Entity Extraction in Biochemical Text using Multiobjective Optimization

Utpal Kumar Sikdar, Asif Ekbal, and Sriparna Saha

Department of Computer Science and Engineering,
Indian Institute of Technology, Patna,
India

{utpal.sikdar,asif,sriparna}@iitp.ac.in

Abstract. In this paper we propose a multiobjective modified differential evolution based feature selection and classifier ensemble approach for biochemical entity extraction. The algorithm performs in two layers. The first layer concerns with determining an appropriate set of features for the task within the framework of a supervised statistical classifier, namely, Conditional Random Field (CRF). This produces a set of solutions, a subset of which is used to construct an ensemble in the second layer. The proposed approach is evaluated for entity extraction in chemical texts, which involves identification of IUPAC and IUPAC-like names and classification of them into some predefined categories. Experiments that were carried out on a benchmark dataset show the recall, precision and F-measure values of 86.15%, 91.29% and 88.64%, respectively.

Keywords. Multiobjective modified differential evolution (MODE), feature selection, ensemble learning, conditional random field (CRF), named entity (NE).

1 Introduction

In recent times information extraction in the biomedical or biochemical domain has drawn significant attention of researchers and practitioners. Nowadays the amount of information available in the web is enormous, but most of these are not properly structured. The significant amount of new information is also being added to it daily, making the size bigger and bigger day after day. New terms, medical terminologies, medicines, etc. are constantly being invented, and therefore, organizing, finding and extracting relevant information from such a huge amount of data pose many challenges. In chemical and/or life science literature, the most important entities are mostly formed by chemical compounds like small signal molecules or other biologically active chemical substances. Past literature shows that there exist many representations and nomenclatures for chemical names like SMILES, InChI and IUPAC. The representations of SMILES and InChI are more flexible than IUPAC and allow direct structure search. However, IUPAC or IUPAC-like names are more frequent in biochemical texts. Finding trivial chemical names is not very complex. This can be easily achieved by developing a dictionary-based approach for entity identification and mapping to the corresponding structures. In contrast, it is quite infeasible to enumerate all the IUPAC or IUPAC-like names. Thus, developing accurate text mining techniques for automatic identification of chemical compounds in texts is of great interest and has a potential in applications of different text processing activities such as predictions of drug-drug/protein-protein interactions, determining relations to adverse reactions of chemical compounds and their associations to toxicological endpoints or the extraction of pathway and metabolic reaction relations. A good entity extraction system can help in semantic search by enabling the search engine to return only those documents containing elements of the entity class.

It is a well-established fact that the performance of any classifier greatly depends on the features of training/testing and the parameters used in the classifier. Feature selection [7, 6], also termed as variable selection, attribute selection or variable subset selection, is a commonly used technique in pattern recognition and machine learning domains. By removing most irrelevant and redundant features from the data, feature selection helps to improve the performance of a classifier. The issue

ISSN 1405-5546
DOI: 10.13053/CyS-18-3-2034

of feature selection can be modeled as an optimization problem. Evolutionary approaches have been effectively used for feature selection in the past for solving many problems, e.g., [3, 4]. In these works, the concepts of single and multiobjective optimization have been used. Classifier ensemble is a technique that is constructed by combining the decisions of many classifiers in order to achieve higher accuracy. Some of the evolutionary approaches for building ensembles have been reported in [2, 4, 1, 3, 8].

In this paper, we propose a multiobjective modified differential evolution based approach for feature selection and classifier ensemble. The strategies used in the modified differential evolution are not exactly similar to that of the standard (or traditional) differential evolution [10]. In particular, the mutation process works differently. We develop the multiobjective optimization (MOO) based feature selection technique by optimizing *recall* and *precision* simultaneously. As a base classifier we make use of Conditional Random Field (CRF) [5]. The algorithm produces a set of solutions on the final Pareto optimal front. None of these solutions dominates other in the objective space. Rather than choosing a unique solution from these, we hypothesize that an ensemble might be more effective if we can effectively combine the classifiers generated from the feature combinations, as represented by the solutions of the final Pareto optimal front.

We develop the MOO based ensemble technique that determines the best weights by which the classifiers are combined. In ensemble construction one of the problems is to find the mechanism to combine the decisions of several classifiers. Existing approaches (e.g., stacking, AdaBoost, bagging, etc.) combine the outputs of all the classifiers by using either majority voting or weighted voting. The weights of votes depend on the error rate/performance of the individual classifiers.

