Soft Comput (2018) 22:6881–6904
https://doi.org/10.1007/s00500-017-2714-4

METHODOLOGIES AND APPLICATION

Feature selection for entity extraction from multiple biomedical corpora: A PSO-based approach

Shweta Yadav[1] · **Asif Ekbal**[1] · **Sriparna Saha**[1]

Published online: 17 August 2017

Abstract Entity extraction is an important step in biomedical text mining. Among many other challenges, there are two very crucial issues, *viz*. determining the most applicable feature set so that the model can be precise and less complex, and adapting the system across multiple benchmark corpora. In this paper, we propose a novel method for feature selection using the search capability of particle swarm optimization. The compact feature set used for training the classifier yields much better results when compared to the baseline model, which was developed with a complete set of features. A large number of features suitable for named entity recognition task from biomedical domain are also developed in the current paper. The complete set of features is implemented by studying the properties of datasets and from the domain knowledge. We have used conditional random field, a robust classifier as the underlying learning algorithm which has shown success in solving similar kinds of problems. Our experiments on multiple benchmark corpora yield the level of performance which are at par the state-of-the-art techniques.

Keywords Particle swarm optimization (PSO) · Feature selection · Condition random field · Entity extraction

Communicated by V. Loia.

✉ Shweta Yadav
shweta.pcs14@iitp.ac.in

Asif Ekbal
asif@iitp.ac.in

Sriparna Saha
sriparna@iitp.ac.in

[1] Department of Computer Science Engineering, Indian Institute of Technology Patna, Patna, Bihar 800013, India

1 Introduction

The abundance of biomedical information available in the web has necessitated developing robust models for extracting relevant information from these unstructured sources of biomedical data. Manual curation of relevant information incurs huge complexities in terms of time and manpower. In the previous 20 years, biomedical literature has increased rapidly. PubMed known to be one of the huge biomedical database consists of more than 25 million citations from various sources like biomedical literature (MEDLINE), life science journals. An increment of around 4.2% can be observed in the size of MEDLINE every year. Therefore, developing automated techniques for finding the most relevant information or discovering new patterns from huge amount of unstructured data is highly desirable. Entity extraction is such a task, which is a very crucial step in biomedical information extraction. It focuses on classifying the identified biomedical entities from text into the predefined categories of interest. Biomedical entities mostly refer to gene or gene-like products, such as Protein, DNA, RNA, cell_line, cell_type (Kim et al. 2004). Literature shows the evidence of a significant number of approaches for entity extraction, but still the best proposed technique in the biomedical domain lacks behind by almost 8–10 points as compared to the traditional newswire domain. Some of the challenges that we encounter for solving the problem are as follows:

- Existence of multiple benchmark corpora which, often, makes it difficult to adapt a system developed for a domain to the other.
- Continuous expansion of new named entities (NEs), while still there does not exist any proper dictionary for several types of biomedical NE.

– Similar words convey different meanings, and therefore, a word can have multiple NE types.
– There is no standard nomenclature for biomedical NEs, and so this arbitrariness makes it difficult to come up with a very well-defined set of rules that captures the properties of names very well.
– Biomedical names are of long length and possess different types of symbols, and so its boundary detection becomes problematic.
– NEs are often embedded with each other. Identifying these types of NEs requires more effort.

All these problems have contributed heavily to the drop of accuracies in entity extraction problem from biomedical datasets. In biomedical domain, many benchmark corpora exist which are developed using various annotation scheme. Therefore the system, developed by targeting a domain, often fails to show reasonable accuracy when it is evaluated in other domain. Thus, it is highly desirable to develop a generic system that can be easily adapted to several domains.

This paper focuses on reducing dimensionality of the biomedical data set that can lead to overcome some major problems that we face in retrieving the named entities. Feature selection (a.k.a. attribute selection) (Yu and Liu 2004) is an useful preprocessing step that aids in achieving good accuracy in many applications related to machine learning, data mining, pattern recognition etc. It is the way in which the subset of appropriate features is selected for model construction. On the basis of some evaluation criterion, feature selection is performed to reduce the feature space. The basic assumption of using feature selection technique is that, data contain some redundant or insignificant features. Performance of classifier is fully dependent on the features that we use for training. The main purpose of feature selection is to simplify the dataset and find the feature subset that results in high classification accuracy. The two most popular paradigms of feature selection are 'filter' and 'wrapper' model (Das 2001). Feature selection using filter model is independent of the learning algorithm that is used. Features are chosen prior to the development of model just by analyzing some properties of those features. On the other hand, wrapper model selects the features on the basis of some learning algorithm. Evaluation of the attributes is done on the basis of accuracy estimation that takes into view of the actual training algorithm. Literature shows that wrapper model is effective than filter-based model as the list of features is selected according to the predetermined learning algorithm which results in higher accuracy. However, wrapper model is computationally expensive compared to the filter model.

In this paper, we propose a novel method for feature selection based on wrapper model that determines the most optimized feature set for a classifier. The feature set, thus obtained, when used to train the classifier, improves performance. The method we propose is based on the search capability of particle swarm optimization (PSO) (Kennedy and Eberhart 1997), which is a popular evolutionary algorithms inspired by the behavior of birds. This paper presents the application of PSO-based feature selection for solving the problem of entity extraction from biomedical data.

RFLAT-1: a new zinc finger transcription factor that activates RANTES gene expression in T lymphocytes. RANTES (Regulated upon Activation, Normal T cell Expressed and Secreted) is a chemoattractant cytokine (chemokine) important in the generation of inflammatory infiltrate and human immunodeficiency virus entry into immune cells. RANTES is expressed late (3-5 days) after activation in T lymphocytes.

Fig. 1 Sample biomedical sentence from JNLPBA corpus

We extract the features by studying the properties of the biomedical datasets, and this process is completely automatic. The features that we use are generic in nature and thus useful for more than one biomedical datasets. The biomedical datasets that we use are not of similar kinds and were created following different annotation guidelines. Therefore, the system which is developed by tuning heavily on a specific domain often fails to perform reasonably whenever the domain is altered. One of our primary goals was to develop a system that could perform with acceptable performance level on multiple benchmark datasets, even if they are generated following different guidelines. We use the same set of features for all the datasets. We expect that PSO will determine the most relevant set of features depending upon the type of data. Thus automatic selection of feature subset for each domain is a very important step to reduce the algorithmic complexity of entity extraction. There has not been systematic attempt of feature selection for entity extraction using PSO. As such the study of effects of PSO for feature selection in this domain is really interesting and a new contribution. Figure 1 shows the snippet of the biomedical dataset taken from the benchmark JNLPBA shared task which is the natural language text extracted from the biomedical articles. Tables 1 illustrates some of the features that are extracted from this dataset. It shows that mostly the features are of non-numeric types.

Use of PSO-based feature selection, extensive experimental results and analysis are the key parts of the current paper. To prove that our system is more generalized and not biased toward any specific biomedical dataset, we perform experiments on multiple biomedical corpora, namely GENIA, AIMed, GENETAG and BioCreative-II (BC-II) gene mention challenge datasets. We use binary version of PSO, where

Table 1 Exemplar feature generation on the biomedical sentence "These data showed that sensitivity of lymphocytes to glucocorticoids changed only with a decrease of GR level"

