseline 2*: All the individual classifiers are combined with the help of a weighted voting approach. In each classifier, weight is calculated based on the average F-measure value of the 3-fold cross validation on the training data. The final output label is selected based on the highest weighted vote.

6.1. Datasets for NER

Indian languages are resource-constrained in nature. For NER, we use a Bengali news corpus (Ekbal and Bandyopadhyay, 2008c), developed from the archive of a leading Bengali newspaper available in the web. A portion of this corpus containing approximately 250K wordforms has been 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 expressions and measurement expressions. We collet the data mostly from the national, states, sports and politics domains of the 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)[3] shared task data of around 100K wordforms that were originally tagged with a fine-grained tagset of twelve tags. This

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

Table 1: Feature types and parameters used for training different ME based classifiers 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

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

data is mostly from the agriculture and scientific domains. An appropriate conversion routine is defined to convert this fine-grained NE annotated data to the desired forms, i.e., tagged with a coarse-grained tagset of four tags. In order to report the evaluation results, we select approximately 37K wordforms from the total 350K wordforms as the test set. The rest is used as the training set. Some statistics of the training and test sets are presented in Table 2. There are 35.1% unseen NEs in the test set.

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. For example, the name *mahatmA gAndhi*[Mahatma Gandhi] is tagged as *mahatmA*[Mahatma]/I-PER *gAndhi*[Gandhi]/I-PER. But, the names *mahatmA gAndhi*[Mahatma Gandhi] *rabIndrAnAth thAkur* [Rabindranath Tagore] are to be tagged as *mahatmA*[Mahatma]/I-PER *gAndhi*[Gandhi]/I-PER *rabIndrAnAth*[Rabindranath]/B-PER *thAkur*[Tagore]/I-PER if they appear sequentially in the text. This is the standard IOB format that was followed in the CoNLL-2003 shared task (Sang et al., 2003).

6.2. Results and Discussions

We build a number of different ME models from the available NE features. We consider various combinations from the following set of features:

context of size five (previous two 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 1 with the various combinations of the available features. The best individual classifier shows the recall, precision and F-measure values of 69.76%, 81.75% and 75.28%, respectively. The corresponding features are: previous two and next two words, first word, prefixes of length upto three characters of only the current word, suffixes of length upto four characters of only the current word, word length, infrequent word, POS information of the current word and the various digit features. Overall evaluation results are presented in Table 3. It reports the overall performance of the best individual classifier, two different *baseline* ensembles and the best ensemble classifier identified by the proposed single objective GA based technique. Results show that the overall performance attained by the classifier ensemble determined by the proposed algorithm outperforms all the other models. It shows the improvement in recall, precision and F-measure values by 1.38%, 2.32% and 1.83%, respectively, over the best individual classifier. In comparison to the first *baseline*, the proposed algorithm performs superior with more than 1.31%, 1.17% and 1.30% in recall, precision and F-measure values, respectively. We also observe the improvement of 0.89% recall, 1.10% precision and 1.03% F-measure over the second *baseline*. The best solution of the proposed GA based classifier selection approach selects the following classifiers for ensembling: M_2, M_3, M_4, M_5, M_7, M_9, M_{10}, M_{11}, M_{12}, M_{14}, M_{16}, M_{18} and M_{19}.

Just for an illustration, we show the boxplot of the F-measure values of the solutions on the final population of the proposed GA based ensemble in Figure 3. The variations of the best F-measure values over generations are shown in Figure 4. This figure shows that the proposed algorithm converges within 21 generations for this particular data set.

Statistical analysis of variance, (ANOVA) (Anderson and Scolve, 1978), is performed in order to examine whether the GA based ensemble technique really outperforms the best individual classifier and two *baseline* ensembles. 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. ANOVA results are reported in details in Table 4. We present the confusion matrix in Table 5 that gives an indication about the merits and demerits of our proposed technique. Table shows that the most of the errors are concerned with O vs. I-ORG, I-PER

Table 2: Statistics of training and test sets

Set	PER	LOC	ORG	MISC
Training	6,717	5,591	3,070	8,058
Test	648	670	374	1,008

Table 3: Overall results for Bengali

Classification Scheme	recall (in %)	precision (in %)	F-measure (in %)
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

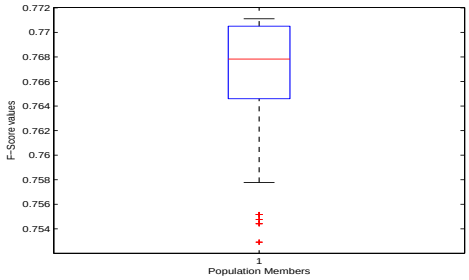

Figure 3: Boxplot of the F-measure values of the solutions on the final population of the proposed GA based technique

