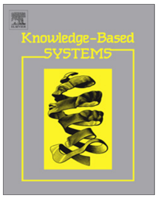
Knowledge-Based Systems 46 (2013) 22–32

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Stacked ensemble coupled with feature selection for biomedical entity extraction

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

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 800 013, Bihar, India

ARTICLE INFO

Article history:
Received 6 July 2012
Received in revised form 26 December 2012
Accepted 10 February 2013
Available online 15 March 2013

Keywords:
Biomedical entity extraction
GA based feature selection
Support Vector Machine (SVM)
Conditional Random Field (CRF)
Stack based Ensemble

ABSTRACT

Entity extraction is one of the most fundamental and important tasks in biomedical information extraction. In this paper we propose a two-stage algorithm for the extraction of biomedical entities in the forms of genes and gene product mentions in text. Several different approaches have emerged but most of these state-of-the-art approaches suggest that individual system may not cover entity representations with arbitrary set of features and cannot achieve best performance. We identify and implement a diverse set of features which are relevant for the identification of biomedical entities and classification of them into some predefined categories. One most important criterion of these features is that these are *identified and selected largely without using any domain knowledge*. In the first stage we use a genetic algorithm (GA) based feature selection technique to determine the most relevant set of features for Support Vector Machine (SVM) and Conditional Random Field (CRF) classifiers. The GA based feature selection algorithm produces best population that can be used to generate different classification models based on CRF and SVM. In the second stage we develop a stacked based ensemble to combine the classifiers selected in the first stage. The proposed approach is evaluated on two benchmark datasets, namely JNLPBA 2004 shared task and GENETAG. The proposed approach yields the overall *F*-measure values of 75.17% and 94.70% for JNLPBA 2004 and GENETAG data sets, respectively.

1. Introduction

Biological research literature is one of the vital databases of knowledge [1]. One important biomedical research database is MEDLINE which has over 14 million abstracts, with 60,000 new abstracts appearing each month. All of these resources are largely annotated manually paying enormous expense. Thus developing some automatic techniques to solve problems such as tokenization, entity extraction, topic classification, word sense disambiguation etc. in the biomedical domain (cf. MEDLINE's Indexing Initiative [2]) is very much essential.

The past history of text mining (*TM*) shows the great success of different evaluation challenges based on carefully curated resources. All these challenges 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 [3] and BioCreative [4]. 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 named entity (NE) extraction systems with performance capable of supporting practical applications, the recent interest of the bio-TM community is shifting toward information extraction (IE).

Entity extraction in the biomedical domain is an important component for many advanced and popular information extraction tasks like automatic extraction of protein–protein interaction information. But the inherent complex structures of biomedical entities make this task more difficult and challenging. Millions of ambiguous genes exist and simultaneously new genes are created. So, finding these genes are not too easy in biomedical domain. Applying information extraction in this domain has been growing research area over years. And for implementing this task, the initial step is to identify gene and protein names in the text, and if required classify them into further sub-classes. In BioCreative-I, the first challenge was carried out in 2003 and the workshop was held in 2004. There were 15 participants in this shared task. The highest *F*-measure for the gene mention detection was 82.2%. The second BioCreative challenge (BioCreative-II)[2] was held in 2006. The challenge addressed the tasks of gene mention detection, gene normalization and protein–protein interaction. The highest accuracy for this task among the participants of this competition is 87.21 *F*-measure for the gene mention detection [5]. The BioCreative organization is motivated by the increasing number of groups working in this field.

* Corresponding authors.
E-mail addresses: asif@iitp.ac.in (A. Ekbal), sriparna@iitp.ac.in (S. Saha).

[1] Authors equally contributed for the paper.

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

0950-7051/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.knosys.2013.02.008

In [6], a named entity (NE) extraction, especially for gene name identification from the biomedical domain, is proposed. A new edit-distance measure is used as a feature to resolve the spelling variant problem. Support Vector Machine (SVM) is used as the underlying machine learning technique. Additionally an expansion method using virtual examples is used to increase the training set size. This helps in improving the recall of the whole system. Another SVM based gene mention identification system is developed in [7]. Several different features and combinations of features, such as n-grams, neighborhood defined by a sliding window, classification results of preceding words, appearance of special characters or digits, or appearance of the word in a dictionary are used. Multiword entity names are gathered in a context-sensitive post-processing step. In another work by Mitsumori et al. [8], a SVM based gene mention detection system is developed which uses some lexical features and a gene/protein name dictionary collected from SWISS-PORT and TrEMBL. Among other machine learning techniques, Maximum Entropy (ME) and Conditional Random Field (CRF) are more popular. A ME-based system for gene mention detection and classification (i.e. entity extraction) is developed in [9], where a rich set of features derived from the training data at multiple levels of granularity while focussing on correctly identifying entity boundaries are used. Additionally several external knowledge sources including full MEDLINE abstracts and web searches are used to further improve the performance. A CRF-based approach is developed in [10] to extract gene names. A diverse feature set containing standard orthographic features combined with expert features in the form of gene and biological term lexicons is used to solve gene mention extraction problem. An ensemble based technique is developed in [11] in which three classifiers, one based on SVM and two others based on discriminative hidden Markov model (HMM), are combined effectively using a simple majority voting strategy. Moreover three post-processing modules, including an abbreviation resolution module, a protein/gene name refinement module and a simple dictionary matching module, are also incorporated into the system to further improve the performance.

The performance of any classification technique depends on the features used to represent training and test patterns. Feature selection [12,13] is the technique of automatically selecting a subset of relevant features for any classifier in order to build the robust learning models. This is also termed as attribute selection/ subset selection etc. Feature selection helps to improve the performance of a classifier. Classifier ensembles have been a fruitful research direction in machine learning in recent years. It is an effective method to increase the generalization accuracy. Ensemble combines the results of many classifiers; thus helps to overcome the possible drawbacks of individual classifiers and produces a more stable result. An important issue in classifier ensemble is that the classifiers should be as much diverse as possible in nature. This can be achieved by using different feature sets or different training sets to homogeneous classifiers, as well as using different classification principles for each of the individual classifiers, i.e. using heterogenous classifiers. An ensemble may be thought of as a supervised learning algorithm because it can be used to predict the outputs of test samples. As ensemble combines the outputs of many classifiers it is more effective than the base classifiers. In many cases it often overcomes the drawback of individual systems.

In this paper we focus on the problem of entity extraction, where gene/gene product names have to be identified and classified according to some shallow semantic categories. We propose a two-stage approach, the first stage of which deals with a genetic algorithm (GA) [14] based feature selection, and in the second stage we propose a stacked based ensemble [15] technique. Stacking [15] is an important classifier ensemble technique which follows a layered architecture. At the very first level, classifiers are trained using the original dataset and each classifier outputs a prediction for each token. Successive layers receive the predictions of the layer immediately preceding it as an input. Finally at the top level, a single classifier, also called meta-classifier, outputs the final prediction. We identify a very rich and effective feature set that includes variety of features based on orthography, local contextual information and global contexts. One most important characteristic of our system is that the *identification and selection of features are mostly done without any deep domain knowledge and/or external resources*. As classification methods we use Conditional Random Field (CRF) and Support Vector Machine (SVM). The GA based feature selection technique is used to select appropriate feature combinations for each of CRF and SVM. Finally it produces a set of solutions on the best population. These solutions represents different feature combinations for CRF and SVM based models. Some of these solutions are best with respect to *recall* and some are good with respect to *precision*. These are used as the base classifiers. In the second step we use CRF as a meta classifier. This takes as the features the predicted values of all the base level classifiers along with the original features.