Words	Feature 1	Feature 2	Feature 3	Feature 4	Feature 6	Feature 7	Feature 8	Feature 9	Feature 10	Feature 11	Feature 12	Feature 13	Feature 14	Feature 15	Feature 16
These	T	Th	The	e	es	ese	Aaaaa	Aa	DT	B-NP	0	N.A.	N.A.	data	showed
data	d	da	dat	a	at	ata	aaaa	a	NNS	I-NP	0	N.A.	These	showed	that
showed	s	sh	sho	d	de	dew	aaaaaa	a	VBD	B-VP	1	These	data	that	sensitivity
that	t	th	tha	t	ta	tah	aaaa	a	IN	B-SBAR	0	data	showed	sensitivity	of
sensitivity	s	se	sen	y	yt	yti	aaaaaaaaa	a	NN	I-NP	0	showed	that	of	lymphocytes
of	o	of	N.A	f	fo	N.A.	aa	a	IN	B-PP	0	that	sensitivity	lymphocytes	to
lymphocytes	l	ly	lym	s	se	set	aaaaaaaaaaa	a	NNS	B-NP	1	sensitivity	of	to	glucocorticoids
to	t	to	N.A.	o	ot	N.A.	aa	a	TO	B-PP	0	of	lymphocytes	glucocorticoids	changed
glucocorticoids	g	gl	glu	s	sd	sdi	aaaaaaaaaaaaa	a	NNS	B-NP	1	lymphocytes	to	changed	only
changed	c	ch	cha	d	de	deg	aaaaaa	a	VBD	B-VP	0	to	glucocorticoids	only	with
only	o	on	onl	y	yl	yln	aaaa	a	RB	B-ADVP	0	glucocorticoids	changed	with	a
with	w	wi	wit	h	ht	hti	aaaa	a	IN	B-PP	0	changed	only	a	decrease
a	a	N.A.	N.A.	a	N.A.	N.A.	a	a	DT	B-NP	0	only	with	decrease	of
decrease	d	de	dec	e	es	esa	aaaaaaaa	a	NN	I-NP	1	with	a	of	GR
of	o	of	N.A.	f	fo	N.A.	aa	a	IN	B-PP	0	a	decrease	GR	level
GR	G	GR	N.A.	R	RG	N.A.	AA	A	NN	B-NP	0	decrease	of	level	.
level	l	le	lev	l	le	lev	aaaaa	a	NN	I-NP	0	of	GR	.	N.A.
.	.	N.A.	N.A.	.	N.A.	N.A.	.	.	.	O	0	GR	level	N.A.	N.A.

the presence and absence of features are denoted by 1 and 0, respectively. Conditional random field (Lafferty et al. 2001), a robust statistical classifier that showed success in many sequence labeling tasks, is used as an underlying learning algorithm.

Specifically, the key contributions of the current paper can be summarized as follows:

- Biomedical text is the natural language text (an example is shown in Fig. 1). As the biomedical datasets are text based, so the way of generating the features is quite different from the other domains.
- A sophisticated set of features which can aid in proper recognition of biomedical names is utilized in the current paper. These features are generic in nature and applied on multiple domains.
- In the previous studies, PSO-based feature selection technique was applied on the datasets whose sizes were very limited. In contrast, there has not been any study that focuses on PSO-based feature selection in the biomedical domain. Thus the way the features are generated for the biomedical domain and PSO is applied for feature selection are entirely different from the existing works. In addition, the datasets that we use are varying in nature and have good sizes. We use the same set of features for all the domains. Our algorithm is generic in nature and has been evaluated on multiple benchmark datasets.
- Genetic algorithm (GA)-based feature selection has been applied for entity extraction in biomedical domains, for example, Ekbal and Saha (2013). PSO is computationally less expensive compared to GA. It has been shown in the literature that PSO is less time complex and converges faster compared to GA (Eberhart and Shi 1998). Motivated by these facts, we use PSO for feature selection in the related domains. Additionally in comparison with GA, PSO makes use of less parameters.
- Detailed analysis on the experiments carried out on four benchmark datasets is presented. Computational complexity of the algorithm is also presented.
- We compare the PSO-based feature selection technique with filter-based feature selection technique that utilizes the concept of correlation.
- We performed comprehensive comparison with the popular wrapper-based technique based on deterministic and randomized wrapper model.
- An extensive comparative study with the existing approaches are presented.

The remainder of the paper is organized as follows. Section 3 gives a brief introduction to the standard PSO and binary PSO algorithms. Section 4 provides the description of our proposed feature selection technique. We present a brief description of conditional random field that we use as a base learner in all our experimental settings. We describe the features in Sect. 5. Details of experiments and their analysis are presented in Sects. 6 and 7, respectively. Finally, we conclude in Sect. 8.

2 Literature review

In the recent past, PSO has gained a lot of attention due to its property to discover optimal sets of solution rapidly. Over the years, several variants of PSO have evolved ranging from the basic PSO which is synchronous in nature. In basic PSO, synchronization is because of the communication between the particles. Each particle shares its best position with its neighbors. Thus each particle is having information about its neighbors before updating its position. The other variant of PSO is asynchronous in nature where a particle shares its memory after moving to the new position. Thus, each particle can immediately share information without getting waited for the next iteration.

The traditional PSO takes into account only the continuous valued spaces. Thus it would be difficult to handle finite variables using PSO. The other variant of PSO, binary PSO, takes into account this drawback. In case of binary PSO, each particle's position is encoded using a binary value, i.e., 0 or 1. Other versions of PSO include discrete binary PSO (Kennedy and Eberhart 1997) which takes into account the discrete binary variables.

Fitness-distance-ratio-based PSO (FDR-PSO) (Peram et al. 2003) is another type of PSO where every particle irrespective of the memory also considers the fitness-distance ratio as an information to select particle with higher fitness value. The introduction of population manager in one of the versions of PSO known as efficient population utilization strategy for PSO (EPUS-PSO) (Hsieh et al. 2009) helps in improving the performance of PSO. This way elimination of the redundant particle is performed. Another PSO variant known as aging leader and challenge PSO (ALC-PSO) (Chen et al. 2013) enhances the PSO by removing the problem of premature convergence. Here, the maximum lifetime is assigned to the swarm on the basis of its performance.

Other more popular versions of PSO include hybrid version of PSO exploring the benefits of neural network such as (Zhang et al. 2007) using the back-propagation for training feed forward neural network. Some other approaches include the amalgamation of the genetic approach and the PSO for recurrent network design (Juang 2004). Kao and Zahara (2008) have also developed hybrid GA-PSO that includes the benefits of GA and PSO to populate new individuals for next iteration with not just by the GA basic operations like crossover and mutation but also after considering the memory concept of PSO.

Feature selection using PSO has been attempted in the several domains, particularly more in domain like pattern recognition (Krisshna et al. 2014; Ramadan and Abdel-Kader 2009), bioinformatics (Ding and Peng 2005; Chuang et al. 2008). In some of the text processing applications, PSO-based feature selection has been applied, for example in Lin et al. (2008) and Tran et al. (2014).

In recent years, PSO has been applied to solve different real-life problems. In Cagnina et al. (2008), a clustering technique is proposed using the search capability of PSO named CLUDI-PSO. This is having robust performance compared to other existing clustering techniques as shown (Cagnina et al. 2008). Apart from this Samadzadegan and Saeedi (2009); Merwe and Van der Engelbrecht (2003) have also exploited the benefits of using PSO in algorithm. For, the classification problem, recently Gupta et al. (2015), Liu et al. (2016) and Shang et al. (2016) have used the PSO-based feature selection to identify the best set of features which could help in identifying the sentiment. Lu et al. (2015) have also studied the benefits of PSO for text feature selection. Chinnaswamy et al. (2016) also explored the PSO-based feature selection for the gene-expression data. They adapted the hybrid approach to select the feature making use of the correlation coefficient in addition to the PSO. Hybrid approach to PSO was defined by Xi et al. (2010) where they proposed a binary quantum-behaved PSO (BQPSO) for selecting cancer feature. This approach was the enhanced version of the discretized version of original QPSO for binary 0–1 optimization problems. Ghamisi and Benediktsson (2015) proposed a hybrid approach-based on genetic algorithm and PSO to select features. In the domain of text categorization, Aghdam and Heidari (2015) have used PSO to select relevant features. Other PSO-based feature selection approaches include the works of Kumar et al. (2016) and Sheikhpour et al. (2016).

However, these algorithms mainly deal with the datasets whose features are limited and have numeric types. Moreover, the datasets were small in size. In order to show the proper utility of any feature selection technique, it is required to evaluate on a dataset having significantly good number of instances. In the current study, we demonstrate the utility of PSO-based feature selection on a text dataset containing approximately 4,92,506 biomedical instances with 57 features. The biomedical dataset is very unstructured in nature that contains many nested long entities, abbreviations, symbols, punctuations etc., and hence the features generated for this domain are different from those corresponding to other datasets. Features that we generated for solving the problems have mostly non-numeric values, which further complicates the process of feature extraction and processing. It is to be noted that unlike other domains, feature values are not explicitly mentioned for the biomedical textual datasets that we used.

3 Brief introduction to particle swarm optimization

Particle swarm optimization (PSO) is a meta-heuristic intelligent technique inspired by social behavior of the swarm for its survival (Eberhart and Shi 1998; Kennedy and Eberhart 1997). This is a population-based technique which is perceived in birds and fishes for the search of the best path. Here, as the name suggests, PSO, the optimization process is done on the swarm of the particles. PSO consists of the swarm of the particles where each particle has its particular position in the search space which moves around the search space by some velocity. The particle selects the best path on each iteration by using its memory and by learning the effective path that was followed previously by the swarm. The new position is chosen on the basis of the knowledge gained previously by its self best position and the best position of the swarm. PSO, being a meta-heuristic model, makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. This makes PSO highly efficient for the optimization purpose (Yan et al. 2013). The algorithm iterates by keeping track of two variables:

Global best solution that represents the most promising vector found so far and personal best solution which denotes the particle's own personal best solution.

