Soft Comput (2015) 19:3529–3549
DOI 10.1007/s00500-014-1565-5

FOCUS

MODE: multiobjective differential evolution for feature selection and classifier ensemble

Utpal Kumar Sikdar · Asif Ekbal · Sriparna Saha

Published online: 10 January 2015

Abstract In this paper, we propose a multiobjective differential evolution (MODE)-based feature selection and ensemble learning approaches for entity extraction in biomedical texts. The first step of the algorithm concerns with the problem of automatic feature selection in a machine learning framework, namely conditional random field. The final Pareto optimal front which is obtained as an output of the feature selection module contains a set of solutions, each of which represents a particular feature representation. In the second step of our algorithm, we combine a subset of these classifiers using a MODE-based ensemble technique. Our experiments on three benchmark datasets namely GENIA, GENETAG and AIMed show the F-measure values of 76.75, 94.15 and 91.91 %, respectively. Comparisons with the existing systems show that our proposed algorithm achieves the performance levels which are at par with the state of the art. These results also exhibit that our method is general in nature and because of this it performs well across the several domain of datasets. The key contribution of this work is the development of MODE-based generalized feature selection and ensemble learning techniques with the aim of extracting entities from the biomedical texts of several domains.

Communicated by E. Lughofer.

U. K. Sikdar · A. Ekbal (✉) · S. Saha
Department of Computer Science and Engineering,
Indian Institute of Technology Patna, Patna, Bihar, India
e-mail: asif.ekbal@gmail.com; asif@iitp.ac.in

U. K. Sikdar
e-mail: utpal.sikdar@iitp.ac.in

S. Saha
e-mail: sriparna@iitp.ac.in

1 Introduction

Entity extraction in the biomedical domain aims to identify and classify each word of a document into some predefined target categories such as protein, RNA, DNA, cell_type and cell_line. Entity extraction is an important component in many text mining applications including question–answering, information extraction, information retrieval and automatic summarization. Accurate prediction of biomedical entities is crucial for its integration in a text mining system, which is deployed to solve some practical application(s). Similar to the other domains, biomedical names also belong to the open class of expressions, i.e., there is an infinite variety and new expressions are constantly being invented. But, compared to the traditional news-wire domain, identification and classification of biomedical entities are more challenging because of the following facts: names are, in general, very long and complex and hence their boundary identification is more difficult; names contain many nested and compounded entities with symbols, punctuation marks, etc., and therefore require more sophisticated features for their accurate classification. Another crucial issue is to develop a domain-independent system that is general enough to handle entity extraction for more than one domain. In biomedical domain there exists several benchmark corpora that were developed following different annotation guidelines. Therefore the system developed for a particular domain of dataset often fails to perform reasonably on the other domain.

In this paper, we develop a multiobjective differential evolution (MODE)-based algorithm that performs feature selection and ensemble learning in sequence. In the first step, we determine the most relevant features for the target task within the framework of a supervised machine learning algorithm. Feature selection is the technique of selecting a subset of relevant features for building a robust learning model. As we

already mentioned, our feature selection algorithm is based on the concept of multiobjective optimization (MOO) that makes use of differential evolution (DE) (Storn and Price 1997) as an underlying optimization technique. DE (Storn and Price 1997) is a parallel direct search method which performs search in complex, large and multi-modal landscapes, and provides near-optimal solutions for an optimization problem. Feature selection is traditionally framed as a single objective optimization (SOO) problem. In SOO, we focus on optimizing only one objective function at a time. In contrast, MOO concerns with the optimization of more than one objective function simultaneously. For MODE-based feature selection, we optimize two functions as follows: minimize the number of features and maximize the F-measure value. Features are encoded in the form of a chromosome. Traditional crossover and mutation operators are used. But a new selection operator is developed to deal with the MODE. The final output of feature selection is a Pareto optimal front that contains a set of potential solutions, each of which represents a different classifier. We perform feature selection for a well-known machine learning algorithm, namely conditional random field (CRF) (Lafferty et al. 2001). The algorithm proposed here is general enough to be applicable for any other supervised machine learning algorithm.

The optimization algorithms have been successfully applied for solving problems from different fields like manufacturing industry, swarm intelligence and fuzzy control system. In Victor et al. (2005), authors have surveyed the recently developed evolutionary algorithms for solving some real-world problems related to manufacturing industry. In Preitl and Precup (2006), authors have dealt with both theoretical and application aspects concerning iterative feedback tuning algorithms in the design of a class of fuzzy control systems. In Heidl et al. (2013), authors have described some machine learning based analysis and design methods which were applied for studying gender differences in visual inspection decision making. In El-Hefnawy (2014), authors have suggested a modified particle swarm optimizer for solving fuzzy bi-level single and multiobjective problems.

Literature shows how several algorithms for feature selection were developed using the search capability of genetic algorithm (GA) (Goldberg 1989). These algorithms are mostly applied for solving the problems related to pattern classification and knowledge discovery. GA (Goldberg 1989) is a randomized search and optimization technique guided by the principles of evolution and genetics, having a large amount of implicit parallelism. GAs are adaptive computational procedures modeled on the mechanics of natural genetic systems. They express their abilities by efficiently exploiting the historical information to speculate on new offsprings with expected improved performance (Goldberg 1989). In Yang and Honavar (1998), a GA-based feature selection technique was developed to select the appropriate subset of features from the different data sets. The features represent financial cost, diagnostic value, risks, etc. In Oliveira et al. (2001), GA is used to select the appropriate subset of features for handwritten digit recognition. The feature vector is consisting of a mixture of concavity and contour-based features. The paper (Dash and Liu 1997) surveys the works done in the domain of feature selection starting from the early 1970s. In this paper, different types of evaluation functions were compared based on the different properties. In Guyon and Elisseeff (2003), variables and features are identified from the datasets with tens, hundreds or thousands of available variables. The algorithm is applied in the domains of text processing of internet documents, gene expression array analysis, and combinatorial chemistry.

We generate CRF-based classifiers from the features represented in the chromosomes of the solutions obtained in the first step. Some of the important classifiers are selected and combined using a MODE-based classifier ensemble technique. The main idea behind classifier ensemble is that ensembles are often much more accurate than the individual classifiers that make them up. An important issue in ensemble learning is to investigate the most suitable way to combine the decisions of the classifiers. There are two conventional methods for combining the classifiers (Dasarathy and Sheela 1979; Dietterich 2000): majority voting and weighted voting. While in majority voting same weights are assigned to all the classifiers, in weighted voting classifiers are combined using some weights. Depending upon how the weights are determined in weighted voting, final decision of the ensemble could vary, and that, in turn, affects the overall classification performance. In reality, all the participant classifiers may not be equally efficient to detect all the target classes. For example, a classifier may be good at detecting DNA while others may be good at detecting RNA. Thus, while combining the classifiers using weighted voting, weights of voting should vary among the different classes for a classifier. Here, we develop a MODE-based technique that can automatically determine the appropriate weights of voting for each class in any classifier.

In recent past, there have been some efforts (Ekbal and Saha 2011b, 2012) for building evolutionary algorithms based on feature selection and ensemble learning techniques, especially focusing on text processing domains. A SOO-based classifier ensemble technique was proposed in Ekbal and Saha (2011b). This was evaluated for named entity (NE) extraction from multiple natural language texts. In addition, a GA-based feature selection technique was also developed. In Ekbal and Saha (2010a), a GA-based classifier ensemble selection technique was developed. This approach determines only a subset of classifiers that can form the final classifier ensemble.

In Ekbal and Saha (2012), a multiobjective GA-based ensemble technique was developed. Along with feature selec-

tion, exhaustive evaluation was also carried out. In Ekbal and Saha (2011a), a simulated annealing-based MOO technique, AMOSA Bandyopadhyay et al. (2008) was used to develop an ensemble method. Several different versions of the objective functions were exploited. In Sikdar et al. (2012), a DE-based feature selection and classifier ensemble technique was developed. This algorithm was based on SOO. The present work deals with MODE, which has a different perspective compared to GA or simulated annealing.

We perform experiments on three existing benchmark datasets namely JNLPBA 2004 shared task,[1] GENETAG[2] and AIMed.[3] Evaluation results show the F-measure values of 76.75, 94.15 and 91.91 % for the JNLPBA, GENETAG and AIMed datasets, respectively. Comparisons with the existing systems show that our proposed technique achieves performance at par with the existing state-of-the-art systems.

Below, we highlight how the current research differs with the prior work reported in Sikdar et al. (2012):

- In Sikdar et al. (2012), we tackled the problem of appropriate feature selection and ensemble learning using DE-based SOO. But in the current paper, the automatic feature and classifier ensemble selection problem is modeled as a MOO problem. As an optimization technique we use DE. SOO has a different perspective compared to MOO. In SOO only one function is optimized at a time, but in MOO multiple functions are optimized simultaneously. Another advantage of MOO is that it produces a set of trade-off solutions, and depending on requirement of the user, a single solution may be selected. In the current paper, at first we extend the single objective DE to solve the MOO problem. Thereafter, it is used to solve the problem of automatic feature selection and classifier ensemble. Existing literature shows that, in general, MOO-based techniques are more effective than the SOO-based techniques for solving the difficult optimization problems. In the current paper, we also show the effectiveness of MOO for the target task. Thus, the proposed technique is substantially different from the approach proposed in Sikdar et al. (2012).
- The algorithm proposed in Sikdar et al. (2012) was applied for evaluating NE extraction problem in three Indian languages, namely Bengali, Hindi and Telugu. But in the current paper we have evaluated the algorithms for biomedical entity extraction from three benchmark datasets, namely JNLPBA 2004 shared task,[4] GENETAG[5] and AIMed.[6] Entity extraction in biomedical texts is inherently more challenging compared to the other traditional domains such as newswire.
- In Sikdar et al. (2012), we used less number of features as compared to our current method. The task handled in Sikdar et al. (2012) was to extract named entities from the Indian language texts and therefore made use of a feature set that is much smaller compared to what we used in our current work.

2 Overview of multiobjective differential evolution

Differential evolution Storn and Price (1997) is a parallel direct search method which performs search in complex, large and multi-modal landscapes, and in general provides near-optimal solutions for an optimization problem. In DE, the parameters of the search space are encoded in the form of strings called chromosomes. A collection of such strings is called a population denoted by NP. It is a collection of $|NP|$ D-dimensional parameter vectors $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}]$, $i = 1, 2, \ldots, NP$ for each generation G. The value of D represents the number of real parameters on which optimization or fitness function depends. The value of NP does not change during the optimization process. The initial vector population is chosen randomly which represents different points in the search space and should cover the entire parameter space. For MOO, more than one objective or fitness functions are associated with each string. These represent the degrees of goodness of the string. Differential evolution generates new parameter vector by adding the weighted difference between two population vectors to a third vector. This operation is called mutation. The mutated vector's parameters are then mixed with the parameters of another predetermined vector, the target vector, to yield the so-called trial vector. Parameter mixing is often referred to as crossover. For selection, these NP number of trial vectors are merged to the current population. Hence the total number of solutions becomes $2 \times NP$. The solutions are ranked based on the concept of domination and non-domination. In the next generation we have to select NP number of chromosomes from the entire set of solutions. The process starts to include the solutions from the first rank. If it exceeds NP we sort the solutions using the crowding distance sorting algorithm. Thereafter, we keep on including the solutions until it becomes equal to NP. The rest of the solutions of the first rank are not considered thereafter. If the number of solutions of the first rank is less than NP, we select the solutions from the subsequent ranks until the total number reaches our desired count, i.e., NP. Following are the steps

[1] http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm.