The proposed system is evaluated on two benchmark datasets, namely GENETAG[3] and JNLPBA 2004 shared task [16]. In GENETAG, the training dataset contains 7500 sentences with 8881 gene mentions. The average length per protein (or, gene) mention is 2.1 tokens. The test dataset consists of 2500 sentences with 2986 gene mentions. Gene names were annotated with three classes, namely NEWGENE, NEWGENE1 and “others”. In order to show that our system is not limited to any particular domain, the proposed approach is evaluated on the JNLPBA 2004 shared task datasets. The training and test datasets contain 2000 and 404 MEDLINE abstracts of the GENIA corpus, respectively. The training set has 18,546 sentences with 492,551 wordforms, whereas test dataset contains 3856 sentences with 101,039 wordforms. The datasets were annotated with five NE classes, namely *protein*, *DNA*, *RNA*, *cell_line* and *cell_type*. The datasets were boundary marked with the well-known IOB2[4] format. For each entity, two different tags (classes) result in 10 classes for the NEs and one additional class for all non-NEs. Accordingly, there are in total 11 potential classes.

Experiments with the GENETAG datasets yield the overall recall, precision and *F*-measure values of 95.12%, 94.29% and 94.70%, respectively. Evaluation results of our proposed system on the JNLPBA-2004 data set show the recall, precision and *F*-measure values of 75.15%, 75.20% and 75.17%, respectively. Experiments suggest that our proposed system achieves performance superior compared to all the individual classifiers as well as two conventional baseline ensembles. Detailed comparisons exhibit that our proposed approach achieves state-of-the-art accuracies for both the domains. The key contributions of our work are (i). use of rich and diverse features that are very effective for entity extraction; (ii) problem specific feature selection for CRF and SVM using a GA based technique; (iii). use of stack based model for classifier ensemble that further improves the performance over the baseline models; and (iv). proposal of a technique that can perform well across various domains.

The rest of the paper is organized as follows. Section 2 describes the diverse set of features that are very effective for entity extraction. Feature selection problem is formulated in Section 3. Section 4 describes the GA based feature selection technique for the specific problem. In Section 5, we describe our proposed stacked based ensemble technique. Detailed experiments along with evaluation scheme and comparisons are reported in Section 6. Finally we conclude in Section 7 with future work road maps.

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

[4] B, I and and O denote the beginning token, intermediate token (s) and outside token of a NE.

2. Features for entity extraction

Feature selection plays an important role for the success of machine learning techniques. We identify a diverse set of features for constructing the various models based on two robust machine learning algorithms, namely CRF and SVM. These features are general in nature and can also be applied for other biomedical domains. Due to the use of variety of features, the individual classifiers achieve very high accuracies.

1. **Context words**: These are the words occurring within the context window $w_{i-3}^{i+3} = w_{i-3} \ldots w_{i+3}$, $w_{i-2}^{i+2} = w_{i-2} \ldots w_{i+2}$ and $w_{i-1}^{i+1} = w_{i-1} \ldots w_{i+1}$, where w_i is the current word. This feature is considered with the observation that surrounding words carry effective information for the identification of NEs. We use GA to automatically determine the value in the range of $w_{i-5}^{i+5} = w_{i-5} \ldots w_{i+5}$.
2. **Word prefix and suffix**: These are the word prefix and suffix character sequences of length up to n. The sequences are stripped from the leftmost (prefix) and rightmost (suffix) positions of the words. We set the feature values to 'undefined' if either the length of w_i is less than or equal to $n - 1$, w_i is a punctuation symbol or if it contains any special symbol or digit. We experiment with $n = 3$ (i.e., 6 features) and 4 (i.e., 8 features) both.
3. **Word length**: We define a binary valued feature that fires if the length of w_i is greater than a pre-defined threshold. Here, the threshold value is set to 5. This feature captures the fact that short words are likely not to be NEs.
4. **Infrequent word**: A list is compiled from the training data by considering the words that appear less frequently than a predetermined threshold. The threshold value depends on the size of the dataset. Here, we consider the words having less than 10 occurrences in the training data. Now, a feature is defined that fires if w_i occurs in the compiled list. This is based on the observation that more frequently occurring words are rarely the NEs.
5. **Part-of-Speech (PoS) information**: PoS information is a critical feature for entity extraction. In this work, we use PoS information of the current and/or the surrounding token(s) as the features. This information is obtained using GENIA tagger[5] V2.0.2, which is a freely available well-known system used to extract PoS information from the biomedical texts. The accuracy of the GENIA tagger is 98.26%. In the GENETAG training and test datasets, PoS information were provided only for the non-gene proteins. We preprocessed this data and assigned the PoS class, NNP, i.e. proper noun to each instance of gene.
6. **Chunk information**: We use GENIA tagger V2.0.2 to get the chunk information. Chunk information (or, shallow parsing features) provide useful evidences about the boundaries of biomedical entities. In the current work, we use chunk information of the current and/or the surrounding token(s).
7. **Dynamic feature**: Dynamic feature denotes the output tags $t_{i-3}t_{i-2}t_{i-1}$, $t_{i-2}t_{i-1}$, t_{i-1} of the word $w_{i-3}w_{i-2}w_{i-1}$, $w_{i-2}w_{i-1}$, w_{i-1} preceding w_i in the sequence w_1^n. This feature is used for SVM model. For CRF, we consider the bigram template that considers the combination of the current and previous output labels.
8. **Unknown token feature**: This is a binary valued feature that checks whether the current token was seen or not in the training corpus. In the training phase, this feature is set randomly.
9. **Word normalization**: We define two different types of features for word normalization. The first type of feature attempts to reduce a word to its stem or root form. This helps to handle the words containing plural forms, verb inflections, hyphen, and alphanumeric letters. The second type of feature indicates how a target word is orthographically constructed. 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' is normalized to 'AA', 'IL-2' is normalized to 'AA-0' and 'IL-88' is also normalized to 'AA-0'.
10. **Head nouns**: Head noun is the major noun or noun phrase of a NE that describes its function or the property. For example, *transcription factor* is the head noun for the NE *NF-kappa B transcription factor*. In comparison to other words in NE, head nouns are more important as these play key role for correct classification of the NE class. In this work, we use only the unigram and bigram head nouns like *receptor*, *protein*, *binding protein* etc. For domain independence, we extract these head nouns from the training data only. These are compiled to generate a list of 912 entries that contain only the most frequently occurring head nouns. Apart from these head nouns, we also consider the unigrams and bigrams extracted from the left ends of the NEs of the training data. A list of 578 entries is created by considering only the most frequent such n-grams. A feature is defined that fires iff the current word or the sequence of words appears in either of these lists.
11. **Verb trigger**: These are the special type of verb (e.g., *binds*, *participates* etc.) that occur preceding to NEs and provide useful information about the NE class. However, for the sake of domain independence, we do not use a predefined list of trigger words. Based on their frequencies of occurrences, these trigger words are automatically extracted from the training data. A feature is then defined that fires iff the current word appears in the list of trigger words.
12. **Word class feature**: Certain kinds of entities, which belong to the same class, are similar to each other. The word class feature is defined as follows: For a given token, capital letters, small letters, numbers and non-English characters are converted to "A", "a", "O" and "-", respectively. Thereafter, the consecutive same characters are squeezed into one character. This feature will group similar names into the same NE class.
13. **Informative words**: In general, biomedical NEs are too long and they contain many common words that are actually not NEs. For example, the function words such as *of*, *and* etc.; nominals such as *active*, *normal* etc. appear in the training data often more frequently but these don't help to recognize NEs. In order to select the most important effective words, we first list all the words which occur inside the multiword NEs. Thereafter digits, numbers and various symbols are removed from this list. For each word (w_i) of this list, a weight is assigned that measures how better the word is to identify and/or classify the NEs. This weight is denoted by *NEweight* (w_i), and calculated as follows:

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

The effective words are finally selected based on the two parameters, namely *NEweight* and *number of occurrences*. The threshold values of these two parameters are selected based on some experiments. The words which have less than two occurrences inside the NEs are not considered as informative. The remaining words are divided into the following classes:

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

- Class 1: This includes the words that occur more than 100 times. Here, we consider those words whose *NEweights* are greater than 0.4.
- Class 2: This includes the words having occurrences $\geqslant 20$ and <100. Here, we set *NEweight* $\geqslant 0.6$.
- Class 3: This class includes the words having occurrences $\geqslant 10$ and <20. Here, we chose *NEweight* $\geqslant 0.75$.
- Class 4: This includes the words having occurrences $\geqslant 5 < 10$. Here, we chose *NEweight* $\geqslant 0.85$.
- Class 5: This includes the words having occurrences <5. Here, we chose *NEweight* $\geqslant 1.00$.

We compile five different lists for the above five classes of informative words. A binary feature vector of length five is defined for each word. If the current word in training (or, test) is found in any particular list then the value of the corresponding feature is set to 1. This feature is a modification to the one used in [17].

14. **Content words in surrounding contexts**: 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 $w_{i-3}^{i+3} = w_{i-3} \ldots w_{i+3}$ 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. For the GENETAG test set we used the NE outputs predicted by the GENIA tagger to compute this feature. In contrast we used the PoS information of test instances (extracted from the GENIA tagger) to compute this feature for GENIA.
15. **Orthographic features**: We define a number of orthographic features depending upon the contents of the wordforms. Several binary features are defined which use capitalization and digit information. These features are: initial capital, all capital, capital in inner, initial capital then mix, only digit, digit with special character, initial digit then alphabetic, digit in inner. The presence of some special characters like (‘,’,‘-’,‘.’,‘)’,‘(’, etc.) is very much helpful to detect NEs, especially in biomedical domain. For example, many biomedical NEs have ‘-’ (hyphen) in their construction. Some of these special characters are also important to detect boundaries of NEs. We also use the features that check the presence of ATGC sequence and stop words. The complete list of orthographic features is shown in Table 1.

3. Problem formulation for feature selection

In general, feature selection problem is formulated under the single objective optimization. It is stated as follows: Given a set of features Ω and a classification quality measure *P*, determine the feature subset F^* such that:

$$P(F^*) = \max_{F \in \Omega} P(F)$$

In general the search space for this type of problems is 2^d, where *d* is the total number of possible features. Thus, exhaustive search strategies can not be applied in this case. Some heuristics based techniques like GA [14] can be used to search for the appropriate feature combination.

Table 1
Orthographic features.

Feature	Example	Feature	Example
InitCap	Src	AllCaps	EBNA, LMP
InCap	mAb	CapMixAlpha	NFkappaB, EpoR
DigitOnly	1, 123	DigitSpecial	12–3
DigitAlpha	2 × NFkappaB, 2A	AlphaDigitAlpha	IL23R, EIA
Hyphen	-	CapLowAlpha	Src, Ras, Epo
CapsAndDigits	32Dc13	RomanNumeral	I, II
StopWord	at, in	ATGCSeq	CCGCCC, ATAGAT
AlphaDigit	p50, p65	DigitCommaDigit	1,28
GreekLetter	alpha, beta	LowMixAlpha	mRNA, mAb

3.1. Overview of genetic algorithm

Genetic Algorithms (GAs) [14] are randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. GAs perform search in complex, large and multi-modal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. In GAs, the parameters of the search space are encoded in the form of strings called *chromosomes*. A collection of such strings is called a *population*. Initially, a random population is created, which represents different points in the search space. An *objective* or a *fitness* function is associated with each string that represents the degree of *goodness* of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

4. Method for feature selection

In this section we present our method for automatic feature selection using evolutionary GA. It optimizes a single classification quality measure, namely *F*-measure which is a combination of both recall and precision. It determines the most relevant set of features for CRF and SVM with respect to the problem of entity identification and classification.

4.1. Chromosome representation and population initialization

If the total number of features is *F*, then the length of the chromosome is *F*. As an example, the encoding of a particular chromosome is represented in Fig. 1. Here, *F* = 12 (i.e., total 12 different features are available). The chromosome represents the use of 7 features, i.e., first, third, fourth, seventh, tenth, eleventh and twelfth for constructing the particular classifier. The entries of each chromosome are randomly initialized to either 0 or 1. Here, if the *i*th position of a chromosome is 0 then it represents that *i*th feature does not participate in constructing the classifier. Else, if it is 1 then the *i*th feature participates in constructing the classifier.

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

4.2. Fitness computation

We execute the following steps for fitness computation.

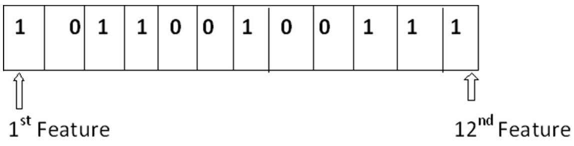


Fig. 1. Chromosome representation for GA based feature selection.

1. Suppose, there are N number of features present in a particular chromosome (i.e., there are total N number of 1's in that chromosome).
2. Construct a CRF/SVM based classifier with only these N features.
3. The training data is divided into 3 parts. The CRF/SVM classifier is trained using 2/3 parts of the training data with the set of features encoded in the corresponding chromosome, and evaluated with the remaining 1/3 part.
4. The overall recall, precision and *F*-measure values of this CRF/SVM classifier for the 1/3 training data are calculated.
5. Steps 2–4 are repeated 3 times to perform 3-fold cross validation. Then, the overall average *F*-measure value of the CRF/SVM classifier is determined from this cross validation experiment.

In case of single objective GA the objective function corresponding to a particular chromosome is: $f_1 = \frac{1}{F\text{-measure}_{avg}}$. The objective is to minimize this objective function.

4.3. Other operators

For single objective GA, normal single point crossover [18] is used. Here, mutation operator is applied to each entry of the chromosome where the entry is randomly replaced by either 0 or 1. Roulette wheel selection is used to implement the proportional selection strategy.

4.4. Generation of several CRF and SVM models

In case of GA, it produces several solutions on the final best population. Each of these solutions provides a feature subset for a CRF/SVM based classifier. Some of these solutions are good with respect to *recall* and some are good with respect to *precision*. All the solutions are equally important from the algorithmic point of view. Thus we generate several CRF and SVM based classifiers by varying the feature combinations, represented in the chromosomes of the final best population. The output of these solutions are then combined using a stacked based ensemble technique.

