Feature Subset Selection Using Genetic Algorithm for Named Entity Recognition

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

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

Abstract. In this paper, genetic algorithm (GA) is utilized to search for the appropriate feature combination for constructing a maximum entropy (ME) based classifier for named entity recognition (NER). Features are encoded in the chromosomes. The ME classifier is evaluated for the 3-fold cross validation with the features, encoded in a particular chromosome, and its average F-measure value is used as the fitness value of the corresponding chromosome. The proposed technique is evaluated for determining the suitable feature combinations for NER in three resource-constrained languages, namely Bengali, Hindi and Telugu. Evaluation results show the effectiveness of the proposed approach with the overall recall, precision and F-measure values of 71.27%, 83.95% and 77.09%, respectively for Bengali, 74.72%, 87.15% and 80.46%, respectively for Hindi and 60.91%, 94.15% and 73.97%, respectively for Telugu.

Keywords: Genetic algorithm, Feature Selection, Maximum Entropy, Named Entity Recognition.

1 Introduction

Named Entity Recognition (NER) is a well-established task that has immense importance in many Natural Language Processing (NLP) application areas such as Information Retrieval, Information Extraction, Machine Translation, Question Answering and Automatic Summarization (Babych and Hartley, 2003; Nobata *et al.*, 2002) 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.

** All authors have equal contributions

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 languages can be found in (Ekbal and Bandyopadhyay, 2007; Ekbal and Bandyopadhyay, 2009a; Ekbal and Bandyopadhyay, 2009b) for Bengali, in (Li and McCallum, 2004) for Hindi and in (Shishtla *et al.*, 2008) for Telugu.

The performance of any classification technique depends on the features of 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 a machine learning approach, feature selection is an optimization problem that involves choosing an appropriate feature subset. In ME model, appropriate feature selection is a very crucial problem and also a key issue to improve classifier's performance. However, it does not provide any method for automatic feature selection and heuristics are usually used for this task. In this paper, we propose a feature selection technique for ME based NER using the search capability of genetic algorithm (GA) (Goldberg, 1989).

Genetic algorithms (GAs) (Goldberg, 1989) are randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. GAs perform 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* and *fitness* function is 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 process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

In this paper we consider different contextual and orthographic word-level features. These features are language independent in nature, and can be very easily derived for almost all the languages with a very little effort. Thereafter GA is used to search for the appropriate feature selection. Here, features are encoded in the chromosomes with binary encoding scheme. Adaptive mutation and crossover operators are used to accelerate the convergence of GA. We also use elitism. In order to compute the fitness of each chromosome, ME classifier is evaluated with the features encoded in the particular chromosome and the average F-measure value is calculated for the 3-fold cross validation on training data.

The proposed approach is evaluated for three resource-constrained languages, namely Bengali, Hindi and Telugu. 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. Telugu is one of the popular languages and predominantly spoken in the *southern* part of India. Evaluation results show the effectiveness of the proposed approach with the overall recall, precision and F-measure values of 71.27%, 83.95% and 77.09%, respectively for Bengali, 74.72%, 87.15% and 80.46%, respectively for Hindi and

60.91%, 94.15% and 73.97%, respectively for Telugu.

2 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 obtained for almost all the 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, Hindi and Telugu. 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, its 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 newswire data.
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, 10 and 5 for Bengali, Hindi and Telugu, 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 (Ekbal and Bandyopadhyay, 2008a) 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 Proposed Approach

The proposed GA based feature selection technique is described below. The basic steps of the proposed approach, that closely follow those of the conventional GA, are shown in Figure 2.

3.1 String Representation and Population Initialization

If the total number of features is F, then the length of the chromosome is F. As an example, the encoding of a particular chromosome is represented in Figure 1. Here $F = 12$ (i.e., total 12 different features are available). The chromosome represents the use of 7 features for constructing a classifier (first, third, fourth, seventh, tenth, eleventh and twelfth features). 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} feature does not participate in constructing the classifier. Else if it is 1 then the i^{th} feature participates in constructing the classifier.

If the population size is P then all the P number of chromosomes of this population are initialized in the above way.

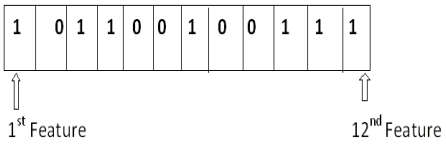

Figure 1: Chromosome representation for GA based feature selection

3.2 Fitness Computation

For the fitness computation, the following procedure is executed.

