International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 5(2013) pp. 631-641

Bio-molecular Event Extraction using A GA based Classifier Ensemble Technique

Asif Ekbal[1], Sriparna Saha[2*], Md. Hasanuzzaman[3] and Amit Majumder[4]

[1]Department of Computer Science and Engineering, IIT Patna, India
asif@iitp.ac.in

[2]Department of Computer Science and Engineering, IIT Patna, India
sriparna@iitp.ac.in

[3]Department of Computer Science and Engineering, Jadavpur University, Kolkata
hasanuzzaman.im@gmail.com

[4]Department of Computer Science and Engineering, Academy of Technology, Kolkata
jobamit48@yahoo.co.in

*: Corresponding author

Abstract: [1]**The main goal of Biomedical Natural Language Processing (BioNLP) is to capture biomedical phenomena from textual data by extracting relevant entities, information and relations between biomedical entities (i.e. proteins and genes). In general, in most of the published papers, only binary relations were extracted. In a recent past, the focus is shifted towards extracting more complex relations in the form of bio-molecular events that may include several entities or other relations. In this paper we propose an approach that enables event extraction (detection and classification) of relatively complex bio-molecular events. We approach this problem as a supervised classification problem and use the well-known algorithms, namely Conditional Random Field (CRF) and Support Vector Machine (SVM) as the underlying classifiers. These algorithms make use of statistical and linguistic features those represent various morphological, syntactic and contextual information of the candidate bio-molecular trigger words. Here the outputs of these classifiers are combined using a newly developed genetic ensemble technique. The genetic algorithm based ensemble technique will be able to automatically determine the appropriate weights of votes for each classifier for each output class in order to combine the outputs of different classifiers using weighted voting. Experiments on the BioNLP 2009 shared task datasets yield the overall average recall, precision and F-measure values of 53.56%, 51.47%, and 52.50%, respectively, on development data.**

Keywords: Event extraction; Support vector machine; Conditional random field; Biomedical natural language processing; Detection and classification; Text mining.

I. Introduction

The past history of text mining (*TM*) shows the great success of different evaluation challenges based on carefully curated resources. All these shared tasks have significantly contributed to the progress of their respective fields. This has also been similar for bio-text mining (*bio-TM*). Some of the bio-text mining evaluation challenges include the LLL [1], and BioCreative [2]. The first two shared tasks addressed the issues of bio-information retrieval (*bio-IR*) and bio-Named Entity Recognition (*bio-NER*), respectively. The JNLPBA and BioCreative evaluation campaigns were associated with the bio-information extraction (*bio-IE*). These two addressed the issues of seeking relations between bio-molecules. With the emergence of NER systems with performance capable of supporting practical applications, the recent interest of the bio-TM community is shifting toward IE.

Relations among biomedical entities (i.e. proteins and genes) are important in understanding biomedical phenomena and must be extracted automatically from a large number of published papers. Most researchers in the field of Biomedical Natural Language Processing (BioNLP) have focused on extracting binary relations, including protein–protein interactions (PPIs) such as LLL and BioCreative challenges. Binary relations are not sufficient for capturing biomedical phenomena in detail, and there is a growing need for capturing more detailed and complex relations. For this purpose, two large corpora, BioInfer [3] and GENIA [4], have been proposed.

Similar to previous bio-text mining challenges (e.g., LLL and BioCreative), the BioNLP'09 Shared Task also addressed bio-IE, but it tried to look one step further toward finer-grained IE. The difference in focus is motivated in part by different applications envisioned as being supported by the IE methods. For example, BioCreative aims to support curation of PPI databases such as MINT [5], for a long time one of the primary tasks of bioinformatics. The BioNLP'09 shared task contains simple events and complex events. Whereas the simple events consist of binary relations between proteins and their textual triggers, the complex events consist of multiple relations among proteins, events, and their

[1]All the authors equally contributed for the paper

textual triggers. Bindings can represent events among multiple proteins, and regulations can represent causality relations between proteins and events. These complex events are more informative than simple events, and this information is important in modeling biological systems, such as pathways. The primary goal of BioNLP-09 shared task [6] was aimed to support the development of more detailed and structured databases, e.g. pathway [7] or Gene Ontology Annotation (GOA) [8] databases, which are gaining increasing interest in bioinformatics research in response to recent advances in molecular biology. In [9] a machine learning based biological event extraction system was developed.

In the present paper, we propose a system that enables the extraction of bio-molecular events from the medical literature. The main goal of event extraction is to detect the biomolecular events from the texts and to classify them into nine predefined classes, namely *gene expression*, *transcription*, *protein catabolism*, *phosphorylation*, *localization*, *binding*, *regulation*, *positive regulation* and *negative regulation*. We approach the problem from a supervised machine learning perspective based on Conditional Random Field (CRF) and Support Vector Machine (SVM) those make use of statistical and linguistic features that represent various morphological, syntactic and contextual information of the candidate bio-molecular trigger words. We target the event extraction problem as one-step procedure where event identification and classification are performed together.