5. A stacked model for classifier ensemble

In this work we develop a stacked ensemble model for combining the outputs of several classifiers. The first step, i.e. GA based feature selection (c.f. Section 4) provides necessary inputs to the second stage, i.e. ensemble construction. As mentioned in the previous section several base classifiers are generated using the various features, represented in the chromosomes of the final population. All these classifiers are trained with the training data and evaluated on the test data. In the second step we use a meta classifier that is based on CRF. The feature vectors corresponding to the meta-classifier, i.e. of the second level classifier are calculated as follows.

Suppose the number of available features is *F*. The base classifiers are trained using a subset of these set of feature values. Suppose there are M number of base classifiers available; $C_1, C_2,$ - $\ldots, C_M$. A portion of the training dataset is randomly selected to be used as the development set. These base classifiers are first evaluated on this development data to predict the outputs of all instances. Suppose for an instance i of the development data the predictions are $C_1(x), C_2(x), \ldots, C_M(x)$. These predicted classifications are used to form a "meta level training instances" termed as *T*, which is used as a training set to a meta-learning algorithm. The feature vector corresponding to a meta classifier is: $T = \{\text{original attribute values}, C_1(x), C_2(x), \ldots, C_M(x)\}$. This feature vector is extracted for all the tokens of the development data. This will form a training data which is used to train the second level meta classifier. In the test phase, at first base classifiers are used to predict the class labels of each instance. Then these values are used as the features for the meta classifier. The skeleton of this two-stage stacking procedure is shown in Fig. 2. In our work we use CRF as the meta classifier.

6. Evaluation results and discussions

In this section we report the evaluation scheme, our detailed experiments and the necessary comparisons with respect to the current state-of-the-art systems. We use CRF and SVM as the base classifiers in the first stage of our proposed algorithm. Conditional Random Filed (CRF) [19] considers a global exponential model that has the freedom to include arbitrary features and the ability of feature induction to automatically construct the most useful feature combinations. Since, CRFs are log-linear models, and high accuracy may require complex decision boundaries that are non-linear in the space of original features, the expressive power of the models is often increased by adding new features that are conjunctions of the original features. However, it is infeasible to incorporate all possible conjunctions as these might result in overflow of memory as well as overfitting. For constructing CRF based classifiers, we use the C^{++} based CRF^{++} package,[6] a simple, customizable, and open source implementation of CRF for segmenting or labeling sequential data. For CRF training, we use CRF^{++} 0.54 version and set the following parameter values, regularization parameter (a): default setting, i.e. L2; soft-margin parameter (c): trades the balance between overfitting and underfitting (default value); and cut-off threshold for the features (f): uses the features that occur no less than f times in the given training data (set to 1, i.e. all the features that appear at least once in the training dataset are considered). Support Vector Machine (SVM) technique [20,21] takes a strategy that maximizes the margin between the critical samples and the separating hyperplane. In particular, SVMs achieve high generalization even with training data of a very high dimension. Moreover, with the use of *kernel function*, SVMs can handle non-linear feature spaces, and carry out the training considering combinations of more than one feature. For constructing SVM based classifiers, we use YamCha[7] toolkit, an SVM based tool for detecting classes in documents and formulating the NE extraction task as a sequential labeling problem. Here, we use both the *one-vs-rest* and *pairwise multi-class decision* methods, and the *polynomial kernel function*. We use TinySVM-0.07[8] classifier.

We define two different *baseline* ensemble models as below:

- *Majority vote based ensemble*: In this *baseline* model, all the individual classifiers identified by the first stage are combined together into a final system based on the majority voting of the output class labels. If all the outputs differ then anyone is selected randomly.
- *Weighted vote based ensemble*: This is a weighted voting approach. In each classifier, weights are calculated based on the average *F*-measure value of the 3-fold cross validation on the training data.

6.1. Evaluation scheme

All the classifiers are evaluated in terms of recall, precision and *F*-measure. Precision is the ratio of the number of correctly found *NE chunks* (i.e., more than one token) to the number of found NE chunks, and recall is the ratio of the number of correctly found

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

[7] http://chasen-org/taku/software/yamcha/.

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

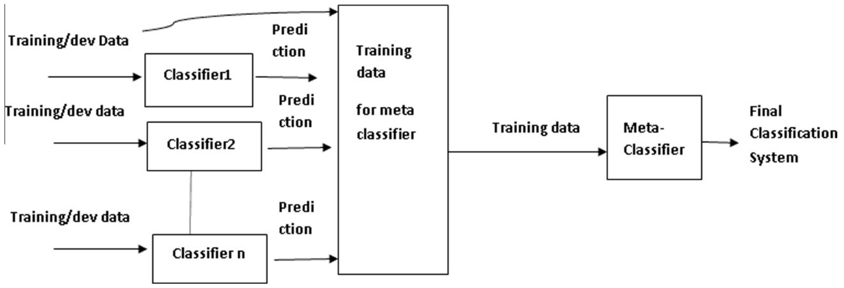


Fig. 2. A stacked based model.

NE chunks to the number of true NE chunks. The definitions of precision and recall are given below:

$$\text{precision} = \frac{\text{Number of NE chunks correctly identified by the system}}{\text{Number of NE chunks identified by the system}} \quad (2)$$

$$\text{recall} = \frac{\text{Number of NE chunks correctly identified by the system}}{\text{Number of NE chunks in the gold standard test data}} \quad (3)$$

From the definitions, it is clear that while recall tries to increase the number of tagged entries as much as possible, precision tries to increase the number of correctly tagged entries. 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:

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

For evaluation with the JNLPBA shared task dataset, we use the script available at.[9] These are the modified versions of the CoNLL-2003 shared task [22] evaluation script. The script outputs three sets of *F*-measure according to the exact, right and left boundary matches. In the right boundary matching only right boundaries of entities are considered without matching left boundaries and vice versa. For evaluation with the GENETAG dataset we use the same strict matching criterion that was followed in the Biocreative-II shared task evaluation script[10] for the gene mention detection task.

6.2. Experiments on GENTAG datasets

We evaluate our proposed approach on GENETAG dataset, which is a variant of the benchmark dataset of Biocreative-II gene mention detection task. GENETAG covers a more general domain of PubMed. It contains both true and false gene or protein names in a variety of contexts. In GENETAG, not all the sentences of abstracts were included, rather more named entity (NE) informative sentences were considered. GENETAG selects longer text fragments as entity reference, includes the semantic category word 'protein' for protein annotation, and is more inclined to select more descriptive expressions for protein annotations. During annotations of GENETAG dataset, some semantic constraints were chosen to make sure that tagged entities must contain their true meanings in the sentence contexts. Based on the gene names from GeneBank,[11] the GENETAG corpus includes domains, complexes, subunits, and promoters when the annotated entities refer to specific genes/proteins.

We evaluate our proposed approach with the GENETAG training and test datasets, available at.[12] Entities related to gene mentions in both the training and test datasets were annotated with the 'NEWGENE' tag and the overlapping gene mentions were distinguished by another tag 'NEWGENE1'. However, in this work, we use the standard IOB2 notations (as in GENIA corpus of JNLPBA 2004 shared task, c.f. Section 6.3) to properly denote the boundaries of gene names, and we replace all the 'NEWGENE1' tags by 'NEWGENE' for training and testing. The training dataset contains 7500 sentences with 8881 gene mentions. The average length per protein (or, gene) mention is 2.1 tokens. The test dataset consists of 2500 sentences with 2986 gene mentions.