However, in reality, in an ensemble system all the classifiers are not equally efficient in detecting all types of output classes. Thus, weights should be varied depending upon the strength or weakness of the classifiers. The weight should be high for the class for which the corresponding classifier performs well, and low otherwise. Therefore it is crucial to determine the appropriate weights of votes for all the classes in each classifier. The single objective DE based ensemble technique proposed in [8] is based on this hypothesis. In contrast to this work, here we present a method based on the concept of MOO that can optimize more than one objective functions simultaneously. The working principle of MOO is inherently distinct from that of SOO. The MOO algorithm provides a set of alternative solutions, each of which is non-dominated with respect to the other. This paper presents an extension of the work reported in our earlier attempt [9].

The work reported in [9] concerns with the SOO. But the current work deals with the concept of multiobjective optimization (MOO) and solves the issues of feature selection and ensemble learning. As already mentioned, from the algorithmic point of view, MOO has completely different behaviors to SOO. Some of the key advantages of MOO over SOO are (i) the ability to optimize more than one objective function simultaneously and (ii) the ability to generate more than one solution on the Pareto optimal front. Multiobjective optimization provides the user with a set of alternative solutions, and hence s/he can choose a solution depending upon the requirement. Experiments on the benchmark datasets yield the recall, precision and F-measure values of 86.15%, 91.29% and 88.64%, respectively. Comparisons with the existing work show that our proposed approach attains the performance at par the state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 gives an overview of the multiobjective differential evolution. In Section 3 we present our proposed feature selection approach that is based on multiobjective differential evolution. Section 4 describes our approach for ensemble learning based on multiobjective differential evolution. In Section 5 we describe the features that we have used for chemical entity extraction. Section 6 and Section 7 report on the datasets and experiments, respectively. Finally, Section 8 concludes the paper.

2 Overview of Multiobjective Modified Differential Evolution

Differential Evolution (DE) [10] 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 type of chromosomes is called a population, denoted by NP. This set denotes the $|NP|$ number of 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 denotes the total number of parameters of a chromosome.

The optimization function depends on this D number of parameters. The values of Ds are the same for all the chromosomes in a population.

The population size NP is fixed and does not change during the execution of the DE process. There are mainly four operators: initialization, mutation, crossover, and selection.

In the initialization process, all the chromosomes in the first generation of the population are initialized with the real values which cover the entire search space. In the next step, we modify the mutation operator which is different from the traditional mutation operator of DE. We always select the best chromosome from the whole population and add this to the weighted difference between two randomly chosen chromosomes.

In crossover, mutant vector parameters are mixed with the parameters of another predefined vector called the base vector and generate trial vector. For the concept of MOO, we modify the selection operator. Here, we merge the trial vectors with the current population and generate the solutions arranged in ranks using the concept of domination and non-domination. The solutions in the next generation are selected from the previous generation.

All the solutions of the first rank are added first and if it is less than NP then the solutions from the subsequent ranks are included. If the number of solutions of the first rank is more than NP then crowding distance sorting algorithm is applied to select the best NP solutions. The process of selection, crossover, and mutation continues for a fixed number of generations or till a termination condition is satisfied. The pseudo code for the multiobjective modified differential evolution is shown in Algorithm 1.

3 Proposed Approach for Feature Selection

In this section we present a method of feature selection based on multiobjective modified differential evolution (MODE). The feature selection is performed for a popular statistical classifier, namely, Conditional Random Field [5]. Suppose, the D number of available features for a given classifier are denoted by $F_1, \ldots, F_D$. The MOO based feature selection method is then stated as follows: Determine the appropriate subset of features $\mathcal{A}' \subseteq \mathcal{A}$ such that the classifier trained using this subset of features should have optimized some evaluation metrics. Here we optimize two objective functions, namely, *recall* and *precision*.

3.1 Chromosome Representation and Population Initialization

The features are encoded as bit strings called chromosomes. The length of chromosomes is set equal to the number of available features. The bits are randomly initialized to either 0 or 1. The value of 1 in the i^{th} bit position indicates that the corresponding feature participates in constructing the CRF based classifier, and the value of 0 denotes that the feature does not participate. If the size of the population is NP, then all the chromosomes in the population of the first generation are initialized in the above way. An example of chromosome representation is shown in Figure 1.