Initially, we generate different models based on two powerful learning algorithms, namely Conditional Random Field (CRF) and Support Vector Machine (SVM). These models are constructed by varying the available features and/or feature templates. We identify a very rich and effective feature set that includes variety of features based on orthography, local contextual information and global contexts. We hypothesize that rather than selecting the best-fitting feature set, ensembling several systems where each one is based on either different feature representations or different classification methodologies could be a more effective approach to achieve reasonably high accuracy. But, selection of the appropriate subset of classifiers that could participate in constructing an ensemble remains a difficult problem. It can be noted that all the existing ensemble techniques should have a way of combining the decisions of a set of classifiers. Existing approaches combine the outputs of all classifiers either by using majority voting or by using weighted voting. The weights of votes depend on the error rate/performance of individual classifiers. But none of the existing techniques quantifies the amount of vote for each output class in each classifier. However, in reality, in an ensemble system all the classifiers are not good to detect all types of output classes. Some classifiers are good to detect *transcription* class whereas some are good to detect *binding* class. In the present work, we have used a single objective optimization (SOO) based classifier ensemble technique proposed in [10]. In single objective optimization, we optimize a single classification quality measure (i.e. objective function) such as recall, precision or F-measure at a time. Here, we optimize F-measure which is the harmonic mean of recall and precision both. This optimization technique is based on genetic algorithm (GA) [11] which is a randomized search and optimization techniques guided by the principles of evolution and genetics, having a large amount of implicit parallelism. GA has been successfully applied for solving many real-life problems [12, 13, 14, 15, 16]. The key advantages (or, novelty) of our used approach over the existing ensemble techniques are two-fold, i.e. (i). unlike previous ensembles (like stacking), our GA-based approach does not require to have any technique to select the most suitable subset of classifiers from a set of base classifiers. In contrast the proposed technique in [10] does not discard any classifier during ensemble and (ii). rather than assigning same weights to all the classes in a classifier, like any other existing techniques, it effectively finds the proper weights of all the eligible classes depending upon the prediction confidence.

The best configurations of all the classifiers are obtained using the development data. Due to the non-availability of gold annotations in the BioNLP 2009 shared task test dataset, we perform 3-fold cross validation on the training data to report the final evaluation results. Evaluation results of the GA based approach showed the overall recall, precision and F-measure values of 60.56%, 56.47%, and 58.44%, respectively. Experiments with development set show the overall average recall, precision and F-measure values of 53.56%, 51.47%, and 52.50%, respectively. Results show that the GA based classifier ensemble technique attains performance improvements over best individual classifiers.

II. Related Works

The BioNLP'09 shared task [6] included three subtasks: finding core events (Task 1), finding the secondary arguments (such as location and sites) (Task 2), and recognizing speculation and negation (Task 3). In total, nine potential events were identified for extraction. Among them five events were simple such as *gene expression*, *transcription*, *protein catabolism*, *phosphorylation*, and *localization*. The rest four events, namely *binding*, *regulation*, *positive regulation*, and *negative regulation* were relatively complex. A simple event is an event that includes only a single primary theme protein, and a complex event is an event that includes multiple primary theme and cause arguments. These theme and cause can be either proteins or events. Our proposed system targets a part of Task 1. The goal of Task 1 is to identify events along with their types, textual triggers, and primary theme and cause arguments. Textual triggers are tokens that represent the events. Keeping in mind the complexities and challenges involved in the overall task, in the current work, we tried to address the issues of event extraction, where event triggers were identified from the text and classified into some predefined classes.

III. Proposed Approach for Event Extraction

In this section we describe our proposed approach for event extraction that involves identification of bio-molecular events from the texts and classification of them into some predefined categories of interest. We approach this problem from the supervised machine learning perspectives, namely support vector machine (SVM) [17] and Conditional Random Field (CRF) those make use of statistical and linguistic features that represent various morphological, syntactic and

contextual information of the candidate bio-molecular trigger words. Here, we solve the problem of event extraction in one step by performing event identification and classification together. The training set is highly imbalanced. We filter out those sentences that don't contain any proteins. We use the following set of features (described in Section VI) for both event trigger identification and classification[2]. We use the datasets which were tokenized, stemmed, PoS tagged and NE-tagged, and provided in the CoNLL-X format[3]. We use the McClosky-Charniak parsed outputs [18] [4] which were converted to the Stanford Typed Dependencies format.

A. Support Vector Machine Framework for Event Extraction

In the field of NLP, Support Vector Machines (SVMs) [17] are applied to text categorization, and are reported to have achieved high accuracy without falling into over-fitting even though with a large number of words taken as the features [19, 20]. Suppose, we have a set of training data for a two-class problem: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in R^D$ is a feature vector of the i-th sample in the training data and $y \in \{+1, -1\}$ is the class to which $\mathbf{x}_i$ belongs. In their basic form, a SVM learns a linear hyperplane that separates the set of positive examples from the set of negative examples with *maximal margin* (the margin is defined as the distance of the hyperplane to the nearest of the positive and negative examples). In basic SVMs framework, we try to separate the positive and negative examples by the hyperplane written as:

$$(\mathbf{w}.\mathbf{x}) + b = 0 \quad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}.$$

SVMs find the "optimal" hyperplane (optimal parameter $\overline{w}, b$) which separates the training data into two classes precisely.
The linear separator is defined by two elements: a weight vector $\mathbf{w}$ (with one component for each feature), and a bias b which stands for the distance of the hyperplane to the origin. The classification rule of a SVM is:

$$sgn(f(\mathbf{x}, \mathbf{w}, b)) \tag{1}$$

$$f(\mathbf{x}, \mathbf{w}, b) = <\mathbf{w}.\mathbf{x}> + b \tag{2}$$

being $\mathbf{x}$ the example to be classified. In the linearly separable case, learning the maximal margin hyperplane $(\mathbf{w}, b)$ can be stated as a convex quadratic optimization problem with a unique solution: *minimize* $||\mathbf{w}||$, *subject to the constraints* (one for each training example):

$$y_i(<\mathbf{w}.x_i> + b) \geq 1 \tag{3}$$

The SVM model has an equivalent dual formulation, characterized by a weight vector α and a bias b. In this case, α contains one weight for each training vector, indicating the importance of this vector in the solution. Vectors with non null weights are called *support vectors*. The dual classification rule is:

$$f(\mathbf{x}, \alpha, b) = \sum_{i=1}^{N} y_i \alpha_i <\mathbf{x}_i.\mathbf{x}> + b \tag{4}$$