In the first stage, GA based feature selection technique produces a set of solutions in the best population. Each of these solutions represents different feature combinations. Based on the feature combinations of the best population, we generate several models based on CRF and SVM. Results of the individual models are reported in Table 2. Each of these classifiers is trained with the set of features as described in SubSection 2. The highest performance corresponds to a SVM based classifier (c.f. SVM_4 in Table 2) that yields the overall recall, precision and *F*-measure values of 94.41%, 93.50% and 93.95%, respectively. The base classifiers are combined using three different ensemble techniques, namely *majority vote based ensemble*, *weighted vote based ensemble* and our proposed *stack based ensemble*. For stacking we use CRF as the meta-classifier that makes use of the following set of features:

Context window size of $[-1, +1]$, prefixes of size 4 of the window $[-2, +2]$, suffixes of size 4 of the window $[-2, +2]$, word length, infrequent word, normalization feature, orthographic feature, PoS information, trigger words, unknown word feature, head noun feature, word class feature, informative words feature, semantic feature, and bigram feature.

The proposed stack based ensemble technique (see Table 3) achieves the overall recall, precision and *F*-measure values of 95.12%, 94.29% and 94.70%, respectively. It is actually the increments of 0.75%, 0.65% and 0.41% of *F*-measure points over the best individual classifier, *Baseline 1* and *Baseline 2*, respectively.

We compare the performance of our proposed system with some other biomedical entity extraction systems that made use of the same datasets, i.e. GENTAG. We compare with the systems reported in the BioCreative-2 challenges as well as with those that were developed at the later stages but made use of the same datasets. Our system does not use any deep domain knowledge and/or external resources. Almost all the features were automatically extracted from the training dataset. In our experiment, we use only PoS and chunk (or, phrase) information as the domain dependent knowledge. We present the comparative evaluation results in Table 4 not only with the domain-independent systems but also with the systems that incorporate deep domain knowledge

[9] http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html.

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

[11] http://www.ncbi.nlm.nih.gov/Genbank/.

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

Table 2
Evaluation results on GENETAG datasets with various feature subsets. Here, the following abbreviations are used: 'CW':Context words, 'PS': Size of the prefix, 'SS': Size of the suffix, 'WL': Word length, 'IW': Infrequent word, 'NO': Normalization feature, 'Chunk': Chunk information, 'PoS': PoS information, 'OR': Orthographic feature, 'Tri': Trigger word, 'HN': Head noun feature, 'Dyn': Dynamic feature, 'UN': Unknown word feature, 'WC': Word class feature, 'IN': Informative words feature, FT: Feature template for CRF, B: Bigram feature template of CRF, 'Ct': Content word feature, $[-i, j]$: context words spanning from the left ith position to the jth right position, All $[-i, j]$: All feature combinations within the context except dynamic NE for the left ith positions, X: Denotes the presence of the corresponding feature, 'r': recall, 'p': precision, 'F': *F*-measure (we report percentages).

Classifiers	CW	PS	SS	WL	IW	NO	OR	PoS	Tri	Ct	Dyn	UN	HN	WC	IN	FT	r	p	F
CRF$_1$	[−1, +1]	3	4	X		X	X	X	X	X			X	X	X	B	94.79	92.08	93.42
CRF$_2$	[−1, +1]	1	2		X	X	X	X	X	X		X	X		X	B	94.67	92.36	93.50
CRF$_3$	[−1, +1]	2	4			X	X	X	X	X			X		X	B	94.82	92.17	93.47
CRF$_4$	[−1, +1]	2	2	X	X	X	X	X	X	X		X	X		X	B	94.73	92.25	93.48
CRF$_5$	[−2, +2]	2	3	X	X	X	X	X	X	X		X	X		X	B	93.74	92.16	93.43
CRF$_6$	[−3, +3]	2	2	X	X	X	X	X	X	X		X	X		X	B	94.70	92.30	93.48
CRF$_7$	[−4, +4]	1	2	X	X	X	X	X	X	X		X	X		X	B	94.78	92.08	93.42
CRF$_8$	[−1, +1]	1	3			X	X	X	X	X			X		X	B	94.80	92.21	93.49
CRF$_9$	[−1, +1]	1	2			X	X	X	X	X			X		X	B	94.80	92.21	93.49
SVM$_1$	[−1, +1]	4	4	X	X	X	X	X	X		−1	X	X	X	X		93.60	91.51	92.54
SVM$_2$	[−2, +2]	4	4	X	X	X	X	X	X		−2	X	X	X	X		94.27	93.21	93.74
SVM$_3$	[−2, +2]	4	4	X	X	X	X	[−2, +2]	X		−2	X	X	X	X		94.20	93.36	93.65
SVM$_4$	[−2, +2]	4	4	X	X	X	X	[−2, 0]	X		−2	X	X	X	X		94.41	93.50	93.95
SVM$_5$	[−2, +2]	3	3	X	X	X	X	X	X		−2	X	X	X	X		94.14	93.14	93.64
SVM$_6$	[−2, +2]	3	3	X	X	X	X	[−2, +2]	X		−2	X	X	X	X		94.07	93.32	93.69
SVM$_7$	[−2, +2]	3	3	X	X	X	X	[−2, 0]	X		−2	X	X	X	X		94.10	93.29	93.69
SVM$_8$	[−3, +3]	4	4	X	X	X	X	X	X		−3	X	X	X	X		94.17	93.14	93.65

Table 3
Overall evaluation results on GENETAG datasets (training: GENETAG, test: GENETAG).

Classification Scheme	recall	precision	*F*-measure
Best individual classifier	94.41	93.50	93.95
Majority Vote Based Ensemble	94.45	93.65	94.05
Weighted Vote Based Ensemble	94.67	93.91	94.29
Stacked based ensemble	95.12	94.29	94.70

Table 4
Comparison with the existing approaches for GENETAG data set

System	Approach used	Domain knowledge/ resources	*F*-measure
Our proposed system	Stacked ensemble (CRF and SVM)	PoS, phrase	94.70
Song et al. [6]	SVM	–	66.7
Bickel et al. [7]	SVM	a dictionary	72.1
Kinoshita et al. [23]	TnT [24], the Trigrams 'n' Tags	dictionary based postprocessing HMM-based part-of-speech tagger	80.9
Mitsumoriet al. [8]	SVM	gene/protein name dictionary	78.09
Finkel et al. [9]	ME + post processing		82.2
McDonald and Pereira [10]	CRF		82.4
GuoDong et al. [11]	HMM, SVM, Ensemble technique	Post processing	82.58

and/or external resources. We systematically analyze the contribution of each feature, and it reveals the fact that huge performance gain is achieved with the PoS information which was provided with the dataset. After observing this remarkable performance gain we analyzed each step of our implementation thoroughly. It seems that one possible explanation behind this radical improvement could be as follows. It is to be noted that in the GENETAG training and test datasets, PoS information were provided only for the non-gene proteins. We preprocessed this data and assigned the PoS class, NNP, i.e. proper noun to each of these gene tokens. This PoS information actually plays a crucial role in the overall system performance.