[2] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENEATG.tar.gz.

[3] ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/interactions.tar.gz.

[4] http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm.

[5] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENEATG.tar.gz.

[6] ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/interactions.tar.gz.

for computing the crowding distance d_i of each point i in the non-dominated front I (Deb et al. 2002)

- For $i = 1, \ldots, I$, initialize $d_i = 0$.
- For each objective function $f_k, k = 1, \ldots, K$, do the following:
  - Sort the set I according to f_k in ascending order.
  - Set $d_1 = d_{|I|} = \infty$.
  - For $j = 2$ to $(|I| - 1)$, set $d_j = d_j + (f_{k(j+1)} - f_{k(j-1)})$.

In the proposed multiobjective DE, a binary tournament selection operator which was defined based on the crowding distance operator is used. If two solutions a and b are compared during a tournament, then solution a wins the tournament if either:

- The rank of a is better (less) than the rank of b, i.e., a and b belong to two different non-dominated fronts, or
- The ranks of a and b are the same (i.e., they belong to the same non-dominated front) and a has higher crowding distance than b. This means that, if two solutions belong to the same non-dominated front, the solution situated in the lesser crowded region is selected.

The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied. The pseudocode of the multiobjective differential evolution is shown in Algorithm 1.

Algorithm 1 Pseudocode for Multiobjective Differential Evolution

```
G=0
Create a random initial population X_{i,G}, ∀i, i = 1, ..., NP
for G=1 to G_Max do
  for i=1 to NP do
    U_{i,G+1} = X_{i,G}
  end for
  for i=1 to NP do
    Select randomly three different chromosomes r1, r2 and r3
    i_rand = randint(1,D)/*generate a random integer value from 1 to D */
    for j=1 to D do
      rand_j = randfloat(0,1)/*generate a random real value belongs to [0,1]*/
      if rand_j < CR or j=i_rand then
        u_{NP+i,j,G+1} = x_{r3,j,G} + F × (x_{r1,j,G} − x_{r2,j,G})
      else
        u_{NP+i,j,G+1} = x_{i,j,G}
      end if
    end for
  end for
  /* Evaluate the value of K objective/fitness functions */
  Evaluate f_k(U_{i,G+1}) ∀i, i = 1, ..., 2 × NP and ∀k, k = 1, ..., K
  n = 0
  j = 1
  while n < NP do
    Select all the non-dominated solutions V_{p,G+1} of rank_j from U_{i,G+1}, ∀i, i = 1, ..., 2 × NP and ∀p, p = 1, ..., I where 1 ≤ I ≤ 2 × NP
    if n + k ≤ NP then
      for i=n+1 to n+k do
        X_{i,G+1} = V_{i−n,G+1}
      end for
    else
      Apply crowding distance sorting to V_{p,G+1}
      for i=n+1 to NP do
        X_{i,G+1} = V_{i−n,G+1}
      end for
    end if
    n=n+k
    j=j+1
  end while
end for
```

3 Method for feature selection

In this section, we first formulate the problem of relevant feature selection within the framework of multiobjective differential evolution, and then present the proposed approach.

3.1 Problem formulation for feature selection

Suppose, there are D number of available features, and these are denoted by $F_1, \ldots, F_D$. Let, $\mathcal{A} = \{F_i : i = 1; D\}$. The feature selection problem is then stated as follows:

Determine the appropriate subset of features $\mathcal{A}' \subseteq \mathcal{A}$ such that the classifier trained using these features should have optimized some metrics. In our proposed MOO-based DE setting, we optimize two objective functions namely (1) minimize the number of features and (2) maximize the F-measure value. Please note that we determine the optimal feature combinations based on the development set, and later on use these to perform blind evaluation on the test data.

3.2 Multiobjective DE-based feature selection approach

The basic steps of the multiobjective differential evolution technique are as follows.

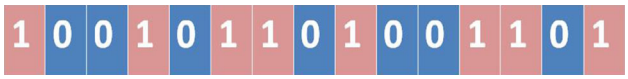

Fig. 1 Problem encoding for feature selection

Chromosome representation and population initialization In multiobjective DE, features are encoded in the chromosomes. If we have D features then we require a vector of length D for its representation.

Initially, all the chromosomes are randomly initialized to either 1 or 0. A value of 1 in the ith position indicates that the respective feature participates in constructing the classifier. Else, the value of 0 indicates that the corresponding feature does not participate in constructing the classifier. The chromosome representation and initialization are shown in Fig. 1.

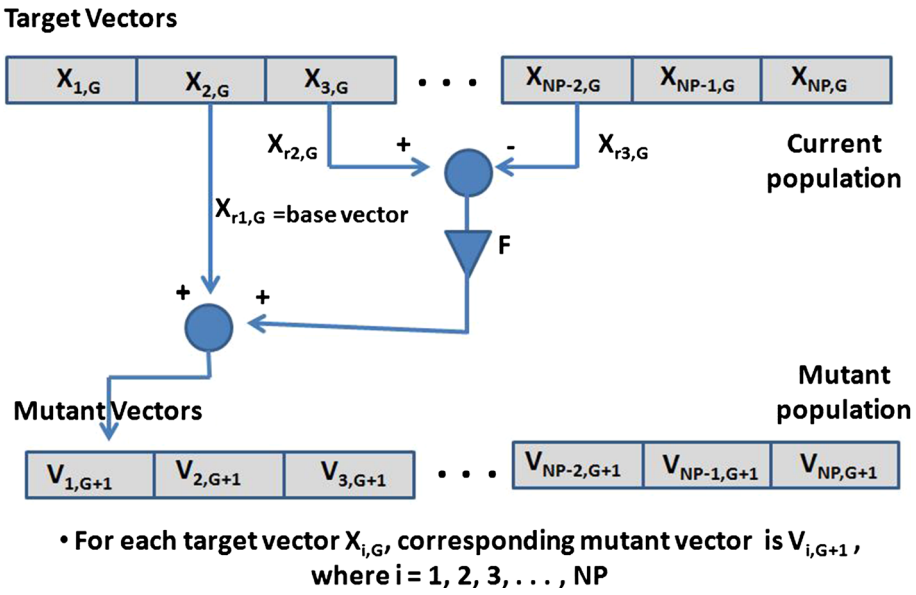

Fig. 2 Multiobjective DE-based mutation process

In this Fig. 1, the value of D is 15 and each gene value of the chromosome is either '1' or '0' that represents the presence or absence of the corresponding feature.

Fitness computation

For the fitness computation, if there are D features present in the chromosome then the classifier is trained with only these features. The trained model is evaluated on the development set.[7] We compute the F-measure value for the development set. Our aim is to select the minimal set of features which will maximize the F-measure value. We optimized the following functions: (1) minimize the number of features present in the chromosome, and (2) maximize the F-measure value.

Mutation

In multiobjective DE, for each target vector $X_{i,G}$; $i = 1, 2, 3, \ldots, NP$, a mutant vector is generated according to

$$V_{i,G+1} = x_{r1,G} + F \times (x_{r2,G} - x_{r3,G}), \tag{1}$$

where $r1$, $r2$, $r3$ are mutually different random indices and belong to $\{1, 2, \ldots, NP\}$, G is the generation number and $F > 0$. The values of randomly chosen integers $r1$, $r2$ and $r3$ are different from the running current index i, so that NP must have value equal to at least four. The value of F is a real and constant factor and we set the value of F equals to 0.5, a parameter in the range of [0, 1] which controls the amplification of the differential variation $(x_{r2,G} - x_{r3,G})$. The $V_{i,G+1}$ is termed as the mutated vector. If it is found that the value of each parameter of the mutant vector $V_{i,G+1} \geq 0.5$ then we set the parameter value to 1, otherwise the parameter value is set to 0. A collection of NP number of mutant vectors is called the mutant population. The mutation operator is described in Fig. 2.

[7] A part of each training set is used as the development set.

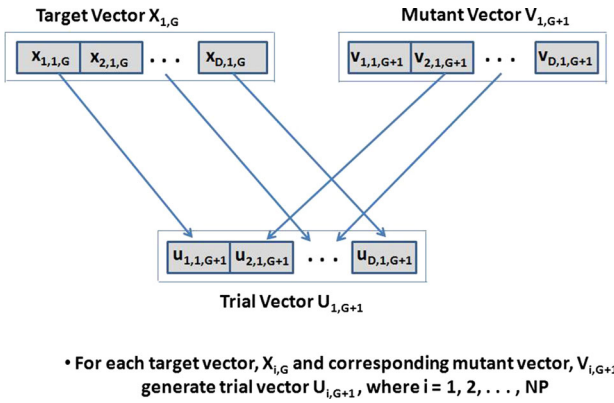

Fig. 3 Multiobjective DE-based crossover operator

Crossover or recombination

The parameter mixing of target vector $X_{i,G}$ and mutant vector $V_{i,G+1}$ is called crossover. Crossover is needed to increase the diversity of mutant vector. To this end, the trial vector:

$$U_{i,G+1} = (u_{1,i,G+1}, u_{2,i,G+1}, \ldots, u_{D,i,G+1}) \tag{2}$$

is generated, where

$$u_{j,i,G+1} = v_{j,i,G+1} \quad \text{if} \quad (rand_j \leq CR) \text{ or } j = i_{rand} \tag{3}$$

$$= x_{j,i,G} \quad \text{if} \quad (rand_j > CR) \text{ and } j \neq i_{rand} \tag{4}$$

for $j = 1, 2, \ldots, D$, In Eq. 3, $rand_j$ is an uniform random number of the jth evaluation which belongs to [0, 1]. CR is the crossover constant belonging to [0, 1] which has to be determined by the user. Here, CR value is 0.5. i_{rand} is a randomly chosen index, belonging to $\{1, 2, \ldots, D\}$ which ensures that the parameters of $U_{i,G+1}$ receive at least one parameter from $V_{i,G+1}$. At the end of this process, we will get the trial population. The crossover operator is shown in Fig. 3.

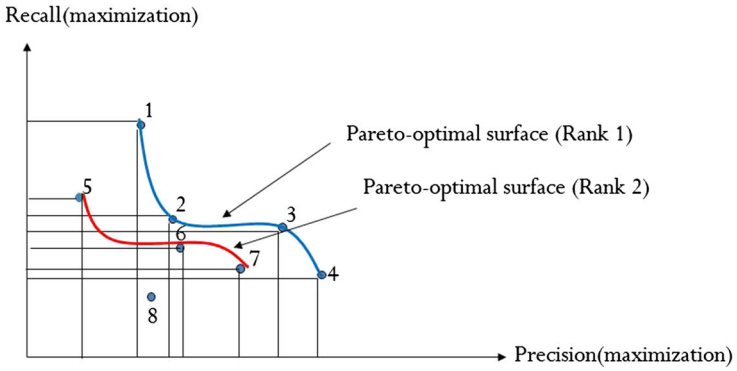

Fig. 4 Representation of dominated and non-dominated solutions

•Rank 1: Solutions 1, 2, 3 and 4 are non-dominating to each other.