In PSO, a possible set of solutions to the problem is defined in the search space of n-dimensions using the vector of particles as $\overrightarrow{P}(i) = (p_{(i,1)}, p_{(i,2)}, \ldots, p_{(i,n)})$. Each particle moves in the search space through some rate of change, i.e., velocity, $v_{(i,k)}$ where $k = 1, 2, \ldots, n$. 'pBest' represents the best position found so far, $f(\overrightarrow{B}(i))$ denotes the best fitness function value of the particle and 'gBest' is the global best position that indicates best solution in whole swarm, $f(G(i))$ represents the fitness value of the swarm (Liu et al. 2011). *rand* specifies any random number generated. Given a candidate solution $f(\overrightarrow{P}(i))$, called the fitness function, represents a merit value that gives overview of our solution reaching to the goal. Each particle maintain a memory to keep track of:

- A velocity value which directs the movement of particles in the search space.
- A personal best $f(\overrightarrow{B}(i))$ value indicating the best solution of itself.
- A global best $f(G(i))$ solution representing the best solution of the entire swarm.

  Every generation shows assessment of each particle, with further stochastic modification of $v_{(i,k)}$ in the movement of particle $\overrightarrow{P}(i)$'s earlier best position and also on the basis of past best position of other neighboring particle. With every iteration, we evaluate each particle and then adjust $v(i, k)$ in the direction of particle $\overrightarrow{P}(i)$'s preceding best position and also neighboring particle existing best position (Correa et al. 2006).

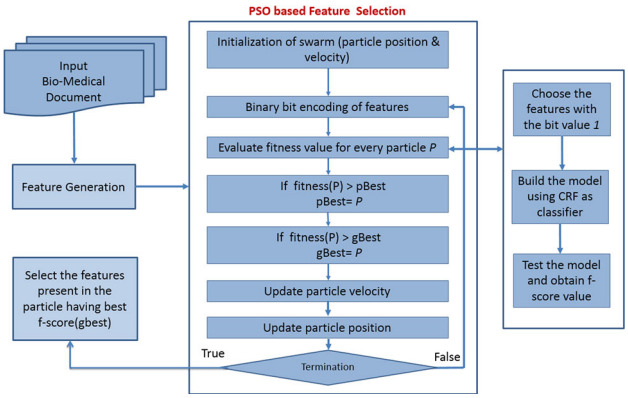

Fig. 2 Proposed system architecture: PSO-based feature selection

4 Proposed feature selection technique

In this section, we propose our technique of feature selection. The set of features is encoded in a vector. Initially, the vector is randomly initialized with the values denoting the presence and absence of features. The robust classifier, CRF, is trained with the feature combination represented by the vector and evaluated on the validation set.[1] The F-score value, which is used as a fitness function, is computed for every vector. The F-score value computed at every iteration is used for updating the features in the subsequent iterations. Finally, we obtain a set of optimized feature set. These features are finally used for final evaluation on the test set. Figure 3 provides insight to the proposed feature selection technique for single iteration. The methodology of our proposed work is presented in the form of flowchart in Fig. 2.

4.1 Conditional random field: base classifier

Conditional random fields (CRF) (Lafferty et al. 2001) are probabilistic models for specifying and classifying sequential data. A CRF is a form of undirected graphical model that defines a single log-linear distribution over the label sequences given a particular observation sequence. As the named entities (NEs) in text appear as a sequence of words, for example, in the sentence:

'Analysis of myeloid-associated genes in human hematopoietic progenitor cells.' Our task is to identify NEs and classify them into some predefined categories of interest. In the above sentence, NEs are '*myeloid-associated*' which belong to class 'DNA' and '*human hematopoietic progenitor cells*' which belongs to class 'Cell_type'. Some of the well-known sequence labeling classifiers are hidden Markov model (HMM) (Rabiner and Juang 1993), maximum entropy Markov model (MEMM) (Berger et al. 1996) and conditional random field (CRF). Literature shows the success of CRF classifier (Ekbal and Saha 2013; Kuo et al. 2007; Klinger et al. 2007) for NE extraction. This has motivated us to use CRF as the base classifier. CRF calculates the conditional probability of label state $Y = \langle y_1, y_2, \ldots, y_T \rangle$ for the given observation sequence $X = \langle x_1, x_2, \ldots, x_T \rangle$. For JNLPBA data, the labels are as follows: $Y = \{$*B-Protein, I-Protein, B-DNA, I-DNA, B-RNA, I-RNA, B-cell_type, I-cell_type, B-cell_line, I-cell_line, O*$\}$. With this given labeling, our example sentence looks like as shown in Table 2. The conditional probability is calculated as:

$$P(y_1, y_2, \ldots, y_T|X) = \frac{1}{Z_x} \prod_i (\xi_i(Y_i, X)\xi_i'(Y_i, Y_{i-1}, X)) \quad (1)$$

[1] A part of training set is used as validation set. We have divided the original training set into two sets: validation set and new training set.

(a)

Token	Feature1	Feature2	Feature3	Feature4	Feature5
Th2	NN	B-NP	T	1	0
type	NNS	I-NP	t	0	0
cytokines	VBD	I-NP	c	0	1
in	IN	B-PP	i	0	0
hepatitis	NN	B-NP	h	0	1
B	NN	I-NP	B	1	1

Sample Token with their features

Particles	Feature1	Feature2	Feature3	Feature4	Feature5
P1	0	1	0	0	1
P2	1	1	1	0	0
P3	1	0	0	1	1
P4	1	0	0	1	0
P5	0	0	0	1	1

Particles position initializations

(b)

Particle	Feature1	Feature2	Feature3	Feature4	Feature5	Fitness vale	Best position
P1	0	1	0	0	1	62.50	62.50
P2	1	1	1	0	0	68.12	68.12
P3	1	0	0	1	1	75.16	75.16
P4	1	0	0	1	0	72.52	72.52
P5	0	0	0	1	1	65.00	65.00

Optimization process after *iteration-i*
Global Best position value= 75.16

Particle	Feature1	Feature2	Feature3	Feature4	Feature5	Fitness value	Best position
P1	1	1	1	0	1	60.21	62.50
P2	1	0	0	0	0	75.21	75.21
P3	1	0	1	1	1	74.12	75.16
P4	1	0	0	1	1	76.54	76.54
P5	1	1	0	1	1	68.10	68.10

Optimization process after *iteration- (i+1)*
Global Best position value= 76.54

Fig. 3 PSO optimization process: **a** A sample token with some of the features to illustrate the process of selecting features at each iteration through PSO algorithm. Here the value of 1 represents that feature is selected and 0 represents that the feature is pruned. **b** A random initialization of particle position after iteration i and next iteration $(i + 1)$, *shadow row* denotes the fittest particles in that iteration. Global best position value after each iteration denotes the global best solution obtained. All fitness values are hypothetical

Table 2 Illustration of CRF labeling for feature function

i	y_i	y_{i-1}	x_i
1	O	–	Analysis
2	O	O	of
3	B-DNA	O	myeloid
4	I-DNA	B-DNA	-
5	I-DNA	I-DNA	associated
6	O	I-DNA	genes
7	O	O	in
8	B-cell_type	O	human
9	I-cell_type	B-cell_type	hematopoietic
10	I-cell_type	I-cell_type	progenitor
11	I-cell_type	I-cell_type	cells

where ξ_i and $\xi_i^{'}$ are defined as follows:

$$\xi_i(Y_i, X) = \exp\left(\sum_k \eta_k s_k(y_i, x, i)\right) \tag{2}$$

$$\xi_i^{'}(Y_i, Y_{i-1}, X) = \exp\left(\sum_j \lambda_j t_j(y_i, y_{i-1}, x, i)\right) \tag{3}$$

where t_j and s_k are transition feature function and state feature function, respectively. The transition feature function t_j depends upon the current label y_i , previous label y_{i-1} and observation sequence x at time i. The state feature function is the function of current label y_i and observation sequence x at time i. Parameters λ_j and η_k are to be estimated from training data. A real-valued features $g(x, i)$ is to be define while making either of two feature function. This feature $g(x, i)$ of observation sequence x expresses some characteristic of the empirical distribution of the training data that should also hold of the model distribution. An example of such a feature is

$$g(x, i) = \begin{cases} 1 & \text{if the observation word at position } i \\ & \quad \text{is 'hematopoietic'} \\ 0 & \text{otherwise} \end{cases}$$