3.2 Fitness Computation

Here we describe how to compute the objective/fitness function values.

1. Suppose, there are K number of features present in a chromosome (i.e., there are total K number of 1's and $D - K$ number of 0's present in a chromosome where $K < D$).

Algorithm 1 Pseudo Code for Multi-Objective Modified Differential Evolution

```
G=0
Create a random initial population X_{i,G}, ∀i, i = 1, ..., NP
Select best vector, rb from the initial population X_{i,G}, ∀i, i = 1, ..., NP
for G=1 to MAX_GEN do
  for i=1 to NP do
    U_{i,G+1} = X_{i,G}
  end for
  for i=1 to NP do
    Select randomly two different chromosomes r1 and r2
    j_rand = randint(1,D)/generate a random integer value from 1 to D */
    for j=1 to D do
      rnd_j = randfloat(0,1)/generate a random real value belonging to [0,1]*/
      if rnd_j < CR or j=j_rand then
        u_{NP+i,j,G+1} = x_{rb,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
  Select the best vector rb from the next generation population X_{i,G+1}, ∀i, i = 1, ..., NP
end for
```

2. Using these K number of features, the classifier is constructed using CRF.

3. Perform 3-fold cross validation and compute the average recall, precision, and F-measure.

4. Using the search capability of DE based MOO, we optimize recall and precision. These two objective functions are maximized.

3.3 New Mutation Operator

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

$$V_{i,G+1} = X_{rb,G} + F(X_{r1,G} - X_{r2,G}), \quad (1)$$

where rb represents the best chromosome with respect to the F-measure value within the current population and $r1$ and $r2$ are the random indices which belong to $\{1, 2, \ldots, NP\}$. The index value of $r1$ and $r2$ are mutually different and $F > 0$. The randomly chosen $r1$ and $r2$ are different from the running index rb and i, so that NP must be greater or equal to four (three in case when i and rb are the same vectors). The value of F belongs $[0, 1]$. It controls the amplification of the differential variation $(X_{r1,G} - X_{r2,G})$. The $V_{i,G+1}$ is termed as the mutated vector. If each parameter of the mutant vector $V_{i,G+1} \geq 0.5$ then we set the parameter value to 1, otherwise 0. A collection of NP number of mutated vectors is called the mutant population.

3.4 Crossover Operator

To increase the diversity of each target vector $X_{i,G}$; $i = 1, 2, 3, \ldots, NP$, in a population, crossover is needed. This is also called recombination. Here the parameter values of the target vector are mixed with the parameter values of the mutated vector. At the end of this process, for each target vector, a trial vector is generated according to

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

where

$$\begin{aligned} u_{j,i,G+1} &= v_{j,i,G+1} \\ &\quad \text{if } (randb(j) \leq CR) \text{ or } j = rnbr(i) \\ &= x_{j,i,G} \\ &\quad \text{if } (randb(j) > CR) \text{ and } j \neq rnbr(i) \end{aligned}$$

for $j = 1, 2, \ldots, D$,

In the above equation, the value of $randb(j)$ belongs to $[0, 1]$. $randb(j)$ is chosen randomly. CR is the crossover constant which has to be determined by the user. CR can take any value between $[0, 1]$ but in our case we set the value of CR equals to 0.5. $rnbr(i)$ returns a random number belonging to $\{1, 2, \ldots, D\}$ which ensures that the trial vector $U_{i,G+1}$ gets at least one parameter from the mutant vector $U_{i,G+1}$. A collection of NP number of trial vectors is called the trial population.

3.5 New Selection Operator

In the selection process, we merge the trial population with the current population. Thus there are $2 \times NP$ chromosomes. In this process we extract best NP number of chromosomes from $2 \times NP$ chromosomes for the population of the next generation, denoted by $G + 1$. For the concept of domination and non-domination relations, ranked solutions are generated from these $2 \times NP$ solutions. Ranked solutions (starting from the first rank) are added to the population of the next generation until its size becomes NP. If the number of solutions exceeds NP, then we apply crowing distance sorting algorithm to choose the best NP solutions. If the number of solutions is below NP then the solutions from the subsequent rank(s) are included. At the end of this process best NP number of chromosomes are found to be stored in the next generation population.