The α vector can be calculated also as a quadratic optimization problem. Given the optimal α^* vector of the dual quadratic optimization problem, the weight vector $\mathbf{w}^*$ that realizes the maximal margin hyperplane is calculated as:

$$\mathbf{w}^* = \sum_{i=1}^{N} y_i \alpha_i^* \mathbf{x}_i \tag{5}$$

The b^* has also a simple expression in terms of $\mathbf{w}^*$ and the training examples $(\mathbf{x}_i, y_i)_{i=1}^N$.
The advantage of the dual formulation is that efficient learning of non-linear SVM separators, by introducing *kernel functions*. Technically, a *kernel function* calculates a dot product between two vectors that have been (non linearly) mapped into a high dimensional feature space. Since there is no need to perform this mapping explicitly, the training is still feasible although the dimension of the real feature space can be very high or even infinite.
By simply substituting every dot product of $\mathbf{x}_i$ and $\mathbf{x}_j$ in dual form with any *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j)$, SVMs can handle non-linear hypotheses. Among the many kinds of *kernel functions* available, we will focus on the d-th *polynomial kernel*:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + 1)^d$$

Use of d-th polynomial kernel function allows us to build an optimal separating hyperplane which takes into account all combination of features up to d.
Support Vector Machines have advantage over conventional statistical learning algorithms from the following two aspects:

1. SVMs have high generalization performance independent of dimension of feature vectors.
2. SVMs can carry out their learning with all combinations of given features without increasing computational complexity by introducing the *Kernel function*.

We develop our system using SVM [19, 17] which perform classification by constructing an N-dimensional hyperplane that optimally separates data into two categories. We have used YamCha[5] toolkit, an SVM based tool for detecting classes in documents and formulating the event extraction task as a sequential labeling problem. Here, the *pairwise multi-class decision* method and the *polynomial kernel function* are used. We use TinySVM-0.07[6] classifier.

B. Conditional Random Field Framework for NERC

Conditional Random Fields (CRFs) [21] are undirected graphical models, a special case of which corresponds to conditionally trained probabilistic finite state automata. Being conditionally trained, these CRFs can easily incorporate a large number of arbitrary, non-independent features while still having efficient procedures for non-greedy finite-state inference and training.
CRF is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence

[2] In our future work, we would like to investigate different feature sets for identification and classification.

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/tools.shtml

[4] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/tools.shtml

[5] http://chasen-org/ taku/software/yamcha/

[6] http://cl.aist-nara.ac.jp/ taku-ku/software/TinySVM

$s =< s_1, s_2, \ldots, s_T >$ given an observation sequence $o =< o_1, o_2, \ldots, o_T >$ is calculated as:

$$P_\wedge(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

where, $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight λ_k, is to be learned via training. The values of the feature functions may range between $-\infty, \ldots + \infty$, but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_o = \sum_{s} \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

which as in HMMs, can be obtained efficiently by dynamic programming.

To train a CRF, the objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequences:

$$L_\wedge = \sum_{i=1}^{N} \log(P_\wedge(s^{(i)}|o^{(i)})) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2},$$

where $\{< o^{(i)}, s^{(i)} >\}$ is the labeled training data. The second sum corresponds to a zero-mean, σ^2 -variance Gaussian prior over parameters, which facilitates optimization by making the likelihood surface strictly convex. Here, we set parameters λ to maximize the penalized log-likelihood using Limited-memory BFGS [22], a quasi-Newton method that is significantly more efficient, and which results in only minor changes in accuracy due to changes in λ.

When applying CRFs to the event extraction problem, an observation sequence is a token of a sentence or document of text and the state sequence is its corresponding label sequence. In general, CRFs can take any value between $-\infty, \ldots + \infty$, although binary values are traditional. A feature function $f_k(s_{t-1}, s_t, o, t)$ has a value of 0 for most cases and is only set to 1, when s_{t-1}, s_t are certain states and the observation has certain properties. We have used the C^{++} based CRF^{++} package [7], a simple, customizable, and open source implementation of CRF for segmenting or labeling sequential data.

IV. Weighted Vote Based Classifier Ensemble Problem Formulation [10]

Suppose, the N number of available classifiers be denoted by $C_1, \ldots, C_N$. Let, $\mathcal{A} = \{C_i : i = 1; N\}$. Suppose, there are M output classes. The weighted vote based classifier ensemble problem is then stated as follows:

Find the weights of votes V per classifier which will optimize some function $F(V)$. Here, V is an real array of size $N \times M$. $V(i, j)$ denotes the weight of vote of the i^{th} classifier for the j^{th} class. More weight is assigned for that particular class for which the classifier is more confident; whereas the output classes for which the classifier is less confident are given less weight. $V(i, j) \in [0, 1]$ denotes the degree of confidence of the i^{th} classifier for the j^{th} class. These weights are used while combining the outputs of classifiers using weighted voting. Here, F_is are some classification quality measures of the combined weighted vote based classifier. The particular type of problem like NERC has mainly three different kinds of classification quality measures, namely recall, precision and F-measure. Thus, $F \in \{\text{recall}, \text{precision}, \text{F-measure}\}$.