1. Suppose there are N number of features present in a particular chromosome (i.e., there are total N number of 1's in that chromosome).
2. Construct a classifier with only these N features.
3. Here, initially the training data is divided into 3 parts. The above classifier is trained using 2/3 of the training set with the features encoded in that chromosome and tested with the remaining 1/3 part.
4. Now, the overall F-measure value of this classifier for the 1/3 training data is calculated.
5. Steps 2 and 3 are repeated 3 times to perform 3-fold cross validation.
6. The average F-measure value of this 3-fold cross validation is used as the fitness value of the particular chromosome. The objective is to maximize this fitness value using the search capability of GA.

3.3 Selection

Roulette wheel selection is used to implement the proportional selection strategy.

3.4 Crossover

Here, we use the normal single point crossover (Holland, 1975). As an example, let the two chromosomes be:
$P1$: 0 1 1 0 0 0 1 1 1 0 1 0
$P2$: 1 0 1 1 0 0 0 0 0 0 1 0
At first a crossover point has to be selected randomly between 1 to 12 (length of the chromosome) by generating some random number between 1 and 12. Let the crossover point, here, be 4. Then after crossover, the two new offsprings are:
$O1$: 0 1 1 0 0 0 0 0 0 0 1 0 (taking the first 4 positions from $P1$ and rest from $P2$)
$O2$: 1 0 1 1 0 0 1 1 1 0 1 0 (taking the first 4 positions from $P1$ and rest from $P2$)

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:
$\mu_c = k_1 \times \frac{(f_{max}-f^{'})}{(f_{max}-\overline{f})}$, if $f^{'} > \overline{f}$,
$\mu_c = k_3$, if $f^{'} \leq \overline{f}$.
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.

3.5 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:
$\mu_m = k_2 \times \frac{(f_{max}-f)}{(f_{max}-\overline{f})}$ if $f > \overline{f}$,
$\mu_m = k_4$ if $f \leq \overline{f}$.
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 and 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 get higher values for low fitness solutions. While the high fitness solutions aid in the convergence of the 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. For a solution with the maximum fitness value, μ_c and μ_m are both zero. 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. Here, mutation operator is applied to each entry of the chromosome where the entry is randomly replaced by either 0 or 1.

3.6 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 up to the last generation provides the solution to the above feature selection problem. Elitism is implemented at each generation by preserving the best string seen up to that generation in a location outside the population. Thus on termination, this location contains the best feature combination.

4 Experimental Results and Discussions

We use the manually annotated data for Bengali. In addition, we use the IJCNLP-08 Shared Task on South and South East Asian Languages (NERSSEAL) data for Bengali, Hindi and Telugu. The ME framework estimates probabilities based on 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.

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 2: Basic Steps of GA

We use the OpenNLP Java based ME package [1]. Model parameters are computed with 200 iterations without feature frequency cutoff. We set the following parameter values for GA: population size=100, number of generations=50, probability of mutation=0.2 and probability of crossover=0.9.

4.1 Datasets for NER

Indian languages are resource-constrained in nature. 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. A portion of this corpus 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)[2] Shared Task data of around 100K wordforms that were originally annotated with a fine-grained tagset of twelve tags. This data is mostly from the *agriculture* and *scientific* domains. For Hindi and Telugu, we use the IJCNLP-08 NERSSEAL shared task datasets. The shared task datasets were originally annotated with a fine-grained NE tagset of twelve tags. The underlying reason to adopt this finer NE tagset was to use the NER system in various NLP applications, particularly in machine translation. One important aspect of the shared task was to identify and classify the maximal NEs as well as the nested NEs, i.e. the constituent parts of a larger NE. But, the training data were provided with the type of the maximal NE only. For example, *mahatmA gAndhi roDa* (Mahatma Gandhi Road) was annotated as location and assigned the tag 'NEL' even if *mahatmA* (Mahatma) and *gAndhi*(Gandhi) are NE title person (NETP) and person name (NEP), respectively. Henceforth, all the Bengali glosses are written using ITRANS notation[3]. The task was to identify *mahatmA gAndhi roDa* as a NE and classify it as NEL. In addition, *mahatmA* and *gAndhi* were to be recognized as NEs of the categories NETP (Title person) and NEP (Person name), respectively.