3.6 Termination Condition

The processes of mutation, crossover (or recombination), fitness computation and selection are executed for a maximum number of generations. In the last generation, the proposed method generates a set of solutions (representing classifiers) with (near)-optimal subset of features. This forms

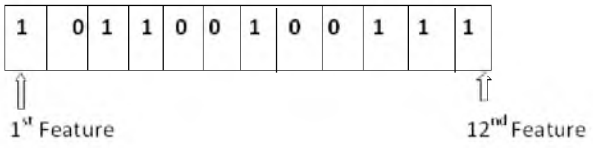

Fig. 1. Chromosome representation for feature selection

the final Pareto optimal front. Some of the solutions are good with respect to *recall* and some are good with respect to *precision*. From the set of all these solutions we select the 14 promising solutions (best 7 with respect to recall and 7 with respect to precision). The classifiers formed with these feature combinations are combined together into a final system (in the second step) using a MODE based classifier ensemble. The key intention was to further improve the performance.

4 Method for Classifier Ensemble

This section presents our method of classifier ensemble that determines the best weight combinations to construct the ensemble. The weighted vote based classifier ensemble problem[3] is stated as follows. Suppose, there are N number of classifiers that are denoted by $C_1, \ldots, C_N$. Let, $\mathcal{A} = \{C_n : n = 1; N\}$ and there are M target classes. The weighted ensemble is then defined as follows.

Determine the voting weights V per classifier which will optimize the fitness function $F(V)$ using the search capability of the modified differential evolution. The size of V is $N \times M$ and it represents a real array. $V(n, m)$ represents the voting weight of the n^{th} classifier for the m^{th} class. These weights can vary from one generation to another. The algorithm ultimately determines the appropriate values of these weights while combining the outputs of the classifiers.

The problem under the MODE based approach can be stated as: For each classifier, find the weights of votes V per classifier such that, $maximize\ \ [F(V)]$, where $F \in \{\text{recall, precision, F-measure}\}$. We optimize F= $\{\text{recall, precision}\}$ as the two objective functions.

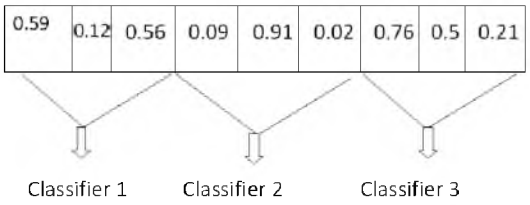

Fig. 2. Problem representation for the ensemble

4.1 Encoding of the Problem

Like in the feature selection problem, ensemble weights are also represented by the chromosomes. The length of this chromosome depends on the number of classifiers and the set of potential target classes. For example, if we have N number of classifiers and M number of target classes, then the chromosome length is $D = N \times M$. As an example, the chromosome representation is shown in Figure 2. This shows an encoding of three classifiers (i.e. $N = 3$) and three classes (i.e., $M = 3$). Therefore, we have 9 ($3 \times 3 = 9$) votes. The chromosome represents the ensemble with the following weights, respectively:

Classifier-1: 0.59, 0.12 and 0.56;

Classifier-2: 0.09, 0.91 and 0.02;

Classifier-3: 0.76, 0.50 and 0.21.

We use real encoding, and all the chromosomes in the entire population are randomly initialized to a real value (r) which belongs to $[0, 1]$. Here,

$$r = \frac{rand()}{RAND\_MAX + 1}.$$

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

4.2 Objective Functions Computation

We perform the following sequence of steps to compute the objective function values.

1. Let us assume that for N number of classifiers, the F-measure values are denoted by F_n, $n = 1 \ldots N$.

2. For each token instance we have M output classes, coming from the M classifiers. The final predicted output for each token is determined based on the weighted voting of these N classifiers' outputs. The weight of a particular class for a particular token t is:

$$f(o_m) = \sum F_n \times C(n, m),$$

$$\forall n = 1 \text{ to } N \text{ and } op(t, n) = o_m$$

Here, $C(n, m)$ corresponds to the n^{th} classifier and m^{th} class; and $op(t, n)$ denotes the predicted class of the nth classifier for the token t. The final output corresponds to the class that receives the maximum weight.

3. Compute the recall and precision values of the ensemble.
4. Repeat the second and third steps 3 times for 3-fold cross validation.
5. Average recall and precision are considered to be the objective functions, and these are optimized using the multiobjective modified differential evolution algorithm.