The weighted vote based classifier ensemble problem can be formulated under the single objective optimization (SOO) framework is as below: For each classifier, find the weights of votes V per classifier such that, $maximize$ $[F(V)]$, where $F \in \{\text{recall}, \text{precision}, \text{F-measure}\}$. We choose $F =$ F-measure, which is the harmonic mean of recall and precision both.

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

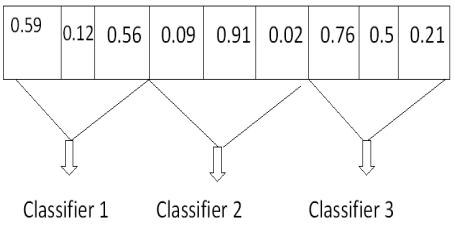

Figure. 1: Chromosome Representation

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

End

Figure. 2: Basic Steps of GA

V. GA Based Classifier Ensemble Technique

In this section, we describe the single objective optimization (SOO) based classifier ensemble approach [10]. It identifies the events from the biomedical texts and classifies them into nine different categories, namely *gene expression*, *transcription*, *protein catabolism*, *phosphorylation*, *localization*, *binding*, *regulation*, *positive regulation* and *negative regulation*. Other than these entities are classified as "Other-than-events", denoted by O. The used ensemble approach is based on GA [11] that closely follows those of the steps as shown in Figure 2.

A. String Representation and Population Initialization

Suppose, there are N available classifiers and O output classes. Then, the length of the chromosome is $N \times O$. Each chromosome encodes the weights of votes for possi-

ble O output classes [8] for each classifier. As an example, the encoding of a particular chromosome is represented in Figure 1, where $N = 3$ and $O = 3$ (i.e., total 9 votes can be possible). The chromosome represents the following ensemble:
The weights of votes for 3 different output classes in classifier 1 are 0.59, 0.12 and 0.56, respectively. Similarly, weights of votes for 3 different output classes are 0.09, 0.91 and 0.02, respectively in classifier 2 and 0.76, 0.5 and 0.21, respectively in classifier 3.
We use real encoding that randomly initializes the entries of each chromosome by a real value (r) between 0 and 1. Here, r is an uniformly distributed random number between 0 and 1. If the population size is P then all the P number of chromosomes of this population are initialized in the above way.

B. Fitness Computation

Initially, the F-measure values of all the classifiers are calculated using 5-fold cross validation on the available training data. Each of these classifiers is built using various representations of the available features and/or feature templates. Thereafter, we execute the following steps to compute the fitness value of each chromosome.

1. Let, the overall F-measure values of the N number of classifiers be F_i, $i = 1 \ldots N$.
2. Initially, the training data is divided equally into 5 parts. Each classifier is trained using 4/5 portions of the training data and evaluated with the remaining 1/5 part (i.e., test dataset). Now, for the ensemble classifier the output class for each token in the 1/5 training data is determined using the weighted voting of these N classifiers' outputs. The weight of the output class provided by the m^{th} classifier is equal to $I(m, i) \times F_m$. Here, $I(m, i)$ is the entry of the chromosome corresponding to m^{th} classifier and i^{th} output class. The combined score of a particular class c_i for a particular token t is:

$$f(c_i) = \sum I(m, i) \times F_m,$$
$$\forall m = 1 : N \ \& \ op(t, m) = c_i$$

   Here, $op(t, m)$ denotes the output class provided by the m^{th} classifier for the token t.

   The class receiving the maximum combined score is selected as the joint decision. Note that in case different boundaries are outputted by the distinct classifiers, the final output is decided by the maximum combined score.
3. The overall F-measure value of the ensemble for the 1/5 training data (i.e., test data) is calculated.
4. Steps 2 and 3 are repeated 5 times to perform 5-fold cross validation.
5. The average F-measure value of this 5-fold cross validation is used as the fitness value of the particular chromosome. This fitness function, fit = F-measure$_{avg}$ is maximized using the search capability of GA.

[8] We also treat the beginning and internals (denoted by BIO labeling scheme) of a multiword NE as the separate classes

C. Selection and Crossover

Roulette wheel selection is used to implement the proportional selection strategy. We use the normal single point crossover [23]. As an example, let the two chromosomes be :
$P1$: 0.24 0.16 0.54 0.87 0.66 0.76 0.01 0.88 0.21
$P2$: 0.12 0.09 0.89 0.71 0.65 0.82 0.69 0.43 0.15
At first a crossover point has to be selected uniformly random between 1 to 9 (length of the chromosome) by generating some random number between 1 and 9. Let the crossover point, here, be 4. Then after crossover, the offsprings are:
$O1$: 0.24 0.16 0.54 0.87 0.65 0.82 0.69 0.43 0.15 (taking the first 4 positions from $P1$ and rest from $P2$)
$O2$: 0.12 0.09 0.89 0.71 0.66 0.76 0.01 0.88 0.21 (taking the first 4 positions from $P1$ and rest from $P2$)
Crossover probability is selected adaptively as in [24]. The expressions for crossover probabilities are computed as follows:
Let f_{max} be the maximum fitness value of the current population, $\overline{f}$ be the average fitness value of the population and $f^{'}$ be the larger of the fitness values of the solutions to be crossed. Then the probability of crossover, μ_c, is calculated as:

$$\mu_c = \begin{cases} k_1 \times \frac{(f_{max} - f^{'})}{(f_{max} - \overline{f})} & \text{if } f^{'} > \overline{f} \\ k_3 & \text{otherwise} \end{cases}$$

Here, as in [24], the values of k_1 and k_3 are kept equal to 1.0. Note that, when f_{max}=$\overline{f}$, then $f^{'} = f_{max}$ and μ_c will be equal to k_3. The aim behind this adaptation is to achieve a trade-off between exploration and exploitation in a different manner. The value of μ_c is increased when the better of the two chromosomes to be crossed is itself quite poor. In contrast, when it is a good solution, μ_c is low so as to reduce the likelihood of disrupting a good solution by crossover.