The value of one of these real-valued observation features $g(x, i)$ has been assigned to each feature function. Both feature functions are therefore real-valued. For example, consider the following transition function:

$$t_j(y_i, y_{i-1}, x, i) = \begin{cases} g(x, i) & \text{if } y_{i-1} = \textit{B-cell\_type} \text{ and } y_i \\ & \quad = \textit{I-cell\_type} \\ 0 & \text{otherwise} \end{cases}$$

For ease of notation, we combined both state feature function and transition feature function into a single feature function $f_j(y_i, y_{i-1}, x, i)$. If there are n such feature function then global feature function $F_j(Y, X)$ can be written as:

$$F_j(Y, X) = \sum_{i=1}^{n} f_j(y_i, y_{i-1}, x, i) \tag{4}$$

By using Eq. (4), we can rewrite conditional probability of state sequence Y given observation sequence X is as follows:

$$P(Y|X, \lambda) = \frac{1}{Z_x} \exp\left(\sum_j \lambda_j F_j(Y, X)\right) \tag{5}$$

The normalization term, Z_x, is determined by computing the above sum for all possible label sequences. There are various methods used to train CRF, and they differ only in the objective function which is optimized (maximized/minimized). One of the method is penalized log-likelihood. The log-likelihood $\tilde{L}$ is computed by summing the log-probabilities for a fixed set of weights $\Lambda = \{\lambda_1, \lambda_2 \ldots \lambda_T\}$, over whole training instances in dataset D. The penalized log-likelihood over whole training instances D is given by

$$\tilde{L}_\Lambda(D) = \sum_{Y,X \in D} \log P_\Lambda(Y|X) - \frac{||\Lambda||^2}{2\sigma^2} \tag{6}$$

σ^2 is 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 (Shanno 1970), a quasi-Newton method that is significantly more efficient.

4.2 Feature encoding

The features are encoded using binary-bit values. Length of the vector is decided by the total number of features available. There are many variants of the standard PSO, and binary PSO is one such version. In binary PSO, a potential solution to a problem is represented by a particle $\overrightarrow{P}(i) = (p(i, 1), p(i, 2), \ldots, p(i, n))$ in an n-dimensional search space, where $p(i, k) \in (0, 1), i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, n$. Here, 'N' represents the total number of particles or potential solutions in a given swarm and 'n' represents the total number of features. Thus the length of each particle is n. Features being selected are distinguished on the values of 0 and 1.

For a set of features 'F', $F = \{f_1, f_2, f_3, f_4\}$, the first element of $\overrightarrow{P}(i)$ corresponds to the first feature (f_1). Similarly, $\overrightarrow{P}(i)$ is the set of total 4 features where f_2 represents the second feature, f_3 is the third feature and so on. Each feature can have value either 0 or 1. Feature with value '1' denotes that corresponding feature takes part in training. The

Table 3 Feature encoding

Particle	Binary feature encoding
$\overrightarrow{P}(1)$	(1, 0, 0, 1, 1)
$\overrightarrow{P}(2)$	(1, 1, 1, 0, 0)
$\overrightarrow{P}(3)$	(0, 1, 0, 1, 0)
$\overrightarrow{P}(4)$	(1, 0, 1, 1, 1)

value of '0' represents that respective feature is not considered further for training. For example, for the feature list $F = \{f_1, f_2, f_3, f_4, f_5\}$ and $N = 4$, the swarm can be represented as shown in Table 3.

Here, for the particle $\overrightarrow{P}(1) = F = \{1, 0, 0, 1, 1\}$, the first position corresponds to 1 which means feature f_1 is present and is selected for training. Second and third bit positions are 0 which represent that f_2, f_3 are not selected for training. The bit positions, fourth and fifth are set to 1 which correspond to the selection of f_4 and f_5. It represents the solution where features f_1, f_4 and f_5 are the only selected features.

The proposed approach based on binary PSO is carried out in the following three steps.

(1) Setting up the initial population;
(2) Updation of the global best position and best position of particles;
(3) Updation of velocity vector; and
(4) Sampling of the new particles.

4.3 Setting up the initial population

The initial population is set randomly for the N binary strings, each having length n. Each particle $\overrightarrow{P}(i)$ is individually produced in the following way. At each position $p(i, k)$ of $\overrightarrow{P}(i)$, an uniform random number $\oint$ is generated between the range (0, 1). For example, if the midvalue is selected to be 0.5, we can set the value of $p_{(i,k)} = 1$ for $\oint < 0.5$ else $p_{(i,k)} = 0$.

4.4 Updation of the global and best position values

The best position of any particle $\overrightarrow{P}(i)$ is denoted by $\overrightarrow{B}(i)$ which is initialized by null value. After the generation of initial particles, we set the value of $\overrightarrow{B}(i)$ to the position vector of the particle $\overrightarrow{P}(i)$. Updation of the 'pBest' occurs in every iteration if it satisfies some condition. Initially, the fitness value for each particle is set to null. Best position is updated in case of improvement in the fitness value of the new position value over previous; otherwise, the value remains unchanged, i.e., when value of $f(\overrightarrow{P}(i))$ exceeds by the value of $f(\overrightarrow{B}(i)$. Same procedure is followed for the updating the 'gBest'. This updation is done only after we retrieve all the best positions 'pBest' values. The 'gBest' is initialized to null value, and it is updated only when the fitness function value $f(\overrightarrow{B}(i))$ of the swarm is better than $f(\overrightarrow{G}(i))$. The value of the global best position vector is not updated if it fails to satisfy this condition.

4.5 Updating velocity

Updation of velocity helps the particles to fly around the search space so that these eventually move closer to the target solution. Each particle has its own velocity vector. At the beginning, we set the value of the velocity vector $\overrightarrow{V}(i) = (v_{(i,1)}, v_{(i,2)}, \ldots, v_{(i,n)})$ randomly. Updation of the velocity and position of each particle is done by the following equation:

$$v_{(i,k)} = \omega v_{(i,k)} + \phi_1(b_{(i,k)} - p_{(i,k)}) + \phi_2(g_{(k)} - p_{(i,k)}) \tag{7}$$

Here, $w(0 < w < 1)$, specifies the inertia weight which was included in Eq. (7) to control the velocity explosion, is a constant value set by user according to their problem specifications. ϕ_1 and ϕ_2 represent the learning parameters and constant value specified by the user, respectively. The velocity is updated as per Eq. (7). Inertia weight keeps track of important information about the path that a particle follows. The value of inertia weight is set according to the problem definition in order to obtain good solutions. Exploration and exploitation of the search space can be controlled using the inertia weight. If the value of $w \geq 1$, the velocity increases over the time which leads to the divergence in swarm. Particles decelerate if w is set as $0 < w < 1$. In this setting, convergence depends on the value of learning parameter (ϕ_1, ϕ_2). Selection of negative value has no impact on the performance of binary PSO as setting this term may not lead to any effect in giving the next path to the particles.

4.6 Selection of new particles

Selection of new particles is quite similar to that of standard PSO with the minor modification. Velocity remains continuous while position is updated using the velocity which is set according to the mathematical expression listed below.

$$p_{(i,k)} = \begin{cases} 1 & \text{if (random} < S(v_{(i,k)})) \\ 0 & \text{otherwise} \end{cases}$$

where $0 \leq \text{random} \leq 1$ is an uniform random number.

$$S(v_{(i,k)}) = \frac{1}{1 + \exp(-\overrightarrow{v}_{(i,k)})}$$

This represents the sigmoid function. Thus, we set the value of 0 and 1 on the basis of the values of the velocity (Correa et al. 2006).