•Rank 2: Solutions 5, 6 and 7 are non-dominating but dominated any one or more of Rank 1 solution.

•Rank 3: Solution 8 is dominated by any one or more solutions from Rank 1 and Rank 2 solutions.

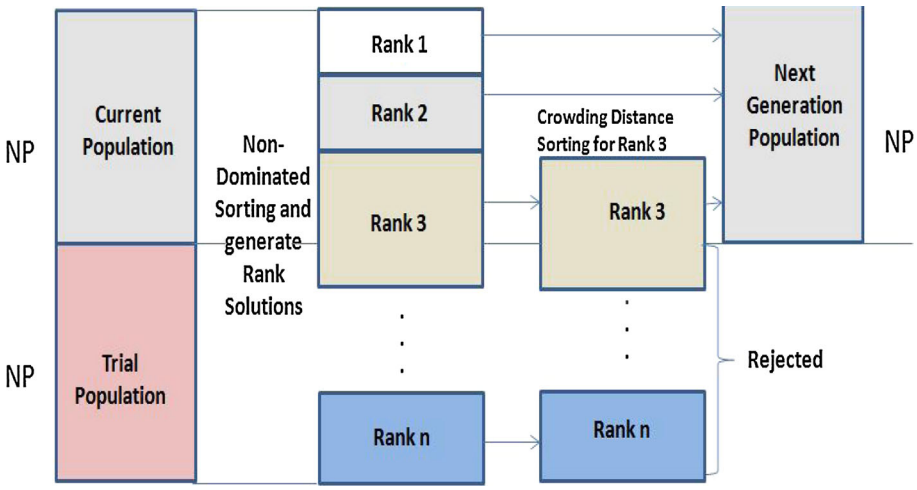

Fig. 5 Selection process

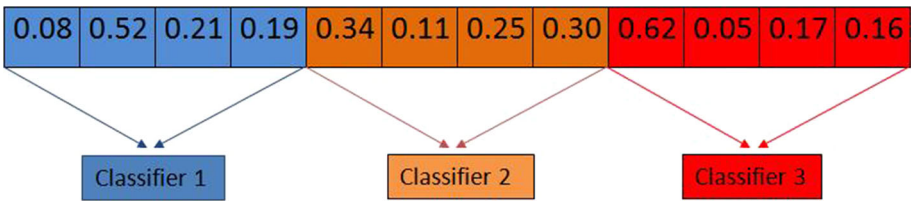

Fig. 6 Problem encoding for DE-based classifier ensemble

Selection

To decide NP number of best chromosomes for the next generation $G + 1$, trial population is merged to the current population. Thus there are $2 \times NP$ chromosomes. These $2 \times NP$ solutions are ranked based on the concept of domination and non-domination relations in the objective function space. The dominated and non-dominated relations are shown in Fig. 4. In this figure, non-dominated solutions are represented in the Pareto optimal surface. Thereafter, these solutions in the descending (rank 1 is considered to be at the top) order are added to the population of the next generation until the total number of solutions becomes equal to NP. If the number of solutions of a particular rank is more than NP then we apply the crowding distance sorting algorithm, and discard the excess solutions. At the end of this process, best NP number of chromosomes are selected for the next-generation population. The entire selection process is shown in Fig. 5.

Termination condition

The processes of mutation, crossover (or, recombination), fitness computation and selection are executed for a G_{Max} number of generations. Finally, we obtain a set of non-dominated solutions on the final Pareto optimal front. None of these solutions is better compared to the other, and each of these represents a set of optimal feature combinations. We construct multiple classifiers using these feature combinations.

4 Method for classifier ensemble

In the output of the first step of the algorithm, we obtain a set of solutions, each of which is equally important from the algorithmic point of view. We generate different classifiers using the feature combinations as represented in each of these candidate solutions. To combine these solutions, we propose an ensemble technique based on multiobjective DE.

4.1 Problem formulation

The weighted vote-based classifier ensemble problem (Ekbal and Saha 2010b) is stated below. Suppose, the N number of available classifiers be denoted by $C_1, \ldots, C_N$. Let, $\mathcal{A} = \{C_n : n = 1; N\}$ and there are M classes. The weighted vote-based classifier ensemble problem is then stated as follows:

Determine the voting weights V for each classifier that optimizes some function $f(V)$. The V denotes a real array of size $N \times M$. $V(n, m)$ is the weight of vote of the nth classifier for the mth class. The class for which the classifier is more confident receives more weight; whereas the class for which the classifier is less confident is assigned less weight. The weights computed in such a way are used to combine the outputs of the classifiers. Here, f_is are some classification quality measures of the ensemble classifier. The problem that we solve here has, in general, three different kinds of classification quality measures f_i, namely recall, precision and F-measure. Thus, $f \in$ {recall, precision, F-measure}.

The ensemble problem under multiobjective DE is formulated as follows. For each classifier, determine the voting weights V per classifier such that, maximize $[f(V)]$, where $f \in$ {recall, precision, F-measure}. We optimize $f =$ recall and precision as the two objective functions.

4.2 Steps of the proposed approach

The various steps of the proposed multiobjective DE algorithm are given below:

4.2.1 Problem representation

Chromosome representation and population initialization
Let us assume that the number of available classifiers and classes be N and M, respectively. The problem can be represented using a chromosome of length equal to $D = N \times M$. Each chromosome encodes the voting weights for possible M classes in each classifier.

As an example, in Fig. 6, a chromosome is represented with real encoding. We use real encoding, and the entries of each chromosome are randomly initialized to a real value (r) between 0 and 1. Here, $r = \frac{rand()}{RAND\_MAX+1}$. If the population size is NP then all the NP number of chromosomes of this population are initialized in the above way. Here, the values of N and M are 3 and 4, respectively. So, total $3 \times 4 = 12$ votes are possible. The chromosome represents the following ensemble:

Weights of votes for four different classes for classifier 1 are 0.08, 0.52, 0.21 and 0.19, respectively. Similarly, weights of votes for four different classes are 0.34, 0.11, 0.25 and 0.30, respectively, for classifier 2 and 0.62, 0.05, 0.17 and 0.16, respectively, for classifier 3.

Please note that for the feature selection problem, bits of a chromosome were encoded with the binary values.

Objective functions computation
The process of computing the objective functions follows the sequence of steps as mentioned below:

1. Suppose, N is the total number of classifiers. The classifiers' F-measure values for the development set be denoted by $F_n, n = 1, \ldots, N$.
2. For each token of the development set, we have M classes from the N classifiers (each output class is taken from a different classifier). Final output of the ensemble is determined using the voting weights of the classes of these N classifiers. The weight for a particular class predicted by the nth classifier is equal to F_n (i.e., F-measure value of the nth classifier). The final weight of a particular token t for a particular class is:

   $$g(c_m) = \sum F_n * Q(n, m),$$
   $$\forall n = 1 \text{ to } N \text{ and } op(t, n) = c_m$$

   Here, $Q(n, m)$ corresponds to the entry of the chromosome that represents the nth classifier and mth class, and $op(t, n)$ denotes the output class provided by the classifier n for the token t. Final prediction of a token depends on the maximum combined weight that a particular class receives. Let us consider the following example for the explanation:

   Examples
   Let us consider the chromosome in Fig. 6. Suppose the four classes be 'protein' (class 1), 'DNA' (class 2), 'RNA' (class 3) and 'cell_type' (class 4); and the F-measure values of three classifiers be 0.75, 0.82 and 0.70, respectively. Let for a token '*c-Fos*' three classifiers produce outputs as follows: classifier 1: 'DNA'; classifier 2:'protein'; classifier 3:'protein'. Then $f(\text{'DNA'}) = 0.75 \times 0.08 = 0.06$; and $f(\text{'protein'}) = 0.82 \times 0.34 + 0.70 \times 0.62 = 0.7128$. Thus, the final output selected for this particular token is 'protein' as $f(\text{'protein'}) > f(\text{'DNA'})$.
3. Compute the recall and precision of the ensemble classifier.
4. Use the *precision* and *recall* as the two objective functions. These are to be maximized using the search capability of multiobjective DE.

Table 1 Minimum, average and maximum F-measure values (last generation)

ME	JNLPBA04		GENETAG		AIMed	
	FS	CE	FS	CE	FS	CE
MinVal	74.83	75.88	92.36	93.81	90.18	91.06
MaxVal	75.26	76.75	93.77	94.15	90.56	91.91
Average	74.98	76.27	93.29	93.97	90.29	91.58

Here 'FS': Method for feature selection, 'CE': Method for classifiers ensemble, 'F': F-measure, ME: Method, 'MinVal', 'MaxVal' and 'Average': Denote the minimum, maximum and average F-measure values, respectively, for the solutions generated on the final Pareto front

Mutation

The mutation operation is performed in the same way as we did for feature selection. If the values of the mutant vector parameter violate the boundary constraints then values of the violating mutant vector parameter are reflected back from the violated boundary as follows:

- if($v_{j,i,G+1} < 0$) then $v_{j,i,G+1} = 2 \times \text{lower} - v_{j,i,G+1}$; where lower $= 0$;
- if($v_{j,i,G+1} > 1$) then $v_{j,i,G+1} = 2 \times \text{upper} - v_{j,i,G+1}$; where upper $= 1$;

where $j = 1, 2, \ldots, D$ and $i = 1, 2, \ldots, NP$.

Other operators

Other operators of the multiobjective DE are similar to those of multiobjective DE-based feature selection technique that we described in the previous section.

Selecting final solution from the pareto optimal front

The MOO algorithm produces a set of non-dominated solutions (Deb 2001) on the final Pareto optimal front. None of these solutions dominates the other, and each provides a way of combining the participating classifiers. Some of these solutions are better with respect to recall; whereas some are better with respect to precision. Though all the solutions are equally important from the algorithmic point of view, we often require to choose the most appropriate solution for the target problem.

Consequently, in this paper we select the best solution based on the F-measure value. The ensemble is evaluated on the development set. Each solution of the Pareto front represents a weight combination along with the recall and precision values of the ensemble classifier, which is constructed with the weight combinations as represented in the chromosome. We compute the F-measure scores of all the solutions of the Pareto front. Finally, we select the particular solution that yields the best F-measure value. Final results on the test data are reported using the classifier ensemble corresponding to this best solution. There can be some other approaches of selecting a solution from the final Pareto optimal front. The minimum, maximum and average values of the solutions generated on the final Pareto front for both the feature selection and classifiers ensemble approaches are reported in Table 1.

5 Features

We use the following set of features for constructing various models based on CRF. These features were generated without using deep domain knowledge and/or domain-specific resources. Hence, these features are general in nature, and could be applied for any other related domains. Short descriptions of the features are also reported in Table 2.