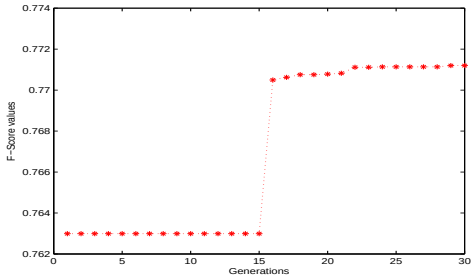

Figure 4: Variations of the best F-measure values over generations

Table 4: Results of pairwise comparisons of different techniques on F-measure values obtained by ANOVA test

Technique Name (I)	Comparing Algo.(J)	Mean Difference (I-J)	Significance Value
GA	Best Classifier	1.38 ± 0.23	0.00
	Baseline 1	1.31 ± 0.18	0.017
	Baseline 2	0.89 ± 0.26	0.023

vs. O, I-LOC vs. O etc. This table also shows that the proposed algorithm performs best for the MISC class followed by PER, LOC and ORG classes.

7. Conclusion and Future Works

In this paper, we propose the use of GA to develop a classifier ensemble for NER. We carried out sufficient experiments to validate our underlying assumption 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 have used ME framework as the base classifier. 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 without any knowledge of them *a priori*. We evaluated our proposed technique for a resource poor language like Bengali. Results show the recall, precision and F-measure values of 71.14%, 84.07% and 77.11%, respectively. Experiments also show the superiority of our proposed technique over the two conventional *baseline* ensembles.

In future we would like to incorporate some more language independent (dynamic NE information etc.) as well as the language specific features to generate more classifiers. In this work, we have considered only ME as the underlying classification technique. Future works include the development of vote based classifier ensembles using some other well-known classifiers like CRF and Support Vector Machine. A 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 can not capture the quality of a good ensembling reliably. Any good ensemble should have it's recall, precision and F-measure parameters optimized simultaneously. Inspired by this, we would like to model the classifier ensemble selection problem under the multiobjective optimization (MOO) framework (Deb, 2001) that can simultaneously optimize more than one classification parameters.

8. Acknowledgement

We gratefully acknowledge Erasmus Mundus Mobility with Asia (EMMA) program of the European Union for their support to carry out our postdoctoral research activ-

Table 5: Confusion matrix for Bengali

O/P tags	0	I-MISC	I-LOC	B-MISC	I-PER	I-ORG	B-PER	B-LOC	B-ORG
O	26615	284	94	3	176	221	0	0	0
I-MISC	52	876	15	1	9	10	0	0	0
I-LOC	116	12	509	0	11	30	0	0	0
B-MISC	12	8	0	35	0	1	0	0	0
I-PER	80	7	4	0	861	1	0	0	0
I-ORG	140	6	8	0	3	674	0	0	0
B-PER	2	0	1	0 0	0	6	0		0
B-LOC	16	1	3	0	0	0	0	21	0
B-ORG	25	0	0	0	0	1	0	0	54

ities in the University of Heidelberg, Germany.

9. References

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

Chinatsu Aone, L. Halverson, T. Hampton, and M. Ramos-Santacruz. 1998. SRA: Description of the IE2 system used for MUC-7. In *MUC-7*. Fairfax, Virginia.

Daniel M. Bikel, Richard L. Schwartz, and Ralph M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, 34(1-3):211–231.

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

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

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

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

A. Ekbal and S. Bandyopadhyay. 2008a. 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)*, pages 51–58, India.

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

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

A. Ekbal 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.

A. Ekbal 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*, pages 202–210.

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

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

J. H. Holland. 1975. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, AnnArbor.

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunnigham, and Y. Wilks. 1998. Univ. Of Sheffield: Description of the LaSIE-II System as Used for MUC-7. In *MUC-7*. Fairfax, Virginia.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pages 282–289.

A. Mikheev, C. Grover, and M. Moens. 1998. Description of the LTG System used for MUC-7. In *MUC-7*. Fairfax, Virginia.

A. Mikheev, C. Grover, and M. Moens. 1999. Named Entity Recognition without Gazeteers. In *Proceedings of EACL*, pages 1–8. Bergen, Norway.

S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schawartz, R. Stone, R. Weischedel, and the Annotation Group. 1998. BBN: Description of the SIFT System as Used for MUC-7. In *MUC-7*, Fairfax, Virginia.

E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings AAAI '99/IAAI '99: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Conference on Innovative Applications of Artificial Intelligence*, pages 474–479.

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*, pages 142–147.

Satoshi Sekine. 1998. Description of the Japanese NE System used for MET-2. In *MUC-7*, Fairfax, Virginia.

R. Srihari, Cheng Niu, and W. Li. 2002. A Hybrid Approach for Named Entity and Sub-type Tagging. In *Proceedings of Sixth Conference on Applied Natural Language Processing (ANLP)*, pages 247–254.

M. Srinivas and L. M. Patnaik. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernatics*, 24(4):656–667.