4.7 Algorithm: at a glance

1. The initial population is set randomly for the N binary strings having the length n (where n denotes the available set of features). Position of a particle $\overrightarrow{P}(i)$, is set to {0, 1} on the basis of the midvalue specified:

$$p_{(i,k)} = \begin{cases} 1 & \text{if(random} > \text{mid)} \\ 0 & \text{otherwise} \end{cases}$$

2. Evaluation of each particle $\overrightarrow{P}(i)$ is done by calculating its fitness function $f(\overrightarrow{P}(i))$. Initially, the particle best position and global best position are initialized to empty.
3. The 'pBest' value is updated if the value of $f(\overrightarrow{P}(i))$ is greater than $f(\overrightarrow{B}(i))$.
4. The 'gBest' value is updated only when the fitness function $f(\overrightarrow{B}(i))$ in the swarm is superior than $f(\overrightarrow{G}(i))$.
5. At the beginning, we set the value of the velocity vector $\overrightarrow{V}(i) = (v_{(i,1)}, v_{(i,2)}, \ldots, v_{(i,n)})$ randomly. Position and velocity of each particle are updated with every iteration using following equation:

$$v_{(i,k)} = \omega v_{(i,k)} + \phi_1(b_{(i,k)} - p_{(i,k)}) + \phi_2(g_{(k)} - p_{(i,k)}) \tag{8}$$

6. Selection of new particle is done on the basis of position which is updated using the velocity that is set according to the following mathematical expression:

$$p_{(i,k)} = \begin{cases} 1 & \text{if (random} < S(v_{(i,k)})) \\ 0 & \text{otherwise} \end{cases}$$

where $0 \leq \text{random} \leq 1$ is an uniform random number.

$$S(v_{(i,k)}) = \frac{1}{1 + \exp(-\overrightarrow{v}_{(i,k)})}$$

This represents the sigmoid function. Thus, we set the value of 0 or 1 on the basis of the value of velocity.

7. Repeat step 2 until convergence.

5 Features for entity extraction

Domain-independent feature set is used to build our model using CRF. Below we have discussed in detail the features used for extracting named entities from biomedical domain. Many of these features were motivated from the prior works such as (Ekbal and Saha 2013) (Table 3).

Table 4 Illustration of input for local context feature, for the current token *glucose*

Token	Feature-1	Feature-2		
Large	JJ	B-NP		
number	PRP	B-NP		
of	VBZ	B-VP		
glucose	DT	B-NP	≫	CURRENT TOKEN
in	JJ	I-NP		
enzyme	NN	I-NP		
constitute	VBZ	B-NP		

Table 5 Illustration of obtained local context feature, for the current token *glucose*

Token	Feature-1	Feature-2
number	PRP	B-NP
of	VBZ	B-VP
glucose	DT	B-NP
in	JJ	I-NP
enzyme	NN	I-NP

1. **Local context** Local context refers to the tokens which appear in the surrounding of the target token. Context can be represented mathematically as: $w_{i-1}^{i+1} = w^{i-1} \ldots w^{i+1}$ where w_i represents the current word. In our work, we consider the context in the range of w_{i-5}^{i+5}. Here in given Table 4 the column position one represents the token and the second and third represent the attributes. If the context window size is selected to be in the range of w_{i-2}^{i+2}, then from the example given in Table 4, following tokens or words would be selected as the context word reported here in Table 5.
2. **Word prefixes and suffixes** These refer to the fixed length character sequences removed either from the left or rightmost positions of the words. For example, the suffix and prefix of word 'Number' are- N, Nu, Num, Numb and r, er, ber, mber, respectively.
3. **Word length** The words which are relatively shorter in length have less chances of being a NE. We define a binary feature that triggers the value 1 if word length is greater than the threshold being specified by the user, otherwise it is set to 0. Here we consider the threshold length to be 5.
4. **Infrequent words** It has been observed that words which are more frequent does not come under the list of named entities. Taking this point into consideration, we compile a dictionary from the given training data and include those words that have the frequency of occurrences less than 10 times. We define a binary feature that triggers the value 1 if the token is there in the dic-

tionary or 0 if not present. This threshold is declared on the basis of the size of the dataset.

5. **Part-of-speech(PoS) information** Part of speech (PoS) is very vital feature for identifying NEs. This provides useful evidence that helps detecting important grammatical properties. Words are assigned the same PoS if they have same syntactic behavior. Here PoS information of the current and/or the surrounding tokens are used as features. The PoS information was extracted from GENIA tagger[2] V2.0.2.
6. **Chunk information** Chunk information is extracted from GENIA tagger v2.0.2 corpus. This feature is highly beneficial in boundary identification for biomedical entities. Here, we consider the chunk information of the present and neighboring tokens.
7. **Unknown token feature** The binary feature is defined that is triggered to value 1 if the target token in the test data was present in the training data, else we set this value to be 0. We randomly set the value of this feature in case of classifier training.
8. **Word normalization** Two distinct features have been selected for word normalization. The first feature is defined that is used to tackle the words having the plural form, hyphen, verb, digit and alphanumeric letters. This feature converts the word to its root form. The other feature specifies the orthographic construction of the target words. Word shape is defined as the mapping of each word to the equivalent class. In order to implement this feature, we normalize the words by converting every capital character by 'A', and small character to 'a'. We reduce every digit by '0'. For example, if we consider the token 'Ly-49', the normalized word for this token would be 'Aa-00'.
9. **Word-class feature** Word-class feature is implemented taking into the consideration that some kind of entities, which reside in the same class, are identical to each other. Here also similar to the word normalization feature, we convert the capital letters to 'A', small letters by 'a', numbers to 'O' and non-English characters to '-', respectively. Further after this conversion, we squeeze the consecutive characters into single character. For example, the word-class feature for the token 'IL-2-mediated' is 'AA-O-aaaaaaaa', which is further reduced to 'A-O-a'.
10. **Head nouns** Head noun in a noun phrase often provides useful evidence in classifying NEs. Head noun describes the function of NE. For example, 'IL-2-mediated' is a noun phrase where the term 'mediated' represents a head noun. From the NLPBA training set, we extracted a list of 912 head nouns. We consider only the most frequently occurring head nouns.
11. **Verb trigger** The verbs that appear in the surrounding context of NE provide useful information for classifying the target entity. We extract the most frequently occurring verbs from the training set and use these to define a binary-valued feature.
12. **Informative words** Informative are the words that help in identifying NEs. Named entity includes the words that are, in general, very long and contains many common words and/or symbols inside it. These include the words like functional words and nominal words that occur most often in the training data, but do not help in NE classification. On the other hand, many words, that appear either as part of NE or outside NE, could be effective in NE identification. We first generate a list of words that occur within the multiword NE. As digits and symbols are not helpful in NE identification, we eliminate these from the list. In order to check how good is the word to identify the NEs, we assign them some weights. Some highly frequent informative words are listed here from GENIA datasets.

IL-2, gene, NF-kappa, B, receptor, T, cell, primary, lymphocytes, complex The list of word is generated that occur in the multiword NE. To identify which word is more prominent in identification of the NEs, weight is assigned. The $\mathrm{NE}_{\mathrm{wt}}(t_i)$ is calculated as follows:

$$\mathrm{NE}_{\mathrm{wt}}(t_i) = \frac{\text{Total no. of occurrences of } t_i \text{ as part of NE}}{\text{Total no. of occurrences of } t_i \text{ in the training data}} \quad (9)$$

The words, whose frequency of appearing as the part of NE in the training set is more than two, are considered to be as informative. The others left words are classified into the following classes:

(a) **Class 1** Words whose frequency is greater than 100 and their $\mathrm{NE}_{\mathrm{wt}}(t_i) > 0.4$.
(b) **Class 2** Words appearing in the range between 20 and 99 and their $\mathrm{NE}_{\mathrm{wt}}(t_i) \geq 0.6$ are categorized in this class.
(c) **Class 3** This class includes the words that have occurred in the range between 10 and 19 with their $\mathrm{NE}_{\mathrm{wt}}(t_i) \geq 0.85$.
(d) **Class 4** Words having the occurrences $\geq$5 and $<$10 with their $\mathrm{NE}_{\mathrm{wt}}(t_i) \geq 0.90$ belong to this class.
(e) **Class 5** Words that have occurred less than 5 times with their $\mathrm{NE}_{\mathrm{wt}}(t_i)$ lying between 0.9 and 1 are included in this class.

The binary feature vector of length 5 (with respect to class) is generated. The feature value '1' is set for the particular class if the target word belongs to any of the above-mentioned classes. If it does not belong to any of the class, the value is set to '0'.

[2] http://www.nactem.ac.uk/GENIA/tagger/.