1. Local contexts: These are the words occurring within the context window $w_{i-5}^{i+5} = w_{i-5}, \ldots, w_{i+5}$, where w_i is the current token. This feature captures the local contextual information that helps to identify the potential entities. We use DE to automatically determine the effective local contexts within this given range.
2. Word prefix and suffix: These are the fixed-length character sequences stripped either from the leftmost or rightmost positions of the tokens. We performed experiment with both $n = 4$ (i.e., eight features) and $n = 3$ (i.e., six features). Exact values to these are automatically determined by our algorithm.
3. Word length: Biomedical names are, in general, longer in length. The feature value of the current token w_i is set to true if the length of the token is greater than 5, and false otherwise.
4. Infrequent word: We prepared a list of words that appear less frequently than a predetermined threshold in the training data. Depending upon the size of the dataset the threshold could vary. Here, we consider the words having less than 10 occurrences in the training data to be infrequent. A feature is then defined that fires if w_i occurs in the compiled list. This is based on the observation that more frequently occurring words are rarely the target entities.
5. Part-of-speech (PoS) information Biomedical names belong to the noun categories. PoS information plays a vital role for identifying the biomedical named entities (NEs). We use PoS information within the context of previous one and next one tokens as features. This information was extracted from the GENIA tagger.[8]

[8] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger.

Table 2 Description of features

Name of the feature	Explanation
AllCapital	All characters are in capital letters
InitialCapital	Initial character is capital or not
CapitalInner	Inner characters are capital or not
InitialCapitalThenMix	First character is capital and next characters are mixed type (allowed characters)
AllDigit	All characters are digits or not
RealNumber	Word is real number or not
DigitWithSpecialCharacter	Word contains special characters along with digit or not
InitialDigitThenAlpha	Word with first digit character followed by alphabets or not
DigitInner	Inner characters of a word are digit or not
SpecialChar	Word contains the special characters or not e.g. [dash, dot, quote]
TwoBegConsecutiveWordMatch	Matching two consecutive words with the beginning two tokens of a multiword name
TwoEndConsecutiveWordMatch	Matching two consecutive words with the last two tokens of a multiword name
StopWordMatch	Matching word with the stopword list
WordMatchFirst	Matching word to the first token of a biomedical entity
WordMatchLast	Matching word to the last token of biomedical entity
WordMatchVerbAfterNE	Matching word with the possible list of verbs
WordMatchVerbBeforeNE	Matching word with the possible list of verbs
WordNormalization	Normalizing surface form of words
RomanNumber	Word is a representation of a Roman number
GreekNumber	Word is a Greek number representation
DigitCommaDigit	Digit, digit is a substring of the word
SingleCapital	Word contains only one capital letter
DigitAlphaDigit	Initial letter is digit, intermediate characters are alphabets and the last character is again a digit
AlphaDigitAlpha	Word starts and ends with alphabet and intermediate characters are all digits
WordPreviouslyOccured	Word previously occurred in the in the training data or not
InitialSmallThenMix	Word starting with small letter and then followed by mixed (capital or small) letters
InitialCapitalThenSmall	Word starting with capital letter and followed by small letters
InitialAlphaThenDigit	Word starting with alphabet followed by digits
InitialCapitalsThenDigit	Word with a sequence of capital letters followed by digits
SemanticFeature	Denotes the set of words that appear more frequently in the surrounding context of an entity
ATGCCharacters	Sequence of ATGC Characters
RootWord	Root of the word e.g. ['go' is the root word of 'went']
ContextFeatures	We have considered various contexts within the window size of $[-m, +n]$ where m and n is decided by feature selections technique
PrefixFeature	Prefixes of length up to n features characters where n is decided by feature selections technique
SuffixFeature	Suffixes of length up to n features characters where n is decided by feature selections technique
InfrequentWord	frequency of occurrences the word in the training data is considered
Part-of-speech	Part of speech information of the current word
Chunk	Chunk information of the current word

6. Chunk information: Chunk information helps to determine the boundaries of biomedical NEs. GENIA tagger is used to get the chunk information.
7. Unknown token feature: This feature is defined based on the concept whether the current token is present in the training data or not. The value of this feature is set to 1 if the current token appears in the training data. During the training phase, the feature value is set randomly.
8. Word normalization: This feature indicates how a word shape is mapped to its equivalent class. This feature will group similar names into the same NE class. For a given word, each small character is replaced by 'a', each capitalized character is replaced by 'A' and each digit is replaced by '0'. If the word contains the characters other than the alphabet and digits, we keep it unaltered. For example, the word 'IL-2' is normalized to 'AA-0'.

9. Head noun: It represents the major noun of a name and describes the property of the name. For example, *transcription factor* is the head noun for the NE *NF-kappa B transcription factor*.
10. Verb trigger: Certain verbs indicate the appearance of biomedical names in their neighboring contexts. The token that immediately follows a trigger word is most likely a NE. The words appearing after these kinds of verbs are assigned the feature values 1.
11. Word class feature: This kind of feature helps for grouping similar names into the same class. For a given token, consecutive capital letters, small letters, numbers and non-English characters are converted to "A", "a", "O" and "-", respectively. For example, the word 'IL-2' is converted to 'A–0'.
12. Informative words: Biomedical names contain many common words that are actually not the NEs. For example, words like *and*, *of*, *normal*, etc. appear inside the NEs but these do not help for their identification. To select the most effective words that help to recognize NEs, we calculate NEweight from the training data depending upon the frequencies of words. This indicates how better the word is to identify and/or classify the NE. The NEweight is calculated as follows:

$$\text{NEweight}(w_i) = \frac{\text{Total no. of occurances of } w_i \text{ as part of a NE}}{\text{Total no. of occurances of } w_i \text{ in the training data}} \quad (5)$$

Two parameters, namely *NEweight* and *number of occurrences* are defined for selecting the effective words. The words with frequencies less than 2 are not considered as informative. The feature is defined in line with the prior works reported in Saha et al. (2009).
13. Content words in surrounding contexts: We define this feature to exploit the global contextual information from the entire document. We consider all unigrams in contexts $w_{i-5}^{i+5} = w_{i-5}, \ldots, w_{i+5}$ of w_i (crossing sentence boundaries) for the entire training data. Tokens are converted to lower case; stopwords, numbers, punctuation markers and special symbols, etc., are removed. We define a feature vector of length 10 using the 10 most frequent content words. Given an instance, the feature corresponding to token t is set to 1 if and only if the context w_{i-5}^{i+5} of w_i contains t. Our evaluation shows that it improves the performance for classifying the NEs.
14. Orthographic features: Depending upon the constructions of the wordforms we define a set of orthographic features. With the alphabetic characters and digits we define the binary-valued features that check whether the word starts with a digit and then followed by alphabet(s); contains only the digits; contains all the capitalized characters; starts with a capital letter; starts with a capital letter and then followed by both capitals and small letters, etc. Another set of features check whether the word contains some special characters like (',','-','.',')' and '('). Many of these features help for boundary identification of NEs. We also implemented some features to check whether the word contains the ATGC sequence and stop words.

6 Datasets and experiments

We evaluate our multiobjective DE-based feature selection and ensemble approaches on three different benchmark datasets, namely JNLPBA 2004 shared task,[9] GENETAG[10] and AIMed.[11] The dataset of JNLPBA 2004 shared task is an outcome of the GENIA project.[12] This dataset contains 2,000 abstracts which are manually annotated with 48 classes among which 36 classes are used for the GENIA corpus. It is further simplified by annotating with only five classes, namely *Protein*, *DNA*, *RNA*, *Cell_line* and *Cell_type* Jin-Dong et al. (2004). The test data contains 404 abstracts. To properly denote the boundaries, the datasets were annotated with the IOB2 format, where 'B-XXX' refers to the beginning of a multiword/single-word NE of type 'XXX', 'I-XXX' refers to the intermediate parts of the NE and 'O' refers to the entities outside the NE.

In GENETAG training and test datasets[13] gene mentions are annotated with the 'NEWGENE' class and the overlapping gene mentions are distinguished by another class 'NEWGENE1'. However, we use IOB2 format (as in GENIA corpus) to properly denote the boundaries of gene names, and we replace all the instances of 'NEWGENE1' classes by 'NEWGENE' tags. The training dataset contains 7,500 sentences with 8,881 gene mentions. The average length per protein mention is 2.1 tokens. The test dataset consists of 2,500 sentences with 2,986 gene mentions. The AIMed corpus consists of 225 abstracts that contain 1,987 sentences with 4,075 protein mentions. The average length of protein mention is 1.3 tokens. We also pre-process this data for the IOB2 boundary marking. For constructing CRF-based classifiers, we use

[9] http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm.
[10] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENEATG.tar.gz.
[11] ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/interactions.tar.gz.
[12] http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm.
[13] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENEATG.tar.gz.

CRF++: Yet Another CRF toolkit,[14] a simple, customizable, and open source implementation of CRF for segmenting or labeling sequential data.

The DE parameters are selected after conducting a thorough sensitivity analysis on the development data. The proper choice of parameters in DE, e.g., population size, number of generations, crossover and mutation rates, etc., is crucial to better optimize the algorithm. The combination of different parameter values might yield very different results. A good setting may result in convergence of the algorithm to the best solution within a reasonable time period. In contrast, a poor setting of parameters might cause the algorithm to execute for a very long time before finding a good solution. Sometimes it may so happen that we may not find a good solution at all. Theoretical results indicating the optimal values of the parameters in an evolutionary algorithm have been proved to be very difficult to derive in the past.

Gmperle et al. (2002) shows that keeping the crossover constant (CR) between 0.3 and 0.9 is a good choice. They also mention that the initial amplification factor (F) value should be kept equal to 0.6. In Ali et al. (2009), authors show that DE is used to produce poor results if the value of F is outside the range of 0.4–1.2. A good choice of F is 0.5 as stated in Ali et al. (2009). We have selected the appropriate values of CR and F in the range of 0.2–0.8. Another important issue is to select the appropriate size of population, which is, in general, related to the problem's difficulty. For a more difficult problem, larger population size should be used in order to reliably achieve a good solution. It is also intuitive to spend more resources for DE to solve the larger problems. Thus, in general, larger population size is necessary when the search space grows. In Brest et al. (2011), authors have proposed to use the initial population size of 100 and then developed an approach to reduce the population size using self-adaptive DE algorithm.

We have also varied the generation number G_{Max} but considered the constant population size, NP which is set to 100. The parameter combinations of DE are optimized on the development data. We achieved the best results when the parameters of DE are set as follows: number of generations, $G_{\text{Max}} = 50$, CR (probability of crossover) = 0.5 and F (mutation factor) = 0.5. The optimized vectors generated by DE (utilizing the development data) are finally evaluated on the test data. We report the results with different parameter configurations in Table 3 for the ensemble on the JNLPBA 2004 shared task data. Results show that the following combination attains the best result: $NP = 100$, $CR = 0.5$, $F = 0.5$ and $G_{\text{Max}} = 50$. Results on the GENETAG and AIMed data sets are obtained using these parameter configurations.