4.3 Mutation

The mutation process is almost similar to the process followed in feature selection. Here, if the values of the mutant vector parameter violate the boundary constraints then the violating mutant vector parameter values are reflected back from the violated boundary as follows:

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

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

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

4.4 Operators

The values of the other operators for multiobjective DE are determined in the similar way as we did in the feature selection approach.

4.5 Selecting the Best Solution

The MODE based ensemble yields a set of solutions on the final Pareto optimal front. Each solution represents a particular voting weight combination to construct the ensemble. None of the solutions is dominated by the others in the objective space, and therefore all are equally important from the algorithmic point of view. However, at the end we must select one unique solution. For each of the voting weight combinations we construct the ensemble and compute the F-measure values. Finally, we select the particular solution that yields the highest F-measure value.

5 Features for Chemical Entity Extraction

We use the following set of features [9] for the classifier's training and testing. Most of these features were generated without much use of the domain-specific knowledge and/or resources.

1. Surface words and lemma: we use the surface forms of the words and their lemmas as the features.
2. Local contexts: we use the local contexts within the previous three and next three words as features in the model. This was incorporated based on the assumption that contexts carry effective information for the identification of biochemical names.
3. Word prefix and suffix: these denote the fixed length character sequences that are stripped from either the leftmost (for prefix) or the rightmost (for suffix) positions of words. We use the prefixes and suffixes of length up to three characters.
4. Word length: in general, chemical names are longer. More the length of an entity, higher is the chance of being a potential chemical compound name. A binary valued feature is set to high when the number of characters in a given word is above some predefined value; otherwise its value is set to low.

5. Infrequent word: a frequent word has less chance of being a chemical name. A feature is defined that fires for the words that appear more than a predetermined number of times in the training set.

6. Part-of-Speech (PoS) information: syntactic information such as PoS provides useful information about the types of the words. We use the PoS information of the current word and its surrounding tokens as the features. GENIA tagger V2.0.2 [1] was used to extract this information.

7. Chunk information: as already mentioned, chemical compounds are longer in lengths and contain many common words, digits and/or symbols in these. Hence it is important to identify the boundaries (i.e., where it starts and where it ends) of a chemical name. Chunk information that we extracted from the GENIA tagger helps to denote the boundaries.

8. Unknown token feature: this feature checks whether the current token was seen in the training set or not. For the training set this feature was set randomly.

9. Word normalization: word shapes refer to the mapping of each word to their equivalence classes. Here each capitalized character of the word is replaced by 'A', small characters are replaced by 'a' and all consecutive digits are replaced by '0'. For example, 'IL-88' is normalized to 'AA-00'. This feature will group the names having similar structures into the same class.

10. Orthographic features: these binary-valued features are defined based on the contents of the wordforms. For example, initial capital (initial letter is capital or not), all capital (all the letters of the word are capitalized or not), capital in inner (word contains any capital letter inside), initial capital then mix (word starts with a capital letter and then a mixture of capitals and small letters), only digit (word contains only a digit), digit with special character (word contains digits along with special characters), initial digit then alphabetic (word starts with a digit and contains alphabets), etc. The presence of some special characters like (',','-','.',')','(' etc.) is highly indicative that the target word is a potential candidate for being a chemical name. Depending on this orthographic information we defined 24 features.

11. Informative words: the words that frequently appear in the surroundings of chemical names can provide useful indicative clues about their identification and classification. We prepared two lists from the training set by extracting most frequently occurring words that appear in the left and right contexts of the chemical names. Two features are then defined that check whether the target word appears in the respective list or not.

12. Chemical prefix and suffix: the frequent prefixes or suffixes that appear with the chemical names may be effective for detecting IUPAC or IUPAC-like names. We extracted frequently occurring prefixes and suffixes of length 2 from the chemical names present in the training data. Based on these two lists we define two features that fire accordingly.

13. PubChem prefix and suffix: we also make use of the PubChem database [2] and extracted frequent prefixes and suffixes of length 2 from the IUPAC names. A binary valued feature is then defined that fires if and only if any of these inflections matches with the character sequences stripped either from the starting or from the end positions of words.