Table 6 Orthographic feature set with example

Feature	Example	Feature	Example
InitCap	Number	AllCaps	IL
InCap	mRNA	CapMixAlpha	AbCa
DigitOnly	1, 10	DigitSpecial	10-0
DigitAlpha	IL-10	AlphaDigitAlpha	IL10RNA
Hyphen	–	CapLowAlpha	Abcd
CapsAndDigit	32Receptor10	RomanNumeral	I, II
StopWord	in, of, at	ATGCSeq	ATAAG, CGCCA
AlphaDigit	m60, p66	DigitCommonDigit	1, 50
GreekLetter	α, β	LowMixAlpha	mBc, mRNA

13. **Content words in surrounding contexts** In order to explore the global context information, this semantic feature was used in our work. We include all those unigrams that occurred within the context window size(w_{i-3}^{i+3}) for w_i(words) from training data. We changed the token to the lowercase and eliminated special symbols, punctuations, numbers and stopwords. By using the 10 most frequent content words, we defined a feature vector of length 10. Given a classification instance, the feature related to token is set to 1 if w_i occurs within the context window size. In the GENETAG test set, we defined this feature using NE output predicted by the GENIA tagger. The obtained features are found to be effective.
14. **Orthographic features** We implement various orthographic features that consider capitalization and digit information. In the biomedical data, usage of the special characters like (‘,’, ‘-’, ‘. ’, ‘_’) is very common. These symbols are very helpful in entity extraction. These features are listed in Table 6. Some symbols like ‘,’ are very useful in detecting the boundaries of the NEs. Some features are also defined to examine the existence of ATGC sequence and stop words. These features defined above are the binary features that trigger the value of 1 when a particular feature is satisfied. We list all the features with example in Table 6.

6 Datasets and experiments

This section describes the datasets that we have used for our experiments, mentions about the assessment scheme, reports the experimental results and presents detailed analysis along with comparisons to the existing systems. To carry out our experiment, Conditional Random Field (CRF) (Lafferty et al. 2001) was used as the base classifier. We used `C++` based `CRF++` package[3] for our implementation with default parameter setting.

[3] https://taku910.github.io/crfpp/.

6.1 Evaluation scheme

For the assessment of classifier’s performance, standard metrics such as recall, precision and F-measure are used for our system evaluation. Precision is defined as the fraction of retrieved NEs that are relevant to the total retrieved NE chunks. Recall is defined as the fraction of accurately predicted NE chunks to the actual NE in labeled data.

$$\text{Precision} = \frac{\text{Number of correctly identified NE}}{\text{Total number of NE predicted by the system}} \tag{10}$$

$$\text{Recall} = \frac{\text{Number of accurately predicted NE}}{\text{Number of actual NE in labeled data}} \tag{11}$$

F-measure is the harmonic mean of precision and recall and is defined as follows:

$$F_\alpha = \frac{(1+\alpha^2)(\text{recall} * \text{precision})}{\alpha^2 * \text{precision} + \text{recall}}, \quad \alpha = 1 \tag{12}$$

Here α is a positive real value. The traditional F-measure (F1 score) is the harmonic mean of precision and recall. For evaluation of GENIA, GENETAG, AIMed we use the script, which was made available for the JNLPBA 2004 shared task. [4] For BC-II, we used the script provided by BioCreative shared task.

6.2 Datasets

We evaluate our system on four datasets, namely GENIA[5], AIMed[6], GENETAG[7] and BioCreative-II(BC-II) gene mention[8] challenge datasets. These datasets were created following different annotation schemes. GENIA version 3.02 corpus was developed by controlled exploration on MEDLINE using the MeSH terms such as ‘*human*’, ‘*blood cells*’

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

[5] http://www.geniaproject.org/shared-tasks/bionlp-jnlpba-shared-task-2004

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

[7] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENETAG.tar.gz.

[8] http://biocreative.sourceforge.net/biocreative_2_dataset.html.

and '*transcription factors*'. Total 2000 abstracts of around 500K wordforms were selected from this search. On the basis of chemical classification, 48 classes having small taxonomy were selected. To define GENIA corpus 36 classes were selected among the above set of classes. This dataset was further reduced to be trained with five NE classes, namely *Protein*, *DNA*, *RNA*, *Cell_line* and *Cell_type*. Test dataset compromises of 404 abstracts having 100K words.

AIMed corpus is generated from 197 abstracts extracted from the Database of Interacting Protein (DIP) and 28 abstracts which contain protein names but do not contain any interaction-related information with the total of 1944 sentences. It has been viewed that AIMed dataset is highly imbalanced as this is heavily biased toward the negative examples.

GENETAG dataset is contained in the dataset of 'MedTag'. The dataset compromises of correct and incorrect gene or names of protein in different contexts. The dataset consists of total 20,000 sentences having gene/protein names. The aim was to identify the 'NEWGENE' term (denoting gene names) in medical abstracts. For building the system, we use 7500 labeled sentences for training and 2500 sentences are used for validation. Testing was done on 5000 unlabeled sentences. In this dataset, the sentences which are more informative (in terms of NE) were included, while other sentences were discarded. GENIA and GENETAG are very similar to each other as compared to AIMed. In the dataset of GENIA and GENETAG, longer text fragments are selected as the entity references. Both GENIA and GENETAG consist of the semantic category word 'protein' for protein annotations. In order to properly denote the boundaries of NE, we use the IOB2[9] encoding scheme.

BioCreative II dataset on gene mention recognition is also used to carry out our experiments. Dataset consists of the sentences from the MEDLINE abstracts which are annotated manually for the gene mentions. The dataset is built by including the abstracts consisting of both the gene names as well as the abstracts which do not contain any gene names. The dataset comprises of 15,000 sentences, out of which we take only 2500 sentences as the development data. As a test set, we use 5000 sentences. For our experimental analysis, we use the same training, development and test sets as provided by the respective shared task organizer. Therefore, the comparison that we present here is fair.

6.3 Results and analysis

We develop three baselines to compare our proposed technique for each dataset. For all of our baseline, we used CRF, as a base classifier to train our model.

[9] I, O and B represent the intermediate, outside and beginning token of a NE.

- **Baseline 1** This baseline model is trained using the complete set of features as described in Sect. 5.
- **Baseline 2** This baseline model is trained on the features selected using correlation-based feature selection technique (filter-based model).
- **Baseline 3** In this baseline model, classifier is trained on the feature selected using recursive feature elimination algorithm (Guyon et al. 2002) (wrapper-based model).

We apply our proposed technique to identify the significant set of features, and the corresponding results are reported in Table 7. This shows that PSO-based feature selection technique achieve better performance for each domain.

We also made an interesting observation that the models developed using the pruned feature sets achieve better accuracies on all the datasets. Results that we obtain through PSO-based feature selection are also better than the baseline 2 and 3. We identified algorithm selected only 28, 29, 25 and 22 features out of 57 features on GENIA, GENETAG, AIMed and BioCreative-II dataset, respectively.

In order to examine the statistical significance of the obtained results, we perform analysis of variance (ANOVA) (Shapiro and Wilk 1965) test. It has been observed that differences between the proposed approach and the baselines in terms of F-measure are statistically significant as p value is less than 0.05.

6.4 Sensitivity analysis of PSO parameters

We perform several experiments using different parameter settings of PSO, and thereby, we provide a thorough sensitivity analysis for determining the optimal parameter settings of PSO. Shi and Eberhart (1998), authors have set the parameters ω as 0.7298 and $\phi_1 = \phi_2$ as 1.49618. Pedersen (2010) authors have suggested an unique way of obtaining the different values of parameters on the basis of the optimization scenarios. On the detailed studies of prior works (Shi and Eberhart 2001; Alatas and Akin 2008) and after performing several experiments with different parameter settings on the validation set, finally we set the different parameter values of PSO. Results using the best five parameter combinations on validation dataset are shown in Table 8. We name these as PSO-1, PSO-2 and so on. We fix the swarm(population) size as 20 particles and the number of iterations as 100 for all our experiments. For the GENIA data set, the best result is obtained for the following parameter combination: inertia weight, learning parameter-I and learning parameter-II to '0.7298' ,'1.49618', '1.49618', respectively. For GENETAG dataset, the highest accuracies were reported by setting inertia weight to '0.3925', learning parameter-I to '2.5586' and learning parameter-II to '1.3358'. Similarly for AIMED dataset, the

Table 7 Results of baseline and PSO-based feature selection

Model	Objective dataset	GENIA	GENETAG	AIMed	BC-II
Baseline-1	No. of features	57	57	57	57
	Recall	70.19	95.70	89.23	81.64
	Precision	67.31	80.83	87.64	82.67
	F-score	68.72	87.64	87.98	82.15
Baseline-2	No. of features	30	24	28	24
	Recall	66.80	91.95	95.58	79.24
	Precision	75.88	89.29	83.07	91.14
	F-score	71.05	90.60	88.89	84.78
Baseline-3	No. of features	23	37	25	25
	Recall	72.14	93.68	89.98	85.47
	Precision	73.27	86.19	86.19	86.52
	F-score	72.70	89.77	88.04	85.99
CRF[PSO]	No. of features	28	29	25	22
	Recall	77.25	96.00	90.73	94.79
	Precision	73.37	90.28	89.46	84.08
	F-score	75.26	93.05	90.09	89.11

Table 8 Results of PSO-based feature selection with different parameter settings

PSO-RUN	Parameter settings			GENIA	GENETAG	AIMed	BC-II
	Inertia weight	ϕ_1	ϕ_2	F-score	F-score	F-score	F-score
PSO-1	0.7298	1.49618	1.49618	75.26	92.70	89.97	88.10
PSO-2	0.3925	2.5586	1.3358	74.10	92.78	90.09	89.11
PSO-3	−0.4349	−0.6504	2.2073	73.90	92.35	89.96	87.56
PSO-4	0.4091	2.1304	1.0575	73.80	92.78	90.09	87.70
PSO-5	−0.3593	−0.7238	2.0289	74.50	93.05	90.03	88.55

best parameter combinations are inertia weight = '0.4091', learning parameter-I = '2.1304' and learning parameter-II = '1.0575'(Fig. 4).