D. Mutation

Each chromosome undergoes mutation with a probability μ_m. The mutation probability is also selected adaptively for each chromosome as in [24]. The expression for mutation probability, μ_m, is given below:

$$\mu_m = \begin{cases} k_2 \times \frac{(f_{max} - f)}{(f_{max} - \overline{f})} & \text{if } f > \overline{f} \\ k_4 & \text{otherwise} \end{cases}$$

Here, values of k_2 and k_4 are kept equal to 0.5. This adaptive mutation helps GA to come out of local optimum. When GA converges to a local optimum, i.e. when $f_{max} - \overline{f}$ decreases, μ_c and μ_m both will be increased. As a result the GA may come out of this. It will also happen for the global optimum and may result in disruption of the near-optimal solutions. As a result, GA will never converge to the global optimum. The μ_c and μ_m will get lower values for high fitness solutions and get higher values for low fitness solutions. While the high fitness solutions aid in the convergence of GA, the low fitness solutions prevent the GA from getting stuck at a local optimum. The use of elitism will also keep the best solution intact. For a solution with the maximum fitness value, μ_c and μ_m are both zero. The best solution in a population is transferred undisrupted into the next generation. Together

with the selection mechanism, this may lead to an exponential growth of the solution in the population and may cause premature convergence.

Here, each position in a chromosome is mutated with probability μ_m in the following way. The value is replaced with a random variable drawn from a Laplacian distribution, $p(\epsilon) \propto e^{-\frac{|\epsilon-\mu|}{\delta}}$, where the scaling factor δ sets the magnitude of perturbation. Here, μ is the value at the position which is to be perturbed. The scaling factor δ is chosen equal to 0.1. The old value at the position is replaced with the newly generated value. By generating a random variable using Laplacian distribution, there is a non-zero probability of generating any valid position from any other valid position while probability of generating a value near the old value is more.

E. Termination Condition

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

VI. Features for Event Extraction

We identify and use the following set of features for event extraction. All these features are automatically extracted from the training datasets without using any additional domain dependent resources and/or tools.

1. **Context words**: We use preceding and succeeding few words as the features. This feature is used with the observation that contextual information plays an important role in identification of event triggers.

2. **Root words**: Stems of the current and/or the surrounding token(s) are used as the features of the event extraction module. Stems of the words were provided with the evaluation datasets of training, development and test.

3. **Part-of-Speech (PoS) information**: PoS information of the current and/or the surrounding tokens(s) are effective for event trigger identification. PoS labels of the tokens were provided by the organizers with the datasets.

4. **Named Entity (NE) information**: NE information of the current and/or surrounding token(s) are used as the features. NE information was provided with the shared task datasets.

5. **Semantic feature**: This feature is semantically motivated and exploits global context information. This is based on the content words in the surrounding context. We consider all unigrams in contexts $w_{i-3}^{i+3} = w_{i-3} \dots w_{i+3}$ of w_i (crossing sentence boundaries) for the entire training data. We convert tokens to lower case, remove stopwords, numbers, punctuation and special symbols. We define a feature vector of length 10 using the 10 most frequent content words. Given a classification instance, the feature corresponding to token t is set to 1 if and only if the context w_{i-3}^{i+3} of w_i contains t. Evaluation results show that this feature is very effective to improve the performance by a great margin.

6. **Dependency features**: A dependency parse tree captures the semantic predicate-argument dependencies among the words of a sentence. Dependency paths between protein pairs have successfully been used to identify protein interactions. In this work, we use the dependency paths to extract events. We use the McClosky-Charniak parses which are converted to the Stanford Typed Dependencies format and provided to the participants by the shared task organizers. We define a number of features based on the dependency labels of the tokens.

7. **Dependency path from the nearest protein**: Dependency relations of the path from the nearest protein are used as the features. Let us consider a path from "phosphorylation" to "CD40" be "nsubj inhibits acomp binding prep to domain num". Due to the large number of possible words, use of these words on the paths may lead to data sparsity problems, and in turn to poor generalization. Suppose we have a sentence with similar semantics, where the synonym word "prevents" is used instead of "inhibits". If we use the words on the path to represent the path feature, we end up with two different paths for the two sentences that have similar semantics. Therefore, in this work we use only the dependency relation types among the words to represent the paths. For example, the path feature extracted for the (phosphorylation, CD40) negative trigger/participant pair is "nsubj acomp prep to num" and the path feature extracted for the (phosphorylation, TRAF2) positive trigger/participant pair is "prep of".

8. **Boolean valued features**: Two boolean-valued features are defined using the dependency path information. The first feature checks whether the current token's child is a proposition and the chunk of the child includes a protein. The second feature fires if and only if the current token's child is a protein and its dependency label is OBJ

9. **Shortest path**: Distance of the nearest protein from the current token is used as the feature. This is an integer-valued feature that takes the value equal to the number of tokens between the current token and the nearest protein.