All the necessary files including the outputs of all CRF-based classifiers, codes to generate features for training and testing, the codes for the MODE-based feature selection and classifier ensemble are kept at this site[15] for user access.

[14] http://crfpp.sourceforge.net.

Table 3 DE parameters and the corresponding results on test data (JNLPBA 2004 shared task)

No	CR	F	G_{Max}	r	p	f
1	0.8	0.2	40	74.14	78.98	76.48
2	0.6	0.4	60	74.83	78.37	76.56
3	0.5	0.5	50	75.03	78.54	76.75
4	0.4	0.6	80	74.89	78.24	76.53
5	0.2	0.8	100	74.32	78.71	76.45

Here 'No': Experiment number, 'CR': Crossover constant, 'F': Amplification factor, 'G_{Max}': Maximum generation number, 'r': Recall, 'p': Precision,' f': F-measure

To compare with our proposed method we define the following baseline models:

- Baseline 1: We construct this baseline by considering the following set of features: Context of previous two and next two tokens along with all the features listed in Sect. 5.
- Baseline 2: The individual classifiers, generated in the first stage, are combined together into a final system based on the majority voting. Random choice is made in case all the outputs differ.
- Baseline 3: The classifiers selected in the first step are combined with the help of a weighted voting approach. In each classifier, weight is computed based on the F-measure value of the threefold cross-validation on the training data. The final output label is selected based on the highest weighted vote.

6.1 Performance measures

All the classifiers are evaluated in terms of recall, precision and F-measure metrics. Precision is the ratio of the number of correctly found *NE chunks* to the number of found NE chunks. Recall is the ratio of the number of correctly found NE chunks to the number of true NE chunks. A chunk may be constructed either by one or more than one token.

The value of the metric F-measure, which is the weighted harmonic mean of recall and precision, is calculated as below:

$$F_\beta = \frac{(1+\beta^2)(\text{recall} + \text{precision})}{\beta^2 \times \text{precision} + \text{recall}}, \quad \beta = 1.$$

Here, JNLPBA 2004 shared task evaluation script[16] is used to measure recall, precision and F-measure. The script

[15] http://www.iitp.ac.in/index.php/schools-and-centers/engineering/computer-science-a-engineering/people/faculty/dr-sriparna-saha.html.

[16] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html.

Table 4 Training time (optimal features vs. available features)

ME	JNLPBA04	GENETAG	AIMed
OF	920	723	78
AF	1,430	982	104

Here OF: 'optimal features', af: 'available features', me: 'method', results reported in seconds

outputs three sets of F-measures according to the exact, right and left boundary matching. In the right boundary matching only right boundaries of entities are considered without matching left boundaries and vice versa. In case of exact match, both the left as well as right boundaries are considered. For evaluation with the GENETAG dataset we use the same strict matching criterion that was followed in the Biocreative-II shared task evaluation script[17] for the gene mention detection task. If D, NP and G_{Max} represent the length of the chromosome, number of chromosomes in a population and number of generations, respectively, then the running complexity of our DE-based technique is $O(D \times NP \times G_{\text{Max}})$.

For feature selection, the training time of the algorithm is mentioned in Table 4. Experiments were carried out on a Linux server with 24 GB memory, Intel(R) Xean(R) CPU E5540@2.53 GHz and the cache size of 8 MB. The running time is dependent on the size of the training data, number of available features, number of output labels, etc. The testing time is very less and can be considered to be insignificant compared to the training time.

Here, we show how F-measure values change with respect to the number of generations in the proposed feature selection and classifier ensemble techniques. The figures are shown in Figs. 7, 8 and 9 for JNLPBA, GENETAG and AIMed datasets, respectively. From the figures this is evident that performance improves over the generations. The classifier ensemble approach performs on the outputs produced by the feature selection approach. This is the reason why it achieves considerably better results as compared to the feature selection approach.

6.2 Analysis of results

The algorithm performs in two different steps, viz., feature selection and ensemble learning. At first, we extract the features mentioned in Sect. 5 to train and test the CRF classifier for each of the datasets.

The multiobjective DE-based feature selection technique is then applied to determine the most relevant set of features for the CRF-based classifier. We perform feature selection for all the three benchmark datasets. Each of these experiments yields a set of solutions on the final Pareto optimal front. The solutions represent various feature combinations. Some of the classifiers are good with respect to recall and some are good with respect to precision. For each domain, we select eighteen promising classifiers. Out of these, nine are selected based on the high recall while the rest are selected based on the high precision values. Results of these classifiers are shown in Tables 5, 6 and 7 for the JNLPBA, GENETAG and AIMed datasets, respectively. In the second step, we construct an ensemble by combining these classifiers. Overall evaluation results are reported in Table 8.

For the JNLPBA 2004 shared task data, proposed multiobjective DE-based feature selection technique yields the overall recall, precision and F-measure values of 73.05, 77.62 and 75.26 %, respectively. The first baseline which is constructed by including all the features in CRF model yields the recall, precision and F-measure values of 71.46, 75.73 and 73.53 %, respectively. This is clearly an improvement of 1.73 F-measure points. The DE-based ensemble shows the overall recall, precision and F-measure values of 75.03, 78.54 and 76.75 %, respectively. This is an improvement of 1.49 points over the first stage, i.e., feature selection technique only. It also demonstrates the overall performance increments of 2.22, 1.45 and 1.38 F-measure points over the first, second and third baselines, respectively.

For GENETAG, multiobjective DE-based feature selection technique attains the recall, precision and F-measure values of 91.34, 96.32 and 93.77 %, respectively. This is an improvement of 5.26 points F-measure over the first baseline. The proposed two-stage approach demonstrates the recall, precision and F-measure values of 92.08, 96.32 and 94.15 %, respectively. Thus multiobjective DE-based two-stage approach attains the overall performance increments of 5.64, 2.05 and 2.42 F-measure points over the first, second and third baselines, respectively.

For AIMed datasets the feature selection technique shows the recall, precision and F-measure values of 89.69, 91.44 and 90.56 %, respectively. Compared to the first baseline this is an increment of 1.52 points. The MOO-based ensemble obtains the recall, precision and F-measure values of 91.47, 92.35 and 91.91 %, respectively. Thus, the proposed two-stage approach shows the overall performance increments of 2.87, 1.61 and 1.52 F-measure points over the first, second and third baselines, respectively.

It is evident from the experimental results that all the baseline models achieve lower performance compared to the proposed approach. The proposed DE-based feature selection approach shows that with a relatively small set of effective features we can achieve reasonably good accuracy levels. The MOO-based ensemble further improves the performance. Except the first baseline we utilize some of our proposed resources and/or techniques in the other two baselines. Despite that the multiobjective DE-based ensemble seems to perform favorably better compared to the second and third

[17] http://www.biocreative.org/news/biocreative-ii/.

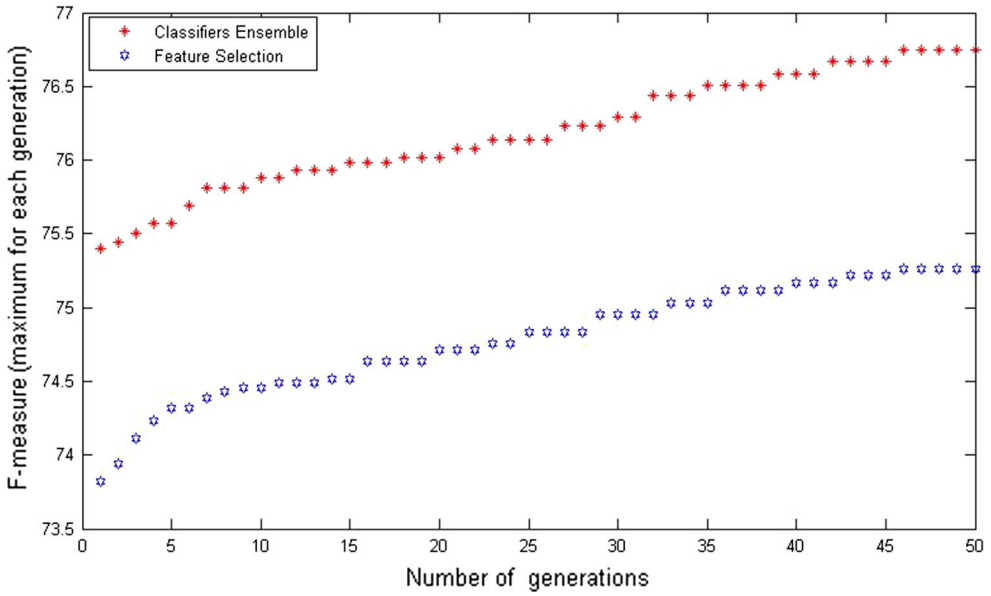

Fig. 7 *F*-measure value vs. number of generations for the JNLPBA04 dataset

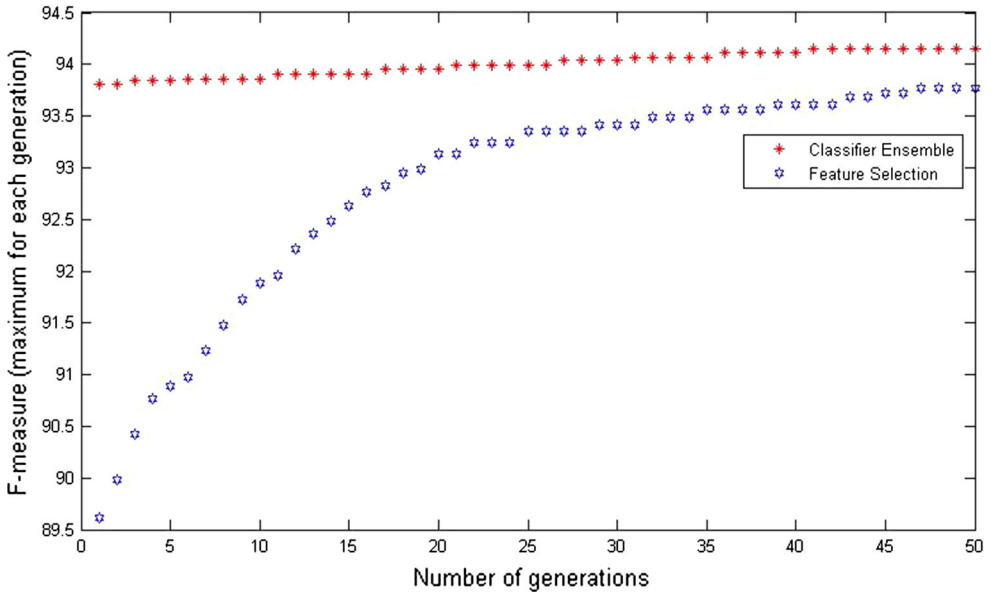

Fig. 8 *F*-measure value vs. number of generations for the GENETAG dataset

baselines. Improvement in the ensemble is due to the fact that rather than blindly combining the outputs we determine the proper voting weights (i.e., near optimal) of the classes in a classifier.

In summary, our proposed approach attains the state-of-the-art performance levels for entity extraction in three benchmark datasets of the biomedical domain.