The best accuracy is reported on BC-II dataset with the setting of the following PSO parameters: inertia weight to '0.3925', learning parameter-I to '2.5586' and learning parameter-II to '1.3358'. Learning parameter-I & II are known as self-learning parameter and global learning parameter, respectively. These are basically the acceleration coefficients that control self best position and global best positions of particle, respectively. The optimal parameter setting varies across the datasets. The factors contributed to this are the variations in the dataset size and the dimensionality of the problem (Pedersen 2010). We observe that with the increase in the number of iterations, F-measure value also increases for all the four datasets. This is because with every iteration the swarm becomes more intelligent with the knowledge of the global best position and self best position, and it converges to the best solution. Figure 5 illustrates this property. We show the learning curves in Fig. 6 by performing experiments with the different sized training samples.

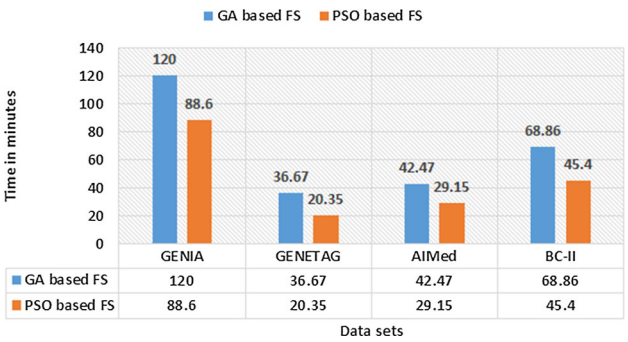

Fig. 4 The Comparison of single iteration execution time of GA and PSO based feature selection on population (particle) size 10 utilizing CRF as base classifier

6.5 Analysis of feature combination

The selected features using PSO technique for different datasets are reported in Table 9. It has been observed that for GENIA dataset, the features that have not been selected are dynamic NE tags, word length and infrequent word. As context feature, the system is able to select 2 features out of

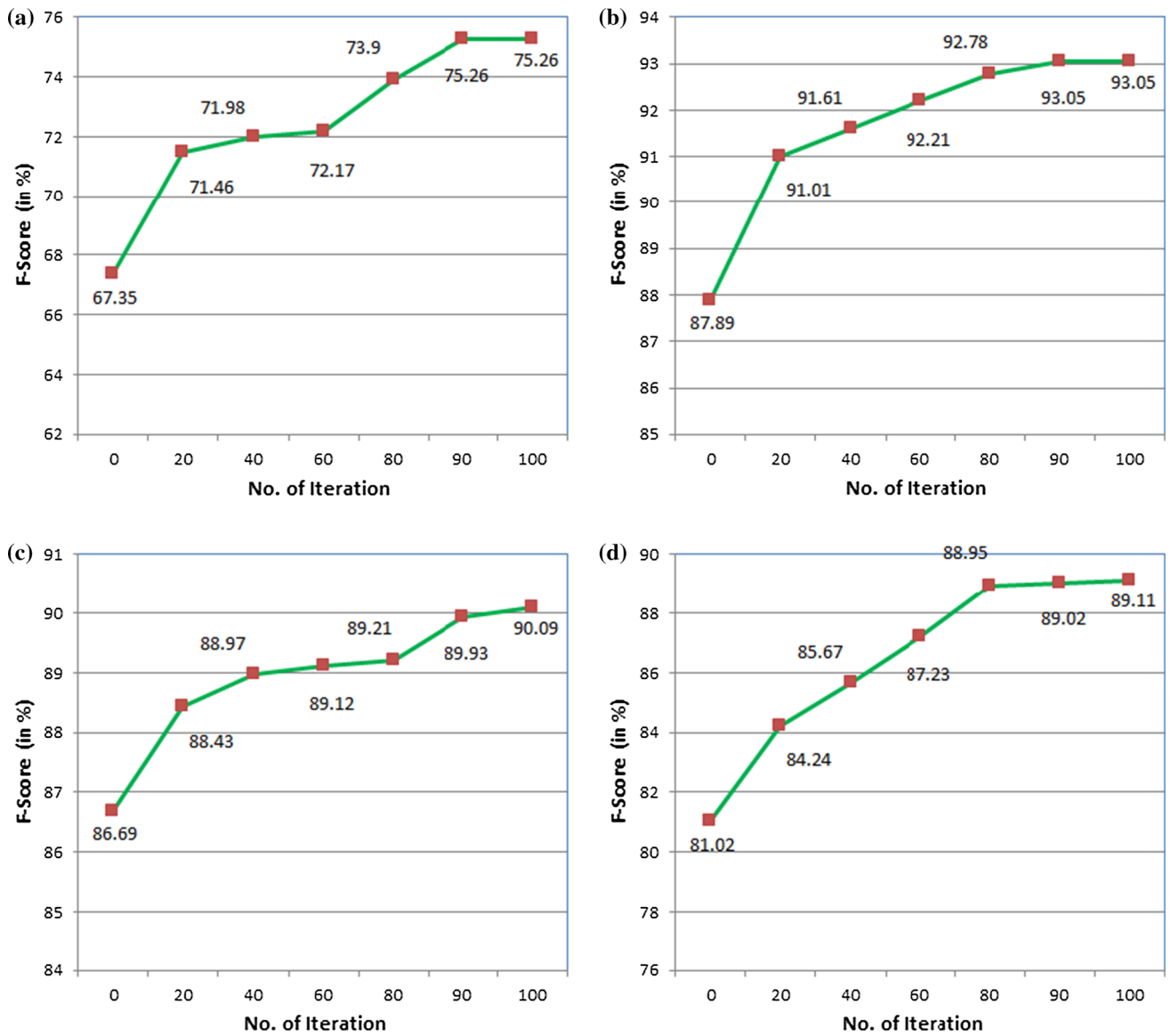

Fig. 5 Variations in F-score values obtained by the proposed PSO-based feature selection technique with the increase in the number of iterations. **a** GENIA, **b** GENETAG, **c** AIMED, **d** BC-II datasets

total 6. In content features of length 10, only 6 are selected. While in the orthographic features of length 17, the system has selected 10 features. The PSO-based feature selection is able to select 1 feature from prefix, 2 features from suffix feature set, 2 features from word normalization feature and 1 feature from the informative word feature.

For the GENETAG dataset, only word length and infrequent words have not been selected fully. While the system is able to select 3 context features from total of 6 features, 6 features from content features of length 10, 9 features from total 17 orthographic features, 2 prefix features, 2 suffix features, 1 feature of word normalization, 1 feature from informative word have been selected.

For AIMed dataset, prefix, informative words, dynamic NE tags, infrequent words features have not been selected. Only 5 features have been selected from context features of length 6, 5 out of 10 features from content feature, 7 orthographic features out of 17 features, 2 suffix features out of 4 have been selected and one feature from word normalization feature have been selected.

For BC-II dataset, we observed that prefix, informative words, dynamic NE tags, word length, infrequent word, verb trigger feature have not been selected, while 3 context features, 5 content features, 9 orthographic, 1 suffix, 1 word normalization have not been selected by the system.