6.3. Experiments on GENIA datasets

In order to show the generalization of the developed two-stage algorithm we apply the proposed technique on the JNLPBA 2004 shared task datasets.[13] The data sets were extracted from the GENIA Version 3.02 corpus of the GENIA project. This was constructed by a controlled search on Medline using MeSH terms such as *human*, *blood cells* and *transcription factors*. From this search, 2000 abstracts of about 500 K wordforms were selected and manually annotated according to a small taxonomy of 48 classes based on a chemical classification. Out of these classes, 36 classes were used to annotate the GENIA corpus. In the shared task, the data sets were further simplified to be annotated with only five NE classes, namely *Protein*, *DNA*, *RNA*, *Cell_line* and *Cell_type* [16]. The test set was relatively new collection of Medline abstracts from the GENIA project. The test set contains 404 abstracts of around 100 K words. One half of the test data was from the same domain as that of the training data and the rest half was from the super domain of *blood cells* and *transcription factors*. For simplification, embedded structures were removed leaving only the outermost structures (i.e. the longest tag sequence). Consequently, a group of coordinated entities involving ellipsis were annotated as one structure like in the following example:

... in [lymphocytes] and [$T -$ and $\mathrm{B} -$ lymphocyte] count in ...

In the example, 'T- and B-lymphocyte' was annotated as one structure but involves two entity names, 'T-lymphocyte' and 'B-lymphocyte', whereas 'lymphocytes' was annotated as one. In order to properly denote the boundaries of NEs, five classes are further divided using the IOB2 format, where 'B-XXX' refers to the beginning of a multi-word/single-word NE of type 'XXX', 'I-XXX' refers to the intermediate parts of the NE and 'O' refers to the entities outside the NE.

Similar to the GENETAG domain, we build a number of different CRF and SVM based classifiers by varying the various available features selected after application of the GA based feature selection technique. The feature selection technique is applied on the following set of features:

(1). various contexts within the previous and next three words, i.e. $w_{i-3}^{i+3} = w_{i-3} \ldots w_{i+3}$, (2). word suffixes and prefixes of length up to three (3 + 3 different features) or four (4 + 4 different features)

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

Table 5
Evaluation results on JNLPBA dataset (GENIA) with various feature subsets. Here, the following abbreviations are used: 'CW':Context words, 'PS': Size of the prefix, 'SS': Size of the suffix, 'WL': Word length, 'IW': Infrequent word, 'NO': Normalization feature, 'Chunk': Chunk information, 'PoS': PoS information, 'OR': Orthographic feature, 'Tri': Trigger word, 'HN': Head noun feature, 'Ct': Content words, 'Dyn': Dynamic feature, 'UN': Unknown word feature, 'WC': Word class feature, 'IN': Informative words, FT: Feature template for CRF, B: Bigram feature template of CRF, [−i, j]: context words spanning from the left ith position to the jth right position, All [−*i*, *j*]: All feature combinations within the context except dynamic NE for the left *i*th positions, X: Denotes the presence of the corresponding feature, 'r': recall, 'p': precision, 'F': *F*-measure (we report percentages).

Classifiers	CW	PS	SS	WL	IW	NO	Chunk	OR	PoS	Tri	Ct	Dyn	UN	HN	WC	IN	FT	r	p	F
CRF_1	[−2, +2]	4	4				X	X	X	X	X		X	X	X	X	B	72.0	75.91	74.10
CRF_2	[−3, +3]	3	3				X	X	X	X				X		X	B	71.82	76.10	73.90
CRF_3	[−3, +2]	4	4	X		X	X	X	X	X	X			X		X	B	72.21	75.93	74.0
CRF_4	[−1, +1]	4	4		X		X	X	X	X				X	X	X	B	72.21	76.10	74.11
CRF_5	[−2, +2]	4	4			X	X	X	X	X			X	X	X	X	B	72.37	76.34	74.30
CRF_6	[−3, +3]	4	4	X	X	X	X	X	X	X			X	X	X	X	B	73.10	76.78	74.90
CRF_7	[−3, +3]	4	4				X	X	X	X	X			X	X	X	B	72.14	75.73	73.89
CRF_8	[−3, +2]	4	4			X	X	X	X	X	X		X	X		X	B	72.47	76.69	74.52
CRF_9	[−1, +1]	4	4				X	X	X	X	X			X	X	X	B	72.25	76.14	74.15
SVM_1	[−3, +3]	4	4	X	X	X	X	X	X	X	X	−2	X	X	X	X		67.70	66.34	67.01
SVM_2	[−3, +3]	4	4	X	X	X	[−2, +2]	X	X	X	X	−2	X	X	X	X		72.43	66.65	69.42
SVM_3	[−3, +3]	4	4	X	X	X	[−2, +2]	X	[−2, +2]	X	X	−2	X	X	X	X		72.82	67.08	69.83
SVM_4	[−3, +3]	4	4	X	X	X	[−1, +1]	X	[−1, +1]	X	X	−2	X	X	X	X		72.86	66.96	69.78
SVM_5	[−3, +3]	4	4	X	X	X	X	X	[−2, +2]	X	X	−2	X	X	X	X		73.05	67.04	69.92
SVM_6	[−3, +3]	4	4	X	X	X	X	X	X	X	X	−3	X	X	X	X		72.47	66.71	69.47
SVM_7	[−3, +3]	4	4	X	X	X	X	X	X	X	X	−3, All [−1, +1]	X	X	X	X		74.77	68.71	71.61
SVM_8	[−3, +3]	4	4	X	X	X	[−1, +1]	X	[−1, +1]	X	X	−3	X	X	X	X		72.45	66.60	69.40

characters of words within the context window $w_{i-2}^{i+2} = w_{i-2} \dots w_{i+2}$, (3). PoS information of the current and/or the surrounding token(s), (4). Chunk information of the current and/or the surrounding token(s), (5). Dynamic NE tag(s) of the previous token(s), (6). Word normalization, (7). Word length, (8). Infrequent word, (9). Unknown tokens, (10). Head nouns (unigram and bigram), (11). Verb trigger, (12). Word class, (13). Informative NE information, (14). Content words, and (15). Orthographic features.

We construct several different versions of CRF and SVM based classifiers by utilizing various features and/or feature templates of the final best population. Here, in Table 5, we show only the results of good-performing 9 and 8 CRF and SVM based classifiers, respectively. Feature combinations are mainly varied with local contexts, prefixes and suffixes. For example the results in Table 5 show that the first classifier has all the features with context in the window of [−2, +2] (i.e. previous two and next two words), and suffixes and prefixes of length up to four characters. The second classifier is constructed with all features but with context window of [−3, +3], and prefixes and suffixes of length up to three characters. We use all the features for training and use the default settings for all other parameters of CRF. For SVM, we use the *polynomial kernel function* of degree two. The CRF-based model exhibits the best performance with the recall, precision and *F*-measure values of 73.10%, 76.78% and 74.90%, respectively. The corresponding feature template (6th row) considers the contexts of previous and next three tokens along with their all possible n-gram ($n \leqslant 3$) combinations from left to right, prefixes and suffixes of length up to four characters of only the current word, feature vector consisting of length, infrequent word, normalization, chunk, orthographic constructs, trigger word, unknown word, head noun, word class, informative NE information of only the current token, and bigram feature combinations. For SVM, all the feature combinations within the context of previous one and next one words are very effective to improve the overall system performance (i.e. SVM_7).

In Table 6, we analyze the effects of each feature inclusion on the overall system performance. We present the analysis only for CRF as this yields the best individual performance. Results show that the system achieves the *F*-measure value of 69.05% with the local contexts of preceding three and following three words along with the orthographic features. The prefixes and suffixes of length up to four characters improve the overall *F*-measure value by 2.90 points. The PoS and chunk information show the increments of less than one point. Results clearly show the effectiveness of "informative NE words" and "head noun" features. The content word feature (or, semantic feature) that exploits global context information does not contribute to the improvement to overall system performance. Please note that for this setting we used the PoS class NNP (denoting proper nouns) to extract content word feature from the test data.