6 Dataset

There exist various ways to represent biochemical names. One of the most popular ways for a standardized representation is the International Union of Pure and Applied Chemistry (IUPAC). It provides a systematic way of naming conventions that maps their chemical structures. Our experiments are based on the datasets that we obtained from the source [3]. The datasets for training and

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

[2] http://pubchem.ncbi.nlm.nih.gov/

[3] http://www.scai.fraunhofer.de/chem-corpora.html

Table 1. Statistics of datasets: #abstracts (Total number of abstracts), #sentences (Total number of sentences), #tokens (Total number of tokens/words) and #IUPAC (IUPAC and IUPAC like names)

Dataset	#abstracts	#sentences	#tokens	#IUPAC
Training dataset	463	3,700	1,61,591	3,712
Test dataset(Patent)	27	160	4,417	471

Table 2. Overall evaluation results

Methods	recall	precision	F-measure
First baseline	90.22	72.91	80.65
Second baseline	82.34	88.26	85.20
Model 1	83.17	89.22	86.09
Model 2	83.98	90.67	87.19
Our Proposed Method	86.15	91.29	88.64

test were generated from the collections of Medline database and patent documents, respectively. The test dataset contains seven classes, namely, IUPAC(e.g., N-methyl), PARTIUPAC(partial chemical names such as 3H-Testosterone, here "3H" is an IUPAC name), TRIVIAL (trade, common or generic names of compounds such as paracetamol, aspirin, etc.), MODIFIER, SUM (molecular formula such as C9H8O4), ABBREVIATION (abbreviations and acronyms of chemicals compounds and drugs such as DMSO) and FAMILY (chemical names associated to some chemical structure like terpenoids). However, the training dataset has only the instances of IUPAC, PARTIUPAC, and MODIFIER classes. Therefore the test dataset was preprocessed to convert all the other classes except IUPAC, PARTIUPAC, and MODIFIER to the "O" class (denoting other than the chemical names). Statistics of the training and test datasets are presented in Table 1.

7 Experiments

We perform experiments with the training and test datasets that we mentioned in the previous section. We define two baseline models as below:

1. First baseline: this baseline is constructed by training CRF with the following feature combination: context of previous one and next one token along with all the features listed in Section 5.
2. Second baseline: we define this baseline based on the single objective optimization based feature selection technique, reported in [9].

We also compare our proposed method with the following two models.

1. Model-1: this model is built based on the MOO based feature selection technique that makes use of simple DE. The best solution from the final Pareto optimal front is determined based on the F-measure value.
2. Model-2: this model corresponds to the MOO based feature selection technique that makes use of modified DE. The process of selecting the best solution is the same as that of the first model.

The parameters of the proposed algorithm are determined by performing 3-fold cross validation on the training set. The parameters of MODE based feature selection are set as follows: population size = 30, CR (probability of crossover) = 0.5, number of generations = 20 and F (mutation factor) = 0.5. Please note that we execute feature selection algorithm using both modified DE and classical DE. Each of these approaches produces a set of solutions on the final Pareto optimal front. We combine

Table 3. Evaluation results with various feature combinations for the CRF based classifiers. Here, the following abbreviations are used: 'A':ContextFeatures, 'B':InitialCapitalThenSmall, 'D':InitialSmallThenMix, 'E' WordPreviously-Occured,'F':InfrequentWord, 'G': AlphaDigitAlpha, 'H': DigitAlphaDigit, 'I': SingleCapital, 'J': DigitCommaDigit, 'K': RomanNumber, 'L': GreekNumber, 'M': PrefixFeature, 'O': SuffixFeature, 'Q': WordNormalization, 'R': WordMatchVerbBeforeNE, 'S': WordMatchVerbAfterNE, 'T': StopWordMatch,'U': DigitInner, 'V': SpecialChar, 'W': InitialDigitThenAlpha, 'Y': DigitWithSpecialCharacter, 'Z': RealNumber, 'a': AllDigit, 'b': InitialCapitalThenMix, 'c': CapitalInner, 'd': AllCapital, 'e': InitialCapital, 'g': PubChem Prefix and Sufix, 'l': Chemical prefix, 'm': Chemical Sufix, 'q': RootWord, 's': Part-Of-Spech Tag, 't': Chunk Information, 'P', 'C' and 'N': Previous, current and next tokens, '$-i, j$': Words spanning from the i^{th} left position to the j^{th} right position, Current token is at 0^{th} position, 'X': Denotes the presence of the corresponding feature, 'r': recall, 'p': precision, 'F': F-measure