Statistical analysis of variance, (ANOVA) (Anderson and Scolve 1978), is performed to examine whether the proposed multiobjective DE-based approach really outperforms the best individual classifier selected after the first stage (i.e., after feature selection) and the three baseline ensembles. Our proposed technique is based on DE, a heuristic-based search and optimization technique. The final results provided by DE largely depend on the seed value of the random variables and values of the parameters. For ANOVA analysis, we consider ten different runs (in maximum of the cases results are almost same) of DE. Thereafter, ANOVA analysis is carried out on these outputs. Results of this analysis are shown in Tables 9, 10 and 11 for the different data sets. Results show that the differences in mean recall, precision and *F*-measure values are statistically significant as p value is less than 0.05 in each of

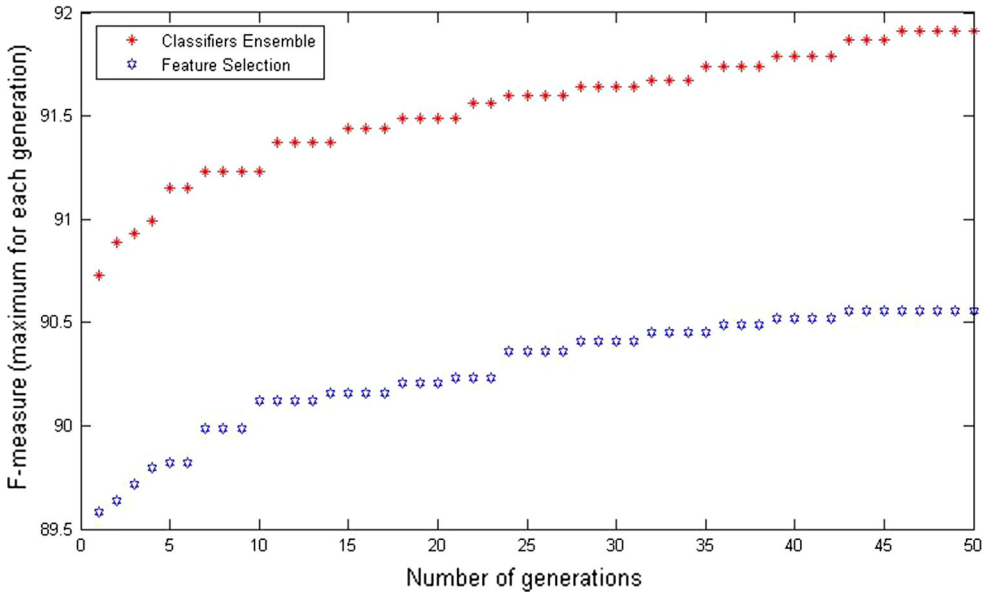

Fig. 9 *F*-measure value vs. number of generations for the AIMed dataset

the cases. Results also reveal that DE-based technique truly performs better than three baseline approaches and the best individual classifier.

7 Comparison with other biomedical NE extraction systems

In this section, we compare the performance of our proposed system with the existing biomedical entity extraction systems that were developed using the same datasets. Please note that we can not directly compare the performance of our system with the others developed using different setups. We did not make use of any deep domain knowledge and/or external resources. In our experiment, we use only PoS and chunk (or, phrase) as the domain-specific external resources. Therefore, it will not be fair to compare the performance of our system with all the available systems. However, we present the comparative evaluation results in Table 12 not only with the domain-independent systems but also with the systems that incorporate deep domain knowledge and/or external resources.

Literature shows that the best performing system on JNLPBA 2004 shared task is described by Wang et al. (2008). The system reported to have achieved the *F*-measure value of 77.57 % with different learning algorithms and domain-dependent features. GuoDong and Jian (2004) developed a system that achieved the *F*-measure value of 72.55 % with several deep domain-dependent knowledge sources. But the *F*-measure value drops to 64.1 % when the system used only PoS and chunk information as the domain knowledge. A maximum entropy (ME)-based system reported in Park et al. (2004) made use of several lexical knowledge sources extracted from the Medline corpus and obtained 66.91 % *F*-measure value. One of the recent works proposed on Saha et al. (2009) achieved the *F*-measure value of 67.41 % without using any deep domain knowledge.

A CRF-based system (Settles 2004) that was developed with different features such as semantic knowledge and orthographic features obtained the *F*-measure value of 70 %. Another CRF-based system was reported in Finkel et al. (2004) that showed the *F*-measure value of 70.06 % with different features and external resources like gazetteers, surrounding abstracts, web-querying and frequency counts from the BNC corpus. Kim et al. (2005) proposed a model based on CRF and ME that achieved the *F*-measure value of 71.19 %. They post-processed the outputs of machine learning models using the rule-based component.

Our proposed approach attains the average recall, precision and *F*-measure values of 75.03, 78.54 and 76.75 %, respectively. This shows the state-of-the-art performance level not only in comparison to the systems that do not make use of any deep domain knowledge and/or external resources, but also to many of the existing systems that made use of deep domain-dependent knowledge and/or resources. In many cases, baseline systems also attain the good performance levels with respect to some other existing systems, possibly because of the rich feature set that we implemented.

To the best of our knowledge, the best performance on the GENETAG dataset was reported in the BioCreative-II gene mention detection task by Ando (2007). They achieved the overall *F*-measure value of 87.2 %, which is inferior to our proposed system by significant margin. Here, the author

Table 5 Evaluation results with various feature combinations for the CRF-based classifiers for JNLPBA04

Cl	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
C_1	−2, 3	X			X				X	X		X	X		4	3		X			
C_2	−2, 3	X	X	X		X			X	X	X	X	X		4	3			X	X	
C_3	−2, 3	X				X		X		X		X			4	3					
C_4	−2, 3	X		X		X			X	X					4	4	X	X		X	X
C_5	−2, 3	X		X		X			X	X		X	X		4	3		X			
C_6	−2, 3	X				X			X	X		X		X	4	3	X	X			
C_7	−2, 3	X		X		X				X	X	X	X	X	4	3					
C_8	−2, 3	X	X	X	X	X				X			X		4	2				X	
C_9	−2, 3	X	X	X		X				X		X			4	3		X			
C_{10}	−2, 3	X	X	X		X					X	X	X		4	3		X			
C_{11}	−2, 3	X		X				X	X	X		X			4	3					X
C_{12}	−2, 3	X				X			X	X		X		X	4	3		X			
C_{13}	−2, 3	X		X		X			X		X	X		X	4	4	X	X		X	X
C_{14}	−2, 3	X						X		X	X	X	X	X	4	3		X		X	X
C_{15}	−2, 3	X			X	X			X	X		X	X		4	2		X			
C_{16}	−2, 3	X		X							X	X	X	X	4	3		X		X	
C_{17}	−2, 3	X		X		X		X		X		X		X	4	2					
C_{18}	−2, 3	X		X						X	X	X	X	X	3	4	X	X		X	X

Cl	V	W	x	Y	Z	a	b	c	d	e	f	g	h	i	j	k	l	*p*	*r*	*F*
C_1			X		X	X	X		X	X			X	X	X	X	X	0.7781	0.7278	0.7521
C_2	X				X		X	X	X		X	X	X	X		X	X	0.7772	0.7288	0.7522
C_3	X		X		X	X	X	X		X		X	X		X	X	X	0.7762	0.7305	0.7526
C_4	X			X	X	X	X					X			X	X	X	0.7746	0.7314	0.7524
C_5			X	X	X	X	X		X	X		X	X	X	X	X	X	0.7764	0.7288	0.7518
C_6	X		X	X	X		X			X		X	X		X	X	X	0.7746	0.7299	0.7516
C_7	X		X		X		X	X				X	X		X	X	X	0.7768	0.7277	0.7514
C_8	X						X	X	X	X			X		X	X	X	0.7774	0.7264	0.7510
C_9	X						X	X	X			X	X	X	X	X	X	0.7766	0.7246	0.7497
C_{10}	X					X	X	X	X			X	X	X	X	X	X	0.7764	0.7260	0.7504
C_{11}					X	X	X	X	X			X			X	X	X	0.7746	0.7297	0.7515
C_{12}			X	X		X	X					X	X	X	X	X	X	0.7758	0.7261	0.7501
C_{13}	X				X		X		X			X	X			X	X	0.7735	0.7296	0.7509
C_{14}	X		X	X			X	X		X		X	X		X	X	X	0.7741	0.7295	0.7511
C_{15}	X		X		X	X			X			X	X	X	X	X	X	0.7746	0.7293	0.7513
C_{16}	X		X	X		X	X	X	X	X			X		X	X	X	0.7754	0.7264	0.7501
C_{17}	X		X			X	X		X		X	X	X	X	X	X	X	0.7739	0.7295	0.7510
C_{18}				X	X	X				X		X	X		X	X	X	0.7735	0.7294	0.7508

Here, the following abbreviations are used: 'A': ContextFeatures, 'B': ContentWordFeature, 'C': InitialCapitalsThenDigit, 'D': InitialAlphaThenDigit, 'E': InitialCapitalThenSmall, 'F': InitialSmallThenMix, 'G': WordPreviouslyOccured, 'H':InfrequentWord, 'I': AlphaDigitAlpha, 'J': DigitAlphaDigit, 'K': SingleCapital, 'L': DigitCommaDigit, 'M': RomanNumber, 'N': GreekNumber, 'O': PrefixFeature, 'P': SuffixFeature, 'Q': WordNormalization, 'R': WordMatchVerbBeforeNE, 'S': WordMatchVerbAfterNE, 'T': WordMatchLast, 'U': WordMatchFirst, 'V': StopWordMatch, 'W': TwoEndConsecutiveWordMatch, 'x': TwoBegConsecutiveWordMatch,'Y': SpecialChar, 'Z': DigitInner, 'a': InitialDigitThenAlpha, 'b': DigitWithSpecialCharacter, 'c': RealNumber, 'd': AllDigit, 'e': InitialCapitalThenMix, 'f': CapitalInner, 'g': AllCapital, 'h': InitialCapital, 'i':ATGCCharacters, 'j': RootWord, 'k': Part-of-Spech Tag, 'l': Chunk Information, 'P', 'C' and 'N': previous, current and next tokens, '$-i$, j': words spanning from the ith left position to the jth right position, current token is at 0th position, 'X': denotes the presence of the corresponding feature, 'r': recall, 'p': precision, 'F': F-measure

Table 6 Evaluation results with various feature combinations for the CRF-based classifiers for GENETAG

Cl	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
C_1	−2, 2			X	X	X						X			3	4	X	X	X	
C_2	−3, 2								X						2	4		X	X	X
C_3	−2, 2			X	X				X						2	4	X		X	X
C_4	−4, 2		X						X	X		X			1	4		X	X	
C_5	−2, 2									X	X	X		X	1	4	X	X	X	
C_6	−3, 2		X			X			X	X		X			3	4		X	X	
C_7	−3, 2		X						X						4	4			X	X
C_8	−2, 3				X						X	X		X	1	4	X		X	
C_9	−3, 2		X	X								X			3	4		X	X	
C_{10}	−3, 2					X			X			X			2	4		X	X	X
C_{11}	−4, 2								X	X					2	4		X	X	X
C_{12}	−2, 2			X	X			X			X	X			2	4	X		X	
C_{13}	−2, 2				X		X			X	X				0	4	X		X	
C_{14}	−2, 2				X			X			X	X			0	4			X	
C_{15}	−4, 2								X						4	4		X	X	X
C_{16}	−2, 2				X			X			X				2	4	X		X	
C_{17}	−3, 2					X			X	X					3	4		X	X	X
C_{18}	−2, 2				X				X			X			1	4	X	X		