We also show the detailed evaluation results (recall, precision and *F*-measure values of individual output classes) in Table 8 for the best individual classifier.

Table 6
Evaluation results on JNLPBA dataset (GENIA) with various feature subsets; different features are added to the best performing classifier one by one.

Classifiers	CW	PS	SS	WL	IW	NO	Chunk	OR	PoS	Tri	Se	Dyn	UN	HN	WC	IN	FT	r	p	F
CRF_1	[−3, +3]							X									B	67.88	70.26	69.05
CRF_2	[−3, +3]							X	X								B	69.16	70.02	69.59
CRF_3	[−3, +3]						X	X	X								B	69.38	70.04	69.71
CRF_4	[−3, +3]	4					X	X	X								B	71.26	70.74	71.00
CRF_5	[−3, +3]	4	4				X	X	X								B	72.43	71.48	71.95
CRF_6	[−3, +3]	4	4			X	X	X	X						X		B	72.62	71.54	72.08
CRF_7	[−3, +3]	4	4			X	X	X	X				X		X		B	72.37	71.43	71.89
CRF_8	[−3, +3]	4	4			X	X	X	X	X			X		X		B	72.51	71.75	72.13
CRF_9	[−3, +3]	4	4			X	X	X	X	X			X	X	X		B	72.22	72.13	71.68
CRF_{10}	[−3, +3]	4	4			X	X	X	X	X			X	X	X	X	B	76.63	72.93	74.73
CRF_{11}	[−3, +3]	4	4	X	X	X	X	X	X	X			X	X	X	X	B	76.78	73.10	74.90
CRF_{12}	[−3, +3]	4	4	X	X	X	X	X	X	X	X		X	X	X	X	B	76.63	73.04	74.79

Table 7
Overall evaluation results on GENIA data set (we report percentages).

Model	recall	precision	*F*-measure
Best individual classifier	73.10	76.78	74.90
Baseline 1	71.03	75.76	73.32
Baseline 2	71.42	75.90	73.59
Stacked based ensemble	75.15	75.20	75.17

Table 8
Evaluation results of the best individual classifier on GENIA data set for individual NE classes.

Class	Recall	Precision	*F*-measure
Overall			
FULLY correct	76.78	73.10	74.90
Correct LEFT boundary	80.56	76.69	78.58
Correct RIGHT boundary	83.98	79.95	81.92
Protein			
FULLY correct	82.31	73.22	77.50
Correct LEFT boundary	86.89	77.30	81.81
Correct RIGHT boundary	88.70	78.91	83.51
Cell_line			
FULLY correct	59.29	56.62	57.93
Correct LEFT boundary	64.31	61.41	62.82
Correct RIGHT boundary	71.68	68.45	70.03
DNA			
FULLY correct	74.03	72.61	73.31
Correct LEFT boundary	76.46	75.00	75.72
Correct RIGHT boundary	81.17	79.62	80.39
RNA			
FULLY correct	71.83	72.86	72.34
Correct LEFT boundary	74.65	75.71	75.18
Correct RIGHT boundary	80.28	81.43	80.85
Cell_type			
FULLY correct	69.21	78.95	73.76
Correct LEFT boundary	71.25	81.28	75.93
Correct RIGHT boundary	76.93	87.75	81.99

- FULLY correct: the boundaries predicted by our proposed system and those of the gold standard data match on both sides.
- Correct LEFT boundary: the boundaries predicted by the system and that of the gold standard data are same on the left side.
- Correct RIGHT boundary: the boundaries determined by our proposed system and that of the gold standard match on the right side.

All the individual models of CRF and SVM are thereafter combined with the proposed stacked based ensemble. Here a CRF based classifier is used as the meta classifier in the second stage. It uses the following feature template: Context window size of [-1, +1], prefixes of size 4 of the window [-2, +2], suffixes of size 4 of the window [-2, +2], word length, infrequent word, normalization feature, orthographic feature, PoS information, trigger words, unknown word feature, head noun feature, word class feature, informative words feature, and bigram feature. The overall evaluation results obtained by the stacked ensemble technique are presented in Table 7. The proposed ensemble technique shows the recall, precision and *F*-measure values of 75.15%, 75.20% and 75.17%, respectively. This is superior to the best individual model, *majority vote based ensemble* and *weighted vote based ensemble* by 0.27%, 1.85% and 1.58% *F*-measure points, respectively. It is interesting to note that all the baseline models perform lower compared to the best individual model. This may be due to the fact that instead of making prioritization among the classes in each classifier, *baseline* techniques blindly combine all the available classifiers. This type of behavior also indicates that the performance of an ensemble system greatly depends on the selection of appropriate votes per output class in each classifier.

We compare the performance of our proposed system with other biomedical entity extraction systems that made use of the same GENIA dataset. We compare with the systems, developed with same datasets. Our system does not make use of any deep domain knowledge and/or external resources. In our experiment, we use only PoS and chunk (or, phrase) information as the domain dependent knowledge. So, it will not be fair to compare the performance of our stack based ensemble with all the available systems. However, we present the comparative evaluation results in Table 9 not only with the domain-independent systems but also with the systems that incorporate deep domain knowledge and/or external resources.

Zhou and Su [25] developed the best system in the JNLPBA 2004 shared task. This system provides the highest *F*-measure value of 72.55 with several deep domain knowledge sources. But when the system used only PoS and chunk information as the domain knowledge, the *F*-measure value drops to 64.1%. Song et al. [30] used CRF and SVM both, and obtained the *F*-measure of 66.28% with virtual samples. The HMM-based system reported by Ponomareva et al. [31] achieved a *F*-measure value of 65.7% with PoS and phrase-level domain dependent knowledge. A ME-based system was reported in [29] where recognition of terms and their classification were performed in two steps. They achieved a *F*-measure value of 66.91% with several lexical knowledge sources such as salient words obtained through corpus comparison between domain-specific and WSJ corpora, morphological patterns and collocations extracted from the Medline corpus. As far our knowledge is concerned, one of the very recent works proposed in [17] obtained the *F*-measure value of 67.41% with PoS and phrase information as the only domain knowledge. This is the highest performance achieved by any system that did not use any deep domain knowledge.

A CRF-based NE extraction system has been reported in [28] that obtained the *F*-measure value of 70% with orthographic

Table 9
Comparison with the existing approaches for GENIA data set.