In the present work, we consider only the tags that denote person names (NEP), location names (NEL), organization names (NEO), number expressions (NEN), time expressions (NETI) and measurement expressions (NEM). The NEN, NETI and NEM tags are mapped to the MISC tag that

[1] http://maxent.sourceforge.net/
[2] http://ltrc.iiit.ac.in/ner-ssea-08
[3] http://www.aczone.com/itrans/

denotes miscellaneous entities. Other tags of the shared task are mapped to the 'other-than-NE' category denoted by 'O'. Hence, the tagset mapping now becomes as shown in Table 1.

Table 1: Tagset mapping table

IJCNLP-08 shared task tag	Coarse-grained tag	Meaning
NEP	PER	Person name
NEL	LOC	Location name
NEO	ORG	Organization name
NEN, NEM, NETI	MISC	Miscellaneous name
NED, NEA, NEB, NETP, NETE	O	Other than NEs

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 (Tjong Kim Sang and De Meulder, 2003). A portion of each datasets has been used for training and the remaining portion is used to report the evaluation results. Some statistics of training and test sets are presented in Table 2.

Table 2: Statistics of the datasets

Language	No. of words in training	No. of NEs in training	No. of words in test	No. of NEs in test
Bengali	312,947	37,009	31,845	4,413
Hindi	496,496	27,650	6,438	461
Telugu	57,179	4,470	6,847	662

5 Discussion of Results

We use various subsets of the following features for constructing the different classifiers based on the ME framework.
(i). Various context word window within the previous three and next three words (ii). Prefixes of length upto three (3 features) or four (4 features) characters (iii). Suffixes of length upto three (3 features) or four (4 features) characters (iv). Part of Speech (POS) information (v). First word of the sentence (vi). Length of the word (vii). Infrequent word (viii). Position of the word and (ix). Various digit features (digitComma, digitPercentage, digitDot, digitSlash, digitHyphen, digitFour and digitTwo).

Initially, we construct following six *baseline* classifiers based on the ME framework using various randomly selected subsets of the above mentioned feature set. Here, $C[-i,+j]$ denotes the context spanning from the previous i^{th} word to the next j^{th} word with the current token at position 0; Pre_i and Suf_i denote the prefixes and suffixes of character sequences up to i of the current word, respectively.

1. *Baseline 1*: $C[-2,+2]$, Pre_3, Suf_3, and the features (iv)-(ix).
2. *Baseline 2*: $C[-2,+2]$, Pre_4, Suf_4, and the features (iv)-(ix).
3. *Baseline 3*: $C[-3,+3]$, Pre_3, Suf_3, and the features (iv)-(ix).

4. *Baseline 4*: $C[-3,+3]$, Pre_4, Suf_4, and the features (iv)-(ix).
5. *Baseline 5*: $C[-1,+1]$, Pre_3, Suf_3, and the features (iv)-(ix).
6. *Baseline 6*: $C[-1,+1]$, Pre_4, Suf_4, and the features (iv)-(ix).

Thereafter, we apply our proposed GA based feature selection technique for NER in three Indian languages, namely Bengali, Hindi and Telugu. The proposed approach finally selects the features as shown in Table 3. The ME classifier is then evaluated with the corresponding test set with the best set of features as identified by the proposed technique. Overall evaluation results along with the *baseline* models are reported in Table 4, Table 5 and Table 6 for Bengali, Hindi and Telugu, respectively. Evaluation of our proposed feature selection algorithm shows the state-of-the-art performance for all the three languages. It yields the overall recall, precision and F-measure values of 71.27%, 83.95% and 77.09%, respectively for Bengali, 74.72%, 87.15% and 80.46%, respectively for Hindi, and 60.91%, 94.15% and 73.97% respectively for Telugu. Results also show that the ME model trained using the feature set automatically identified by the proposed approach performs better than the other six *baseline* models for all the languages. This shows that appropriate feature selection using GA based technique works better compared to the heuristics based manual feature selection in ME framework.