Cl	U	V	W	x	Y	Z	a	b	c	d	e	f	g	h	i	k	*p*	*r*	*F*
C_1	X	X	X	X	X			X	X	X					X	X	0.9647	0.9115	0.9374
C_2		X	X	X			X							X		X	0.9676	0.8903	0.9273
C_3	X	X	X	X	X	X	X	X		X	X			X		X	0.9637	0.9129	0.9376
C_4		X	X	X				X			X		X	X	X	X	0.9672	0.8925	0.9284
C_5	X	X	X	X	X	X			X	X	X		X	X	X	X	0.9642	0.9126	0.9377
C_6		X		X			X		X		X		X	X		X	0.9667	0.8944	0.9292
C_7		X	X	X			X		X		X			X	X	X	0.9671	0.8940	0.9291
C_8	X	X	X	X	X	X	X		X	X	X	X		X	X	X	0.9634	0.9135	0.9377
C_9		X	X	X	X		X	X			X		X	X		X	0.9669	0.8922	0.9280
C_{10}		X	X	X			X		X	X				X		X	0.9672	0.8907	0.9274
C_{11}		X	X	X	X		X									X	0.9676	0.8870	0.9256
C_{12}	X	X	X	X		X	X		X	X		X		X	X	X	0.9634	0.9127	0.9374
C_{13}	X	X	X	X	X	X	X			X	X	X				X	0.9642	0.9125	0.9376
C_{14}	X	X	X	X	X	X	X	X			X		X	X	X	X	0.9632	0.9134	0.9377
C_{15}		X	X	X	X		X							X		X	0.9674	0.8887	0.9264
C_{16}	X	X	X	X	X	X	X			X		X		X	X	X	0.9625	0.9132	0.9372
C_{17}		X		X	X		X		X	X						X	0.9672	0.8896	0.9268
C_{18}	X	X	X	X	X		X	X		X	X			X		X	0.9642	0.9121	0.9374

Notations carry the meanings as defined for the JNLPBA datasets

used a semi-supervised learning technique. The system that performed second highest for gene mention detection in BioCreative-II challenge is by Cheng-Ju Kuo and I-Fang Chung (Smith et al. 2014; Kuo et al. 2007). This system is termed as AIIAGMT, and is based on CRF. The system uses a rich feature set, unification of bidirectional parsing models, a dictionary-based filtering post-processing module. It demonstrated the final recall, precision and *F*-measure values of 89.30, 84.49 and 86.83 %, respectively. The third-ranked system of BioCreative-II challenge was developed by Chun-Nan Hsu and Yu-Shi Lin (Smith et al. 2014; Huang et al. 2007). They reported a combined model consisting of two SVMs and one CRF. The improved performance of this system proves that combining multiple complementary mod-

Table 7 Evaluation results with various feature combinations for the CRF-based classifiers for AIMed

Cl	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
C_1	−1, 1	X	X	X	X		X	X	X	X		X	X	X	0	3	X	X	X		X
C_2	−1, 1	X	X			X	X	X		X			X	X	2	4		X		X	X
C_3	−1, 1	X	X			X	X		X	X			X		1	2	X	X		X	X
C_4	−1, 1	X	X		X	X	X	X	X	X	X			X	1	3	X	X	X	X	X
C_5	−1, 1	X	X			X		X		X	X	X			1	2	X			X	X
C_6	−1, 1	X	X		X	X	X	X			X				1	4			X	X	X
C_7	−1, 1	X	X			X	X		X			X		X	1	3	X	X	X	X	X
C_8	−1, 1	X	X		X		X	X			X				1	3	X			X	X
C_9	−1, 1	X	X	X	X			X							1	4		X	X	X	X
C_{10}	−1, 4	X	X	X		X			X				X	X	2	1		X	X	X	X
C_{11}	−1, 1	X				X				X	X			X	1	2	X				X
C_{12}	−1, 1	X	X	X		X	X	X	X			X	X		2	4	X		X	X	X
C_{13}	−1, 1	X	X			X	X	X	X			X		X	1	3	X	X	X	X	X
C_{14}	−1, 1	X		X	X	X				X			X	X	1	3	X	X	X	X	X
C_{15}	−1, 1	X	X			X	X					X		X	0	2	X	X			X
C_{16}	−1, 1	1	X			X	X		X	X	X	X			1	2	X	X		X	
C_{17}	−1, 1	X			X		X	X		X				X	1	3			X	X	X
C_{18}	−1, 1	X			X	X		X		X	X				1	1			X	X	

Cl	V	W	x	Y	Z	a	b	c	d	e	f	g	h	i	j	k	l	*p*	*r*	*F*
C_1		X	X			X	X			X	X	X			X	X	X	0.9127	0.8982	0.9054
C_2	X	X	X		X		X				X	X	X			X	X	0.9135	0.8975	0.9055
C_3	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	0.9144	0.8969	0.9056
C_4	X		X		X			X		X			X			X	X	0.9144	0.8953	0.9048
C_5	X	X	X	X		X	X			X	X		X			X	X	0.9127	0.8975	0.9050
C_6	X		X		X		X			X	X	X			X	X	X	0.9135	0.8968	0.9051
C_7		X	X		X			X		X	X	X				X	X	0.9135	0.8960	0.9047
C_8	X		X	X	X		X			X	X	X	X		X	X	X	0.9118	0.8974	0.9045
C_9	X		X	X	X		X	X		X	X	X	X			X	X	0.9144	0.8945	0.9044
C_{10}	X				X		X			X	X	X				X	X	0.9144	0.8883	0.9012
C_{11}	X	X		X	X	X		X		X	X	X		X		X	X	0.9135	0.8944	0.9039
C_{12}	X		X	X		X	X			X	X	X		X		X	X	0.9118	0.8966	0.9041
C_{13}			X	X	X					X		X			X	X	X	0.9109	0.8965	0.9036
C_{14}	X		X	X	X		X			X	X		X			X	X	0.9135	0.8936	0.9035
C_{15}			X	X	X		X			X	X	X				X	X	0.9100	0.8964	0.9031
C_{16}	X	X	X	X	X	X	X	X		X	X	X		X		X	X	0.9135	0.8929	0.9031
C_{17}	X	X				X	X					X				X	X	0.9135	0.8898	0.9015
C_{18}	X	X	X		X	X	X			X			X	X		X	X	0.9135	0.8882	0.9007

Here, the notations carry the same meanings as that of GENETAG and JNLPBA

els always improves the performance. The organizers of the competition showed (Smith et al. 2014) that a combination of all the submitted systems can achieve an *F*-measure of 90.66 %.

In Li et al. (2012), authors used a classifier ensemble framework to improve the tagging performance. Based on CRF, SVM and ME, they generated six classifiers by varying the feature sets. Finally, these classifiers were combined using a stack-based ensemble. This system achieved the final *F*-measure value of 88.42 % which is better than the highest performing system of BioCreative-II challenge. This is also less compared to our proposed system. In another system by Li et al. (2010), SVM is used along with a reformed lexicon for gene mention detection. Authors have used an ensemble of rule-based post-processing modules, a integrity check module, a boundary check module, an abbreviation resolu-

Table 8 Overall results of the proposed techniques and the baselines

ME	JNLPBA04			GENETAG			AIMed		
	r	*p*	*F*	*r*	*p*	*F*	*r*	*p*	*F*
BIC	73.05	77.62	75.26	91.34	96.32	93.77	89.69	91.44	90.56
B1	71.46	75.73	73.53	81.97	96.17	88.51	88.02	90.09	89.04
B2	73.00	77.76	75.30	88.76	95.69	92.10	89.35	91.27	90.30
B3	73.08	77.81	75.37	90.06	95.55	91.73	89.44	91.35	90.39
TA	75.03	78.54	76.75	92.08	96.32	94.15	91.47	92.35	91.91

Here '*r*': recall, '*p*': precision, '*F*': *F*-measure, ME: method, BIC: best individual classifier obtained through feature selection, B1: Baseline-1, B2: Baseline-2, B3: Baseline-3, TA: proposed two-stage approach

Table 9 Estimated marginal means and pairwise comparisons between the proposed approach (multiobjective differential evolution-based approach) and several other techniques for JNLPBA 2004 dataset

Evaluation criterion	Technique (I)	Technique (J)	Mean diff. (I − J)	Significance value
F-measure	MODE based approach	Individual classifier	0.49 ± 0.013	1.1623e−009
F-measure	MODE based approach	Baseline 1	1.22 ± 0.014	3.3990e−009
F-measure	MODE based approach	Baseline 2	0.45 ± 0.011	8.6386e−010
F-measure	MODE based approach	Baseline 3	0.38 ± 0.009	5.5376e−010

Table 10 Estimated marginal means and pairwise comparisons between the proposed approach (multiobjective differential evolution-based approach) and several other techniques for GENETAG dataset

Evaluation criterion	Technique (I)	Technique (J)	Mean diff. (I − J)	Significance value
F-measure	MODE based approach	Individual classifier	0.38 ± 0.016	1.1762e−009
F-measure	MODE based approach	Baseline 1	5.64 ± 0.011	4.5760e−010
F-measure	MODE based approach	Baseline 2	2.05 ± 0.019	6.8783e−010
F-measure	MODE based approach	Baseline 3	2.42 ± 0.007	2.3544e−009

Table 11 Estimated marginal means and pairwise comparisons between the proposed approach (multiobjective differential evolution-based approach) and several other techniques for AIMed dataset

Evaluation criterion	Technique (I)	Technique (J)	Mean diff. $(I - J)$	Significance value
F-measure	MODE based approach	Individual classifier	1.35 ± 0.014	7.4831e−010
F-measure	MODE based approach	Baseline 1	2.87 ± 0.015	1.6643e−009
F-measure	MODE based approach	Baseline 2	1.61 ± 0.014	3.6215e−009
F-measure	MODE based approach	Baseline 3	1.52 ± 0.009	1.5372e−009

tion module and a name pruning module, to improve the performance further. The lexicon is made of uni-indicating and co-indicating words inside gene mention phrases. The system achieved the recall, precision and *F*-measure values of 85.66, 90.67 and 88.09 %, respectively. The comparative evaluation results are reported in Table 13.

Our current approach attains very high accuracy compared to the other existing systems for the GENETAG datasets. Comparisons suggest that our system achieves state-of-the-art performance with only three domain-dependent features, namely PoS, chunk (or, phrase) and an external NE extractor. We systematically analyze the contribution of each feature, and it reveals the fact that huge performance gain is achieved with the PoS information which was provided with the dataset.