10. **Word prefix and suffix**: Fixed length (say, n) word suffixes and prefixes may be helpful to detect event triggers from the text. Actually, these are the fixed length character strings stripped either from the rightmost (for suffix) or from the leftmost (for prefix) positions of the words. If the length of the corresponding word is less than or equal to n-1 then the feature values are not defined and denoted by ND. The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. This feature

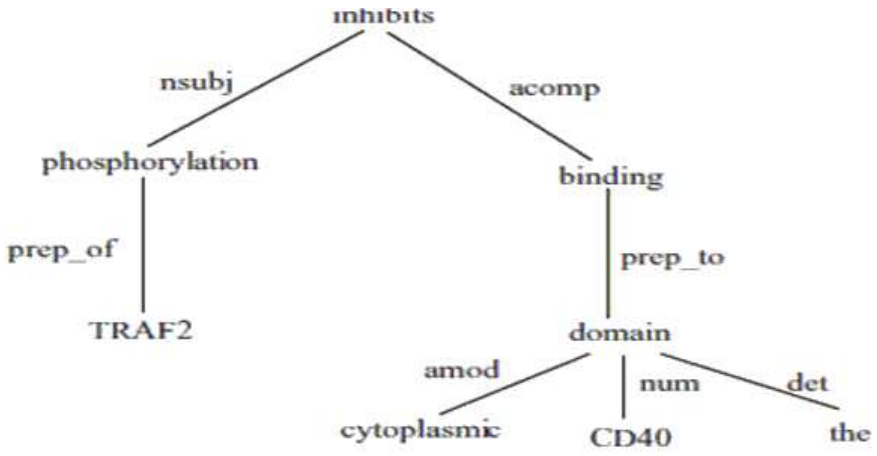

Figure 1: The dependency tree of the sentence *"The phosphorylation of TRAF2 inhibits binding to the CD40 cytoplasmic domain."*

is included with the observation that event triggers share some common suffixes and/or prefixes. In this work, we consider the prefixes and suffixes of length up to four characters.

11. **Named Entities in Context**: We calculate the frequencies of NEs within the various contexts of a sentence. This feature has been defined with the observation that NEs appear most of the times near to the event triggers. Let us suppose that *w* is the current token and L is the size of the sentence in terms of the number of words. We consider various contexts as: context-size= L/K, where K: 1 to 5. Now, considering *w* as centre we define a context window as: context-window-size=2*context-size+1. When the size exceeds the length of the sentence, we added some slots and fill it by the class labels "Other-than-NEs" (denoted by O). For word *w*, a feature vector of length 5 is defined. Depending upon the value of K, the corresponding feature fires. The value is set equal to the number of NEs within the contexts of "context-window-size". For example, for K=1, the entire sentence is considered (i.e., context-size=L). For the first word of the sentence, the context window is equal to more than twice (i.e., 2*context-size+1) of the sentence length. For K=2, the context-size is half of the sentence length. Again, centering the word *w* we define a context of double length by filling the preceding empty slots with O. The feature value is equal to the number of NEs within this window.

VII. Datasets and Experimental Results

In this section we describe datasets used in our task and the experimental results.

A. Datasets

We use the BioNLP-09 shared task datasets. The events were selected from the GENIA ontology based on their significance and the amount of annotated instances in the GENIA corpus. The selected event types all concern protein biology, implying that they take proteins as their theme. The first three event types concern protein metabolism that actually represents protein production and breakdown. *Phosphorylation* represents protein modification event whereas *localization* and *binding* denote fundamental molecular events. *Regulation* and its sub-types, *positive* and *negative* regulations are representative of regulatory events and causal relations. The last five event types are universal but frequently occur on proteins. Detailed biological interpretations of the event types can be found in Gene Ontology (GO) and the GENIA ontology. From a computational point of view, the event types represent different levels of complexity.
Training and development datasets were derived from the publicly available event corpus [25]. The test set was obtained from an unpublished portion of the corpus. But gold annotations are not available for BioNLP 2009 shared task test dataset. We present some statistics of the datasets in Table 1. The shared task organizers made some changes to the original GENIA event corpus. Irrelevant annotations were removed, and some new types of annotation were added to make the event annotation more appropriate. Due to the non-availability of gold annotations in the BioNLP 2009 shared task test datasets, we have reported results on development data and 3-fold cross validation on training data.

B. Experimental Results

We use SVM and CRF for training and testing. In order to properly denote the boundaries of events triggers we use standard IOB notation, where the beginning of a multiword event is tagged as B-Event and the rest of tokens are annotated as I-Event. For example, *interacting receptor-ligand pair* is annotated as *interacting*/B-Event *receptor-ligand*/I-Event *pair*/I-Event in the two-phase approach. The single word token is annotated with B-Event class. For example, *TRAF2 is a . . . which binds*/B-Event *to the CD40 cytoplasmic domain*.

Table 1: Statistics of the datasets

Dataset	#abstracts	#sentences	#words	#events
Training	800	7,449	176,146	8,597
Development	150	1,450	33,937	1,809
Test	260	2,447	57,367	3,182

The system is tuned on the development data, and the results are reported using 3-fold cross validation [9]. The system is evaluated with the standard recall, precision and F-measure metrics. The definitions of recall and precision are given below:

$$\text{recall} = \frac{\text{\# Events correctly identified by the system}}{\text{\# Events in the data set}} \quad (6)$$

$$\text{precision} = \frac{\text{\# Events correctly identified by the system}}{\text{\# Events identified by the system}} \quad (7)$$

From the definitions, it is clear that these two capture two different classification qualities.

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

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

We followed the strict matching criterion, i.e. credit is given if and only if the event types are the same and the event triggers are the same.

A number of CRF and SVM models are generated by varying the available features and/or feature templates.

Evaluation results on the development set showed that the highest performing SVM based model attained the recall, precision and F-measure values of 52.66%, 50.87%, and 51.75%, respectively. Detailed results for individual classes for this classifier are shown in Table 2. Results suggest that apart from the context words, integrating other features within a larger context sometimes decreases the overall performance.