Table 3: Features identified by the proposed GA based approach

Language	Features
Bengali	$C[-2,+2]$, Pre_3, Suf_3, POS, digitDot, digitSlash and digitHyphen
Hindi	$C[-1,+1]$, Suf_4, Pre_4, POS, Infrequent word, digitComma, digitDot and digitSlash
Telegu	$C[-1,+1]$, Suf_3, Pre_4, POS, digitDot and digitSlash

Table 4: Overall results for Bengali

Model	recall (in %)	precision (in %)	F-measure (in %)
GA based approach	71.27	83.95	77.09
Baseline 1	71.15	81.53	75.99
Baseline 2	69.76	81.75	75.28
Baseline 3	70.28	80.93	75.23
Baseline 4	43.81	73.35	54.86
Baseline 5	70.03	83.08	76.00
Baseline 6	60.65	80.16	69.05

Table 5: Overall results for Hindi

Model	recall(in %)	precision (in %)	F-measure (in %)
GA based approach	74.72	87.15	80.46
Baseline 1	62.39	80.63	70.35
Baseline 2	51.93	80.29	63.07
Baseline 3	59.99	81.07	68.95
Baseline 4	48.94	80.05	60.75
Baseline 5	65.32	80.61	72.16
Baseline 6	57.29	80.96	67.10

Statistical analysis of variance, (ANOVA) (Anderson and Scolve, 1978) is performed in order to examine whether the GA based feature selection technique really outperforms the several *baseline*

Table 6: Overall results for Telugu

Model	recall(in %)	precision (in %)	F-measure (in %)
GA based approach	60.91	94.15	73.97
Baseline 1	50.89	91.55	65.42
Baseline 2	41.97	93.21	57.88
Baseline 3	46.81	91.96	62.04
Baseline 4	40.01	81.77	53.73
Baseline 5	54.21	92.21	68.28
Baseline 6	48.17	91.72	63.17

ensemble techniques. 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. This again justifies our observation that the proposed MOO based feature selection technique performs much better than the several *baseline* approaches.

It will not be fair to compare the performance of our proposed system with that of the previous proposals (Ekbal and Bandyopadhyay, 2009b; Saha *et al.*, 2008; Srikanth and Murthy, 2008) as these works use either (i). different data sets or, (ii). different experimental set up or, (iii). more complex set of features or, (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 thus 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 NE annotated data to the coarse-grained forms. Thus, comparing our proposed system with that of the shared task papers [4] is also out of scope.

6 Conclusions and Future Works

In this paper, we proposed a GA based feature selection technique for ME based NER. Features have been encoded in a chromosome. The average F-measure value of the ME classifier trained using the feature set encoded in a particular chromosome has been used as the fitness value of that particular chromosome. One most appealing characteristic of our system is that it makes use of the features that are language independent in nature, and can be easily obtained for many languages. Here, we evaluated our proposed technique for three resource-constrained Indian languages, namely Bengali, Hindi and Telugu. Evaluation results the overall recall, precision and F-measure values of 71.27%, 83.95% and 77.09%, respectively for Bengali, 74.27%, 87.15% and 80.46%, respectively for Hindi and 60.91%, 94.15% and 73.97%, respectively for Telugu.

In future we would evaluate the proposed technique by incorporating some more language independent features. We would also include language dependent features, extracted from the language dependent resources and/or tools. Future works also include investigating best feature combinations for some other well-known classifiers like Conditional Random Field and Support Vector Machines.

References

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

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

Babych, Bogdan and A. Hartley. 2003. Improving Machine Translation Quality with Automatic Named Entity Recognition. In *Proceedings of EAMT/EACL 2003 Workshop on MT and other Language Technology Tools*, pp. 1–8.

Ekbal, A. 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)*, pp. 123–128, India.

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.

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

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

Li, Wei and Andrew McCallum. 2004. 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.

Nobata, C., S. Sekine, H. Isahara, and R. Grishman. 2002. Summarization System Integrated with Named Entity Tagging and IE Pattern Discovery. In *Proceedings of Third International Conference on Language Resources and Evaluation (LREC 2002)*, Spain.

Saha, S., S. Sarkar, and P. Mitra. 2008. A Hybrid Feature Set based Maximum Entropy Hindi Named Entiy Recognition. In *Proceedings of the 3rd International Joint Conference in Natural Langauge Processing (IJCNLP 2008)*, pp. 343–350.

Shishtla, Praneeth M, Prasad Pingali, and Vasudeva Varma. 2008. A Character n-gram Based Approach for Improved Recall in Indian Language NER. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages*, pp. 101–108.

Srikanth, P and Kavi Narayana Murthy. 2008. Named Entity Recognition for Telugu. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages*, pp. 41–50.

Srinivas, M. 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.

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.