It is to be noted that in the GENETAG training and test datasets, PoS information were provided only for the non-gene proteins. We pre-processed this data and assigned the PoS class, NNP, i.e., proper noun to each of these gene tokens. This PoS information actually plays a crucial role in the overall system performance. Another reason is that we used our in-house NE extractor for obtaining the class label information of the test set while extracting the feature that exploits global contextual information.

Table 12 Comparison with the existing approaches for JNLPBA 2004 shared task dataset

System	Used approach	Domain knowledge/resources	FM
Our proposed system	DE based approach (CRF)	POS, phrase	76.75
Wang et al. (2008)	classifier ensemble (general windows, ME, CRF and SVM)	POS, phrase, common gazetteer, species names, chemical name endings, mineral names	77.57
GuoDong and Jian (2004) Final	HMM, SVM	Name alias, cascaded NEs dictionary, POS, phrase	72.55
Kim et al. (2005)	Two-phase model with ME and CRF	POS, phrase, rule-based component	71.19
Finkel et al. (2004)	CRF	Gazetteers, web-querying, surrounding abstracts, abbreviation handling, BNC corpus, POS	70.06
Settles (2004)	ME	POS, semantic knowledge sources of 17 lexicons	70.00
Saha et al. (2009)	ME	POS, phrase	67.41
Park et al. (2004)	ME	POS, phrase, domain-salient words using WSJ, morphological patterns, collocations from Medline	66.91
Song et al. (2004) Final	SVM, CRF	POS, phrase, Virtual sample	66.28
Song et al. (2004) Base	SVM	POS, phrase	63.85
Ponomareva et al. (2007)	HMM	POS	65.7

Table 13 Comparison with the existing approaches for GENETAG dataset

System	Used approach	Domain knowledge/resources	FM
Our proposed system	DE-based approach (CRF and SVM)	POS, phrase	94.15
BiocreativeCombineScore (Smith et al. 2014)	–	–	90.66
Ando (2007)	ASO Semi-supervised approach	POS, word, character types, etc.	87.2
Cheng-Ju Kuo and I-Fang Chung (Smith et al. 2014; Kuo et al. 2007)	CRF	morphological features	86.83
Chun-Nan Hsu and Yu-Shi Lin (Smith et al. 2014; Huang et al. 2007)	SVM, CRF	POS	86.57
Li et al. (2012)	Hybrid method stack based method	POS, morphological, domain-specific features	88.42
Li et al. (2010)	classifer ensemble method SVM, ME and CRF	POS, morphological features	88.09

The proposed algorithm also shows encouraging performance for the AIMed datasets. To extract the feature that takes into account the global contextual information we used our in-house implementation of a NE extractor. Evaluation on three benchmark datasets that were created following the different annotation guidelines shows that our system achieves quite encouraging performance for all the domains, and therefore this not very domain specific.

Note that results show that our baseline approaches perform better than the state-of-the-art techniques for all the three data sets. This is due to the development of rich feature set. We have developed many new features like head noun, verb trigger and informative word feature. The use of these features help the baseline models to achieve state-of-the-art accuracy.

8 Conclusion

In this paper, we have proposed multiobjective differential evolution-based feature selection and classifier ensemble technique. In the first stage, we developed a MOO-based feature selection technique for a well-known supervised machine learning algorithm, namely CRF. It is used to

determine the relevant set of features for three benchmark datasets in the biomedical domains. We implemented a very rich feature set that itself can achieve very high accuracy. The features were derived without using any deep domain knowledge and/or external resources. The final output of the first stage produces a set of solutions on the Pareto optimal front. Each solution in the Pareto front denotes a particular feature combination. We generated various classifiers based on these feature subsets. Among these, we select some of the promising solutions based on the good recall and precision values. These classifiers are thereafter combined into a single model by a MOO-based ensemble technique. Experiments show the F-measure values of 76.75, 94.15 and 91.91 % for JNLPBA 2004 shared task, GENETAG and AIMed datasets, respectively. Detailed comparisons show that our proposed approach performs favorably better compared to the existing state-of-the-art systems that were developed using the same datasets.

In future, we plan to determine the most relevant parameters of classifiers using multiobjective DE-based technique. We would also like to study the effects of our algorithm on the other datasets. Automatically determining the best one from a pool of classifiers for solving the entity extraction problem in biomedical domain is another important research direction. Selecting the appropriate parameter configurations of DE is another optimization problem.

References

Ali M, Pant M, Abraham A (2009) Simplex differential evolution. Acta Polytechnica Hungarica 6(5):95–115

Anderson TW, Scolve S (1978) Introduction to the statistical analysis of data. Houghton Mifflin, Boston

Ando RK (2007) Biocreative ii gene mention tagging system at ibm watson. In: Proceedings of the second biocreative challenge evaluation workshop, Madrid, Spain, pp 101–103

Bandyopadhyay S, Saha S, Maulik U, Deb K (2008) A simulated annealing based multi-objective optimization algorithm: AMOSA. IEEE Trans Evolut Comput 12(3):269–283

Brest J, Mauec MS (2011) Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Comput 15(11):2157–2174

Dasarathy BV, Sheela BV (1979) Composite classifier system design: concepts and methodology. Proc IEEE 67:708–713

Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1:131–156

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, England

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 6(2):181–197

Dietterich TG (2000) Ensemble methods in machine learning. In: Proceedings of the first international workshop on multiple classifier systems, MCS'00. Springer, London, pp 1–15

Ekbal A, Saha S (2012) Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. IJDAR 15(2):143–166

Ekbal A, Saha S (2010a) Classifier ensemble selection using genetic algorithm for named entity recognition. Res Lang Comput 8(1):73–99

Ekbal A, Saha S (2010b) Weighted vote based classifier ensemble selection using genetic algorithm for named entity recognition. In: 15th International conference on applications of natural language to information systems (NLDB 2010), pp 256–267

Ekbal A, Saha S (2010c) Weighted vote based classifier ensemble selection using genetic algorithm for named entity recognition. In: Proceedings of the natural language processing and information systems, and 15th international conference on applications of natural language to information systems, NLDB'10, pp 256–267

Ekbal A, Saha S (2011a) A multiobjective simulated annealing approach for classifier ensemble: named entity recognition in indian languages as case studies. Expert Syst Appl 38(12):14760–14772

Ekbal A, Saha S (2011b) Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach. ACM Trans Asian Lang Inf Process 10(2):1–37

El-Hefnawy NA (2014) Solving bi-level problems using modified particle swarm optimization algorithm. Int J Artif Intell 12(2):88–101

Finkel J, Dingare S, Nguyen H, Nissim M, Sinclair G, Manning C (2004) Exploiting context for biomedical entity recognition: from syntax to the web. In: Proceedings of the joint workshop on natural language processing in biomedicine and its applications (JNLPBA-2004), pp 88–91

Gmperle R, Mller SD, Koumoutsakos P (2002) A parameter study for differential evolution. In: WSEAS international conference on advances in intelligent systems, fuzzy systems, evolutionary computation, pp 293–298

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, New York

GuoDong Z, Jian S (2004) Exploring deep knowledge resources in biomedical name recognition. In: JNLPBA '04: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, pp 96–99

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Heidl W, Thumfart S, Lughofer E, Eitzinger C, Klement EP (2013) Machine learning based analysis of gender differences in visual inspection decision making. Inf Sci 224:62–76

Huang H, Lin Y, Lin K, Kuo C, Chang Y, Yang B, Chung I, Hsu C (2007) High-recall gene mention recognition by unification of multiple backward parsing models. In: Proceedings of the second biocreative challenge evaluation workshop, Madrid, Spain, pp 109–111

Jin-Dong K, Tomoko O, Tsuruoka Y et al (2004) Introduction to the bio-entity recognition task at jnlpba. In: JNLPBA '04: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications. Association for Computational Linguistics, pp 70–75

Kim S, Yoon J, Park KM, Rim HC (2005) Two-phase biomedical named entity recognition using a hybrid method. In: IJCNLP, pp 646–657

Kuo C, Chang Y, Huang H, Lin K, Yang B, Lin Y, Hsu C, Chung I (2007) Rich feature set, unification of bidirectional parsing and dictionary filtering for high f-score gene mention tagging. In: Proceedings of the second biocreative challenge evaluation workshop, Madrid, Spain, pp 105–107

Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML, pp 282–289

Li L, Fan W, Huang D, Dang Y, Sun J (2012) Boosting performance of gene mention tagging system by hybrid methods. J Biomed Inform 45(1):156–164

Li L, Sun J, Huang D (2010) Boosting performance of gene mention tagging system by classifiers ensemble. In: Natural language processing and knowledge engineering (NLP-KE)

Oliveira LS, Benahmed N, Sabourin R, Bortolozzi F, Suen CY (2001) Feature subset selection using genetic algorithms for handwritten digit recognition. In: Proceedings of 14th Brazilian symposium on computer graphics and image processing, Florianopolis, Oct 2001, IEEE, pp 362–369

Park KM, Kim SH, Rim HC, Hwang YS (2004) Me-based biomedical named entity recognition using lexical knowledge. ACM Trans Asian Lang Inf Process 5:4–21

Ponomareva N, Pla F, Molina A, Rosso P (2007) Biomedical named entity recognition: a poor knowledge hmm-based approach. In: NLDB, pp 382–387

Preitl S, Precup RE (2006) Iterative feedback tuning in fuzzy control systems. Theory and applications. Acta Polytech Hung 3(3):81–96

Saha SK, Sarkar S, Mitra P (2009) Feature selection techniques for maximum entropy based biomedical named entity recognition. J Biomed Inform 42(5):905–911

Settles B (2004) Biomedical named entity recognition using conditional random fields and rich feature sets. In: JNLPBA '04: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications. Association for Computational Linguistics, pp 104–107

Sikdar UK, Ekbal A, Saha S (2012) Differential evolution based feature selection and classifier ensemble for named entity recognition. In: COLING, pp 2475–2490

Smith L, Tanabe L, Ando R, Kuo CJ, Chung IF, Hsu CN, Lin YS, Klinger R, Friedrich C, Ganchev K, Torii M, Liu H, Haddow B, Struble C, Povinelli R, Vlachos A, Baumgartner W, Hunter L, Carpenter B, Tsai R, Dai HJ, Liu F, Chen Y, Sun C, Katrenko S, Adriaans P, Blaschke C, Torres R, Neves M, Nakov P, Divoli A, Lopez MM, Mata J, Wilbur WJ (2008) Overview of biocreative II gene mention recognition. Genome Biol 9(Suppl 2)

Song Y, Kim E, Lee GG, Yi B(2004) Posbiotm-ner in the shared task of bionlp/nlpba 2004. In. In Proceedings of the joint workshop on natural language processing in biomedicine and its applications (JNLPBA-2004)

Storn R, Price K (1997) Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

Victor O, Tiwari A, Roy R (2005) Evolutionary computing in manufacturing industry: an overview of recent applications. Appl Soft Comput 5(3):181–299

Wang H, Zhao T, Tan H, Zhang S (2008) Biomedical named entity recognition based on classifiers ensemble. Int J Comput Sci Appl 5:1–11

Yang J, Honavar VG (1998) Feature subset selection using a genetic algorithm. IEEE Intell Syst 13(2):44–49