After tuning the system on the development set, we perform 3-fold cross validation on training data to report the final results. Initially, the training dataset is randomly splitted into nearly three equal subsets. Two subsets are used for training and the remaining one subset is withheld for testing. This process is repeated three times to perform 3-fold cross validation.

Evaluation results of 3-fold cross validation for event detection by the best SVM based classifier shows the overall average recall, precision and F-measure values of 57.66%, 55.87%, and 56.75% , respectively.

Results indicate that *gene expression*, *protein_catabolism*, *localization*, *phosphorylation* and *transcription* are relatively easier for both identification and/or classification. In contrast, regulatory events, i.e. *regulation*, *positive regulation* and *negative regulation* are difficult to identify and/or classify. This proves the importance of proper feature selection for event identification and classification both.

A number of CRF models are also generated by varying the available features and/or feature templates. Evaluation results on the development set showed the highest performance with the recall, precision and F-measure values of 50.52%, 48.41%, and 49.54%, respectively. The detailed results of this classifier for each individual output classes are shown in Table 3.

Thereafter, results are reported for 3-fold cross validation on training data. The best CRF based model exhibits the average recall, precision and F-measure values 59.92%, 54.52% and 56.94%, respectively. Results also indicate that *gene expression*, *protein_catabolism*, *localization*, *phosphorylation* and *transcription* are relatively easier for both identification and/or classification. In contrast, regulatory events, i.e. *regulation*, *positive regulation* and *negative regulation* are difficult to identify and/or classify. This proves the importance of proper feature selection for event identification and classification both.

Finally we apply GA based ensembling technique for combining the outputs of several CRF and SVM based classifiers. Results are reported in Table 4 for development data and in Table 5 for 3-fold cross validation on training data. For development data, our approach achieves an improvement of 0.75% F-measure values over the best individual classifier. In case of 3 fold cross validation on the training data, GA based classifier ensemble technique attains the improvements of 1.69% F-measure over the best individual classifier. Results show the superiority of the GA based approach.

VIII. Conclusion and Future Works

In this paper at first we have proposed two supervised machine learning approaches for biological event extraction that involves identification of complex bio-molecular events and classification of them into the predefined nine classes. We have used CRF and SVM those exploit various statistical and linguistic features in the forms of morphological, syntactic and contextual information of the candidate bio-molecular trigger words. We treated event extraction problem as one-step process, and performed event detection and classification together. Three-fold cross validation experiments on the BioNLP 2009 shared task datasets yield the good performance in all our settings. Finally outputs of all these classifiers are combined together using a genetic algorithm based ensemble technique.

Overall evaluation results suggest that there is still the room for further improvement. In this work, we have used a quite similar set of features for event identification and classification both. In our future work, we would like to investigate distinct and more effective set of features for event identification and classification each. We would like to come up with an appropriate feature selection algorithm. In this work, we emphasized on event identification and classification. In our future work, we would like to identify arguments to these events. Argument identification is a more difficult task that requires more sophisticated features and/or classification methods. We also would like to try with other classi-

[9] It is to be noted that due to the unavailability of gold-standard annotations we were unable to evaluate the system with the test dataset

Table 2: Results on development data for event detection and classification (we report percentages) by SVM based approach

Event type	recall	Precision	F-measure
Gene_expression	77.52	52.71	62.98
Transcription	45.79	43.27	44.50
Protein_catabolism	63.61	62.71	63.18
Localization	52.20	58.89	55.38
Binding	53.14	42.52	47.29
Phosphorylation	79.87	68.41	73.66
Regulation	40.95	33.47	36.84
Positive_regulation	42.91	34.67	38.39
Negative_regulation	52.25	39.58	45.07
Overall	52.66	50.87	51.75

Table 3: Results for event detection and classification (in %) for CRF based approach on development data

Event type	recall	precision	F-measure
Gene_expression	77.12	52.64	62.58
Transcription	45.59	43.11	44.31
Protein_catabolism	63.51	62.68	63.09
Localization	52.10	58.83	55.26
Binding	53.04	42.49	47.18
Phosphorylation	79.67	68.34	73.57
Regulation	40.85	33.39	36.75
Positive_regulation	42.78	34.62	38.27
Negative_regulation	52.17	39.54	44.98
Overall	50.52	48.41	49.54

Table 4: Detailed Evaluation results of different approaches on development data (we report percentages)

Approach	recall	precision	F-measure
Best CRF based approach	50.52	48.41	49.54
Best SVM based approach	52.66	50.87	51.75
GA based ensemble	53.56	51.47	52.50

Table 5: Detailed Evaluation results of different approaches on 3 fold cross validation on training data (we report percentages)

Approach	recall	precision	F-measure
Best CRF based approach	59.92	54.52	56.94
Best SVM based approach	57.66	55.87	56.75
GA based ensemble	60.56	56.47	58.44

fication techniques, especially decision trees. In this paper, we have used GA to optimize only one classification quality measure, namely F-measure to determine the best classifier ensemble. But, sometimes only a single measure may not always capture the quality of an ensemble reliably. For a good ensemble, it may often be necessary to optimize more than one classification quality measures simultaneously. Thus it would be good to use multiobjective optimization (MOO) techniques to solve the problem of classifier ensemble for event classification and detection. Future works include the development of novel classifier ensemble techniques using MOO for solving the event detection problem from biomedical texts.

References

[1] C. Nedellec, "Learning language in logic - genic interaction extraction challenge," in *Proceedings of the 4th Learning Language in Logic Workshop (LLL05)*, 7 August 2005.