	Cl	A	B	D	E	F	G	H	I	J	K	L	M	O	Q	R	S	T	U	V	W	Y	Z	a	b	c	d	e	g	l	m	q	s	t	p	r	F
C_1	-3,3			X	X		X	X				X	3	3				X		X	X	X	X		X	X	X	X		X	X		X		80.67	87.33	83.87
C_2	-1,3	X		X	X		X	X	X		X		2	2		X				X	X	X	X		X	X	X		X	X	X		X	X	90.63	72.30	80.43
C_3	-3,2	X	X		X		X	X			X		2	1	X			X			X	X		X		X	X	X	X	X	X				90.70	72.28	80.45
C_4	-3,3	X		X			X	X				X	3	1		X			X	X	X	X		X	X	X	X	X	X		X	X			90.70	72.20	80.40
C_5	-2,2	X	X		X		X	X	X		X	X	4	1		X				X	X	X	X		X	X	X		X	X	X			X	90.47	72.43	80.45
C_6	-2,3	X		X	X		X	X	X			X	3	1			X			X		X	X	X	X	X		X		X	X			X	90.69	72.49	80.58
C_7	-2,3		X		X		X	X	X		X		2	3					X	X		X			X	X		X	X	X	X	X	X	X	80.29	87.36	83.68
C_8	-3,1			X			X	X					3	3			X			X	X	X		X		X	X		X	X	X		X		80.79	87.39	83.96
C_9	-3,3		X	X	X	X		X					3	3		X				X	X	X				X		X		X	X			X	80.56	87.29	83.79
C_{10}	-1,2				X		X	X			X		2	3						X		X	X		X	X	X	X	X	X	X	X	X		80.54	87.46	83.86
C_{11}	-2,2		X	X	X		X	X			X		2	4		X		X		X	X		X		X	X	X	X	X	X	X		X	X	80.86	87.55	84.07
C_{12}	-2,1			X	X		X						2	3		X			X	X	X	X	X	X		X		X		X		X		X	80.45	87.34	83.75
C_{13}	-2,3	X	X				X	X	X				2	3	X		X		X	X	X			X		X	X		X	X	X		X		90.56	72.18	80.33
C_{14}	-3,3	X	X	X		X	X	X					3	1	X		X	X			X	X	X		X		X		X	X	X	X	X		90.77	72.60	80.67

these two sets of solutions and select 14 promising solutions from the resultant set. These promising solutions correspond to the classifiers that are generated by training with the feature combinations which yield good recall and precision values. Evaluation results of the classifiers with these feature combinations are shown in Table 3.

The second step of our proposed approach combines all these 14 classifiers based on the MODE based ensemble technique. The parameters are fixed as follows: population size=60; number of generations=300; other operators are same as the feature selection approach. Results of the baselines and three different models are shown in Table 2.

The baseline which is constructed by including all the features in CRF model yield s the recall, precision and F-measure values of 90.22%, 72.91% and 80.65%, respectively. The results show that the system suffers because of many false positives, and this, in turn, affects the precision much. This is clearly evident as precision is much lower compared to recall. This ultimately reduces the overall F-measure value. When we apply SOO based feature selection [9], the precision increases significantly (eliminating false positives), but at the cost of recall. The proposed MODE based feature selection technique shows superior performance compared to the SOO based method. This shows the efficacy of MOO over SOO with an increment of 3.44 percentage F-measure points.

The first model which is developed using the MOO technique that incorporates traditional DE shows the recall, precision and F-measure of 83.17%, 89.22% and 86.09%, respectively. The multiobjective modified DE based feature selection shows further performance improvement over the traditional DE.

The ensemble which is constructed in the second stage by combining the classifiers yields the recall, precision and F-measure values of 86.15%, 91.29% and 88.64%, respectively. An improvement of 1.45 percentage F-measure points over the feature selection method (Model-2) is a clear

evidence that we gain in performance if multiple competing classifiers are effectively combined together.

8 Conclusion

In this paper we present our work on feature selection and classifier ensemble for biochemical entity extraction. Our proposed methods for feature selection and ensemble learning are based on the concept of MOO that incorporates modified version of differential evolution as an optimization algorithm. The traditional DE is modified by changing the mutation operator.

We performed feature selection within the framework of a robust statistical classifier, namely, CRF. The classifier is trained using a diverse feature set. Most of these features were generated without using much domain-specific knowledge and/or resources. The MOO based feature selection was developed by finding the optimized feature set with respect to recall and precision.