System	Used approach	Domain knowledge/resources	FM
Our proposed system	Classifier ensemble (CRF and SVM)	POS, phrase	75.17
Zhou and Su [25] Final	HMM, SVM	Name alias, cascaded NEs dictionary, POS, phrase	72.55
Zhou and Su [25]	HMM, SVM	POS, phrase	64.1
Kim et al. [26]	Two-phase model with ME and CRF	POS, phrase, rule-based component	71.19
Finkel et al. [27]	ME	Gazetteers, web-querying, surrounding abstracts, abbreviation handling, BNC corpus, POS	70.06
Settles [28]	CRF	POS, semantic knowledge sources of 17 lexicons	70.00
Saha et al. [17]	ME	POS, phrase	67.41
Park et al. [29]	ME	POS, phrase, domain-salient words using WSJ, morphological patterns, collocations from Medline	66.91
Song et al. [30] Final	SVM, CRF	POS, phrase, Virtual sample	66.28
Song et al. [30] Base	SVM	POS,phrase	63.85
Ponomareva et al. [31]	HMM	POS	65.7

features, semantic knowledge in the form of 17 lexicons generated from the public databases and Google sets. Finkel et al. [27] reported a CRF-based system that showed the *F*-measure value of 70.06% with the use of a number of external resources, including gazetteers, web-querying, surrounding abstracts, abbreviation handling method, and frequency counts from the BNC corpus. A two-phase model based on ME and CRF was proposed by Kim et al. [26] that achieved a *F*-measure value of 71.19% by postprocessing the outputs of machine learning models with a rule-based component. We also compare the performance of our proposed ensemble based approach with BANNER [32] that was implemented using CRFs. BANNER exploits a range of orthographic, morphological and shallow syntax features, such as part-of-speech tags, capitalisation, letter/digit combinations, prefixes, suffixes and Greek letters. Comparisons between the several existing NE extraction systems are provided in [33]. For BANNER, Kabiljo et al. [33] reported the *F*-measure values of 77.50% and 61.00% under the sloppy matching and strict matching criterion, respectively with the JNLPBA shared task datasets.

In summary, our proposed two-stage approach attains the state-of-the-art performance levels for entity extraction in two different kinds of biomedical datasets. The possible reasons behind the better performance in our proposed approach are the (i). use of variety and rich features as described in Section 2; (ii) use of GA based feature selection technique to identify appropriate subset of features of any classifier, particularly for the problem of biomedical entity extraction; and (iii). use of stack based ensemble approach that effectively combines the classifiers and further improves the overall performance.

7. Conclusion and future works

In this paper we have proposed a two-stage approach for biomedical entity extraction, where the gene or gene-product names are identified and then classified into some predefined categories of interest. At the first stage, a GA based feature selection technique is implemented to determine the best set of features for CRF and SVM based classifiers. We came up with a very rich feature set that itself can achieve very high accuracy. The most important characteristic of our system is that all the features are mostly identified and developed without using any deep domain knowledge and/or external resources. The GA based approach identifies a set of best solutions on the final population. The chromosomes in this population represent different feature combinations for CRF and SVM. Several different CRF and SVM based classifiers are generated varying these set of features. These classifiers are then combined using a stacked ensemble technique. As a meta-classifier we have used CRF as the classifier in the second stage of our algorithm. Evaluation results for GENETAG and GENIA benchmark datasets have shown the overall *F*-measure values of 94.70% and 75.17%, respectively. Detailed comparisons show that our proposed technique achieves state-of-the-art performance.

In future we would like to add some more features based on external resources like gene/protein dictionary. Use of multiobjective optimization (MOO) for feature selection in biomedical entity extraction will be an interesting experiment to be carried out. Here for GA based feature selection technique we have optimized only a single classification quality measure, i.e. *F*-measure. But in entity extraction there are some other classification quality measures like recall and precision. More than one such classification quality measures can be efficiently optimized using the search capability of MOO.

References

[1] J. Finkel, S. Dingare, C. Manning, M. Nissim, B. Alex, C. Grover, Exploring the boundaries: gene and protein identification in biomedical text, BMC Bioinformatics 6 (2005).

[2] A. AR, B. O, C. HF, H. SM, M. JG, N. SJ, R. TC, W. WJ, The NLM Indexing Initiative, in: Proceedings of 2000 AMIA Annual Fall Symposium.

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

[4] L. Hirschman, M. Krallinger, e. Alfonso Valencia, in: Proceedings of the Second BioCreative Challenge Evaluation Workshop, 23rd–25th of April 2007.

[5] L. Hirschman, A. Yeh, C. Blaschke, A. Valencia, Overview of BioCreAtIvE: critical assessment of information extraction for biology, BMC Bioinformatics 6 (2005).

[6] Y. Song, E. Yi, E. Kim, G.G. Lee, POSBIOTM-NER: a machine learning approach for bio-named entity recognition, in: Workshop on a Critical Assessment of Text Mining Methods in Molecular Biology, 2004.

[7] S. Bickel, U. Brefeld, L. Faulstich, J. Hakenberg, U. Leser, C. Plake, T. Scheffer, A support vector machine classifier for gene name recognition, in: Proceedings of the EMBO Workshop: A Critical Assessment of Text Mining Methods in Molecular Biology, 2004.

[8] T. Mitsumori, S. Fation, M. Murata, K. Doi, H. Doi, Gene/protein name recognition based on support vector machine using dictionary as features, BMC Bioinformatics 6 (Suppl. 1) (2005).

[9] J. Finkel, S. Dingare, C.D. Manning, M. Nissim, B. Alex, C. Grover, Exploring the boundaries: gene and protein identification in biomedical text, in: Proceedings of the BioCreative Workshop, 2004.

[10] R. McDonald, F. Pereira, Identifying gene and protein mentions in text using conditional random fields, BMC Bioinformatics 6 (Suppl. 1) (2005).

[11] G.D. Zhou, D. Shen, J. Zhang, J. Su, S.H. Tan, Recognition of protein/gene names from text using an ensemble of classifiers, BMC Bioinformatics 6 (Suppl. 1) (2005).

[12] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502. http://dx.doi.org/10.1109/TKDE.2005.66.

[13] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Norwell, MA, USA, 1998.

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

[15] D. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[16] K. Jin-Dong, O. Tomoko, T.Y., et al., Introduction to the bio-entity recognition task at jnlpba, in: JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Association for Computational Linguistics, 2004, pp. 70–75.

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

[18] J.H. Holland, Adaptation in Natural and Artificial Systems, Springer, Berlin, 1975.

[19] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: ICML, 2001, pp. 282–289.

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

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

[22] E.F. Tjong Kim Sang, F. De Meulder, Introduction to the Conll-2003 shared task: language independent named entity recognition, in: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, 2003, pp. 142–147.

[23] S. Kinoshita, K.B. Cohen, P.V. Ogren, L. Hunter, BioCreAtIvE Task1A: entity identification with a stochastic tagger, BMC Bioinformatics 6 (Suppl. 1), S4. doi:10.1186/1471-2105-6-S1-S4.

[24] B. T, TnT a statistical part-of-speech tagger, in: Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000), 2000.

[25] Z. GuoDong, S. Jian, Exploring deep knowledge resources in biomedical name recognition, in: JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, 2004, pp. 96–99.

[26] S. Kim, J. Yoon, K.-M. Park, H.-C. Rim, Two-phase biomedical named entity recognition using a hybrid method, in: IJCNLP, 2005, pp. 646–657.

[27] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, G. Sinclair, C. Manning, Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web, in: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004), 2004, pp. 88–91.

[28] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, in: JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Association for Computational Linguistics, 2004, pp. 104–107.

[29] K.-M. Park, S.-H. Kim, H.-C. Rim, Y.-S. Hwang, Me-based biomedical named entity recognition using lexical knowledge, ACM Trans. Asian Lang. Inform. Process. 5 (2004) 4–21.

[30] Y. Song, E. Kim, G.G. Lee, B. Yi, Posbiotm-ner in the shared task of bionlp/nlpba 2004, in: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004), 2004.

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

[32] R. Leaman, G. Gonzalez, BANNER: an executable survey of advances in biomedical named entity recognition, Proceedings of the Pacific Symposium on Biocomputing (2008).

[33] R. Kabiljo, A.B. Clegg, A.J. Shepherd, A realistic assessment of methods for extracting gene/protein interactions from free text, BMC Bioinformatics 10 (2009).