[2] L. Hirschman, M. Krallinger, and e. Alfonso Valencia, in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 23rd -25th of April 2007.

[3] S. Pyysalo, F. Ginter, J. Heimonen, J. Bjorne, J. Boberg, J. Jarvinen, and T. Salakoski, "BioInfer: a corpus for information extraction in the biomedical domain," *BMC Bioinformatics*, vol. 8, 2007.

[4] L. K. Tanabe, N. Xie, L. H. Thom, W. Matten, and W. J. Wilbur, "Genetag: a tagged corpus for gene/protein named entity recognition," *BMC Bioinformatics*, vol. 6, no. S-1, 2005.

[5] A. Chatr-aryamontri, A. Ceol, L. Montecchi-Palazzi, G. Nardelli, M. V. Schneider, L. Castagnoli, and G. Cesareni, "Mint: the molecular interaction database." *Nucleic Acids Research*, vol. 35 (suppl 1), pp. 572–574, 2007.

[6] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of bionlp'09 shared task on event extraction," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, ser. BioNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 1–9. [Online]. Available: http://dl.acm.org/citation.cfm?id=1572340.1572342

[7] G. D. Bader, M. P. Cary, and C. Sander, "Pathguide: a Pathway Resource List," *Nucleic Acids Research*, vol. 34 (suppl 1), pp. 504–506, 2006.

[8] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler, "The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology." *Nucleic Acids Research*, vol. 32 (suppl 1), pp. 262–266, 2004.

[9] M. Hasanuzzaman, A. Majumder, S. Saha, and A. Ekbal, "Bio-molecular event extraction using support vector machine," in *Proceedings of the 11th International Conference on Hybrid Intelligent Systems (HIS 2011)*. Malacca, Malaysia: IEEE, 2011.

[10] A. Ekbal and S. Saha, "Weighted vote-based classifier ensemble for named entity recognition: A genetic algorithm-based approach," *ACM Trans. Asian Lang. Inf. Process.*, vol. 10, no. 2, p. 9, 2011.

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

[12] K. Matsui and H. Sato, "Neighborhood evaluation in acquiring stock trading strategy using genetic algorithms," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 4, pp. 366–373, 2012.

[13] A. I. S. Nascimento and C. J. A. Bastos-Filho, "Designing cellular networks using particle swarm optimization and genetic algorithms," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 4, pp. 496–505, 2012.

[14] R. Armenise, C. Birtolo, E. Sangianantoni, and L. Troiano, "Optimizing atm cash management by genetic algorithms," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 4, pp. 598–608, 2012.

[15] E. Atilgan and J. Hu, "Improving protein docking using sustainable genetic algorithms," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, p. 248255, 2011.

[16] I. Zelinka and D. Davendra, "Investigation on relations between complex networks and evolutionary algorithm dynamics," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, p. 236247, 2011.

[17] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[18] M. catherine De Marneffe, B. Maccartney, and C. D. Manning, "Generating typed dependency parses from phrase structure parses," in *In LREC 2006*, 2006.

[19] T. Joachims, *Making Large Scale SVM Learning Practical*. Cambridge, MA, USA: MIT Press, 1999, pp. 169–184.

[20] H. Taira and M. Haruno, "Feature Selection in SVM Text Categorization," in *Proceedings of AAAI-99*, 1999.

[21] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML*, 2001, pp. 282–289.

[22] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," in *Proceedings of NAACL '03*, Canada, 2003, pp. 134–141.

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

[24] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms," *IEEE Transactions on Systems, Man and Cybernatics*, vol. 24, no. 4, pp. 656–667, 1994.

[25] J.-D. Kim, T. Ohta, and J. Tsujii, "Corpus annotation for mining biomedical events from literature," *BMC Bioinformatics*, vol. 9, no. 1, p. 10, January 2008, iSSN 1471-2105.

Author Biographies

Asif Ekbal received the M.E. and Ph.D. degrees in computer science from Jadavpur University, Kolkata, India, in 2004 and 2009, respectively. He is currently a Faculty Member in the Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India. He has authored or coauthored more than 70 papers. His current research interests include natural language processing, biomedical information extraction and machine learning.
Dr. Ekbal is the recipient of best Innovative Project Award from the "Indian National Academy of Engineering (INAE)" for the undergraduate project named "GANANAA: A Simple Bangla Programming Language Environment" in 2000. He 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 and from May, 2011-July, 2011. He 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-June 2010.

Sriparna Saha received the M.Tech and Ph.D. degrees in computer science from Indian Statistical Institute Kolkata, Kolkata, India, in 2005 and 2011, respectively. She is currently a Faculty Member in the Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India. She has authored or coauthored more than 60 papers. 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-June 2010.

Md. Hasanuzzaman received the B.Tech degree in computer science from West Bengal University of Technology in the year 2004. He received his M.Tech degrees in information technology from West Bengal University of Technology in the year 2007. He had worked as an Officer-on-Special Duty in West Bengal Industrial Corporation Ltd., a State Govt. Owned Company, from 2007-2011. Currently he is working as a senior research engineer in department of computer science and engineering in Jadavpur University, where he is working towards his Ph.D. His current research interests include natural language processing and biomedical information extraction.

Amit Majumder received the B. Tech degree in computer science and engineering from Kalyani University in the year 2002. He received his ME degree in computer science from Jadavpur University in the year 2004. He is currently an Assistant Professor in computer science and engineering department, Academy Of Technology, Hooghly. His current research interests include natural language processing and biomedical information extraction.