The solutions obtained on the final Pareto optimal fronts of both the traditional and modified DE based feature selection approaches were merged. We selected 14 good classifiers from these merged set and combined them together into a single system by a MODE based ensemble technique.

Our experiments on the benchmark datasets show that our proposed approach attains the level of performance which is superior compared to the baseline constructed by training CRF with all the available features. For feature selection, MOO based approach performs better compared to SOO. Our evaluation also suggests that by combining more than one classifier we can achieve better performance.

An immediate extension of the current work is to test the efficacy of the proposed approach for the other benchmark biochemical corpora that can be obtained from other sources, e.g., recently held BioCreative campaigns, etc. We also plan to adapt the proposed approach for other domains in order to get an overall impression about its generalization ability.

References

1. **Ekbal, A. & Saha, S.** (**2010**). Classifier ensemble selection using genetic algorithm for named entity recognition. *Research on Language and Computation*, 8, 73–99.
2. **Ekbal, A. & Saha, S.** (**2010**). Weighted vote based classifier ensemble selection using genetic algorithm for named entity recognition. In *Proceedings of the Natural language processing and information systems*, NLDB'10, pp. 256–267.
3. **Ekbal, A. & Saha, S.** (**2011**). Weighted vote-based classifier ensemble for named entity recognition: A genetic algorithm-based approach. *ACM Trans. Asian Lang. Inf. Process.*, 10(2).
4. **Ekbal, A. & Saha, S.** (**2012**). Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. *IJDAR*, 15(2), 143–166.
5. **Lafferty, J. D., McCallum, A., & Pereira, F. C. N.** (**2001**). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pp. 282–289.
6. **Liu, H. & Motoda, H.** (**1998**). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA.
7. **Liu, H. & Yu, L.** (**2005**). Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowl. and Data Eng.*, 17(4), 491–502. doi:http://dx.doi.org/10.1109/TKDE.2005.66.
8. **Sikdar, U. K., Ekbal, A., & Saha, S.** (**2012**). Differential evolution based feature selection and classifier ensemble for named entity recognition. In *COLING*, pp. 2475–2490.
9. **Sikdar, U. K., Ekbal, A., & Saha, S.** (**2014**). Modified differential evolution for biochemical name recognizer. In *CICLing*, pp. 225–236.
10. **Storn, R. & Price, K.** (**1997**). Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4), 341–359. doi: 10.1023/A:1008202821328.

Utpal Kumar Sikdar is a Senior Research Fellow in the Department of Computer Science and Engineering, IIT Patna, India. He received his Bachelor and Master degrees in Information Technology and Software Engineering from Vidyasagar University

and Jadavapur University in 2004 and 2008, respectively. Prior to joining to his Ph.D. programme he served as a specialist at Tata Elxsi Ltd., India. His research interests include anaphora resolution and information extraction.

Asif Ekbal is an Assistant Professor in the Department of Computer Science and Engineering, IIT Patna. He received his Master and PhD in Computer Science and Engineering from Jadavpur University in 2004 and 2009, respectively. His Master and PhD theses were related to the broad areas of Natural Language Processing. Before joining IITP, he served as postdoctoral research fellow at the University of Trento, Italy and Heidelberg University, Germany. His broad areas of research include Natural Language Processing (NLP), Information Extraction, Bio-text Mining etc. He has authored/co-authored more than 80 technical articles in international journals, book chapters, and conference/workshop proceedings. He received the Best Innovative Project Award from the Indian National Academy of Engineering in the year 2000.

Sriparna Saha is an Assistant Professor in the Department of Computer Science and Engineering, IIT Patna. She received her Master and PhD in Computer Science from Indian Statistical Institute, Kolkata in 2005 and 2011, respectively. Her current research interests include pattern recognition, multiobjective optimization and biomedical information extraction. She is the recipient of the Lt Rashi Roy Memorial Gold Medal from the Indian Statistical Institute for outstanding performance in MTech (computerscience). She is the recipient of the Google India Women in Engineering Award,2008. She received India4EU fellowship of the European Union to work as a Post-doctoral Research Fellow in the University of Trento, Italy from September 2010-January 2011. She is also the recipient of Erasmus Mundus Mobility with Asia (EMMA) fellowship of the European Union to work as a Post-doctoral Research Fellow in the Heidelberg University, Germany from September 2009 to June 2010.

Article received on 18/01/2014; accepted on 01/02/2014.