We have observed that head noun, verb trigger, chunk information have been selected in the optimal feature set

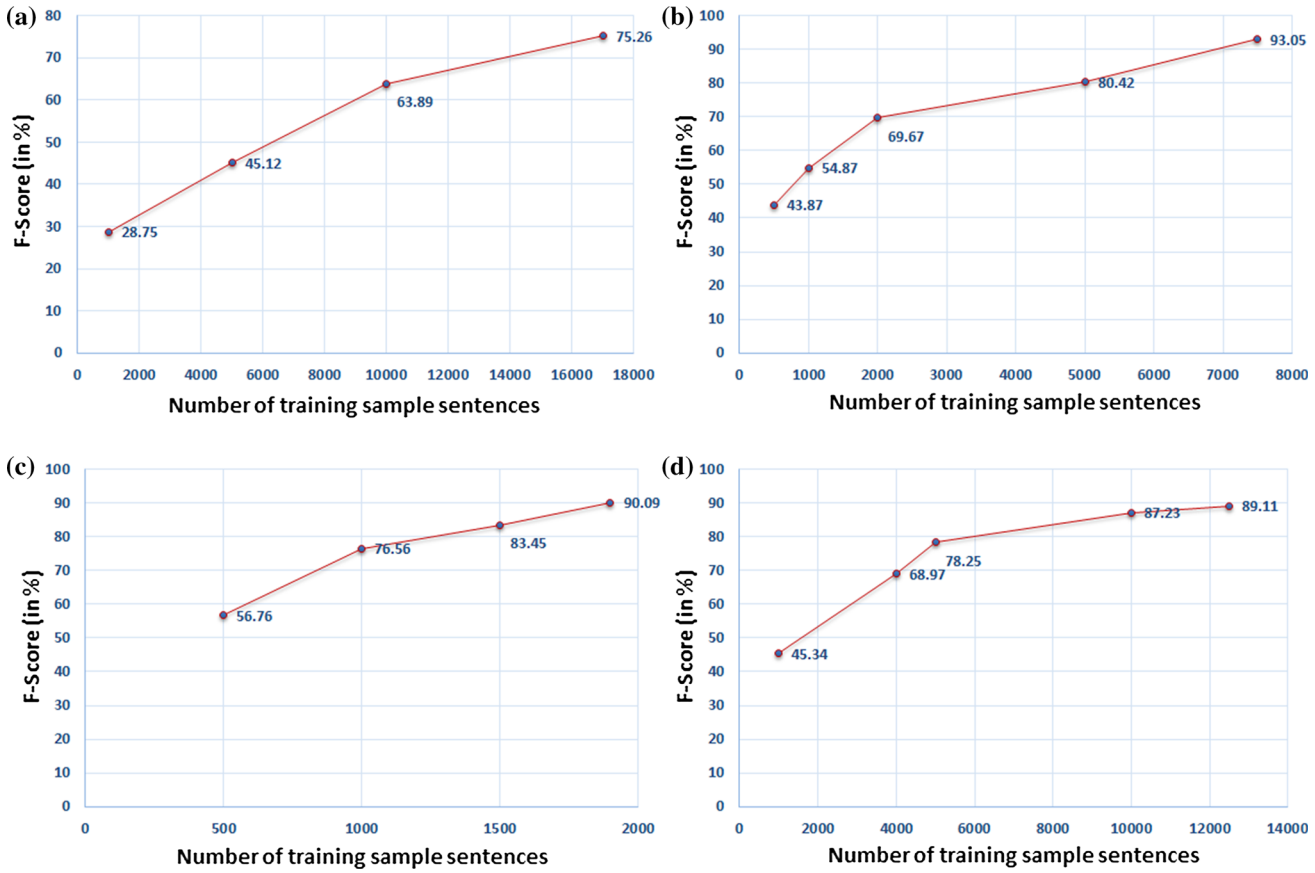

Fig. 6 Variations in F-score values obtained by the proposed approach with the increase in the number of training samples for different biomedical datasets. **a** GENIA, **b** GENETAG, **c** AIMed, **d** BC-II. The trend shows that with the more number of training samples we get higher F-score

for all the four datasets because with the inclusion of these features, model performance has been improved. As PSO guarantees to determine the optimal set of solutions, all the good features via global best solutions have been selected.
We have also evaluated the models for more detailed analysis that considers the left and right boundary matching of each entity. These results are reported in Tables 10, 11, and 12 for GENIA, GENETAG and AIMed datasets, respectively. The precision, recall and F-measure are evaluated with respect to the following three types:

1. *FULLY correct* It specifies the condition when the boundary specified by the proposed generated system matches with that of labeled data on both the sides.
2. *Correct LEFT boundary* It specifies the condition when boundaries estimated by the proposed system and the labeled data are similar on the left side of predicted NE.
3. *Correct RIGHT boundary* It specifies the condition when boundaries estimated by the proposed system and the labeled data are similar on the right side of predicted NE.

6.6 Comparison with existing wrapper-based models

The proposed study performs the comparison with other wrapper-based models. In particular, there are two different variants of wrapper models *viz* Randomized- and Deterministic-type wrapper model. Randomized-type wrapper models perform classifier dependent feature selection and are less prone to stuck at local optima, for example, genetic algorithm (GA) (Holland 1992), randomized hill climbing (Skalak 1994), simulated annealing (Lin et al. 2008). Deterministic-type wrapper model is the simple variant of wrapper-based feature selection which is computationally less complex than randomized type; however, it is more prominent to get stuck at local optima. Popular randomized wrapper-based model includes sequential forward selection (Kittler 1978), sequential backward elimination (Kittler 1978), recursive feature elimination (RFE) (Guyon et al. 2002).

In our study, we have explored popular genetic algorithm (randomized) and recursive feature elimination (deterministic) to perform comparison with our proposed PSO-based approach as shown in Table 17.

Table 9 Feature set selected for each of the datasets by the proposed PSO-based feature selection technique

Feature Datasets	Context features	Content features	Orthographic features	Prefix	Suffix	Word normalization	Informative word	Dynamic NE tag	Word length	Infrequent word	Head noun	Verb trigger	PoS information	Chunk information
	(1–6)	(7–16)	(17–34)	(35–38)	(39–42)	(45–46)	(47–50)	(51)	(52)	(53)	(54)	(55)	(56)	(57)
GENIA	1, 6	7, 8, 11, 13, 15, 16	18, 22, 24, 25, 26, 28, 31, 32, 33, 34	37	38, 42	45, 46	48	–	–	–	✓	✓	✓	✓
GENETAG	1, 3, 6	7, 8, 9, 12, 13, 16	18, 19, 22, 23, 25, 27, 28, 31, 34	36, 38	41, 42	46	50	✓	–	–	✓	✓	✓	✓
AIMed	1, 2, 4, 5, 6	7, 12, 13, 15, 16	17, 18, 20, 23, 25, 29, 33	–	38, 41	46	–	–	✓	–	✓	✓	✓	✓
BC-II	1, 2, 5	7, 8, 11, 14, 16	18, 21, 22, 24, 25, 26, 28, 32, 33	–	38	46	–	–	–	–	✓	–	✓	✓

Feature lengths are specified within brackets, and the corresponding selected feature index numbers are listed for each of the datasets. '✓' denotes the feature has selected, and '–' denotes the feature has not been selected

Table 10 Detailed assessment of the proposed technique on GENIA dataset

Class	Recall	Precision	F-measure
Overall			
FULLY Correct	77.25	73.37	75.26
Correct LEFT boundary	80.98	76.92	78.90
Correct RIGHT boundary	84.41	80.17	82.23
Protein			
FULLY Correct	82.78	72.92	77.54
Correct LEFT boundary	87.53	77.11	81.99
Correct RIGHT boundary	89.53	78.87	83.87
Cell_line			
FULLY Correct	58.11	55.65	56.85
Correct LEFT boundary	63.42	60.73	62.05
Correct RIGHT boundary	70.80	67.80	69.26
DNA			
FULLY Correct	73.86	73.62	73.74
Correct LEFT boundary	76.30	76.05	76.18
Correct RIGHT boundary	80.68	80.42	80.55
Cell_type			
FULLY Correct	70.54	81.21	75.50
Correct LEFT boundary	71.87	82.74	76.92
Correct RIGHT boundary	77.20	88.87	82.62
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	79.71

Table 11 Detailed assessment of the proposed technique on GENETAG dataset

Class	Recall	Precision	F-measure
Overall			
FULLY Correct	96.00	90.28	93.05
Correct LEFT boundary	97.46	91.65	94.47
Correct RIGHT boundary	96.59	90.83	93.62
NEWGENE			
FULLY Correct	97.19	90.31	93.62
Correct LEFT boundary	98.67	91.68	95.05
Correct RIGHT boundary	97.77	90.84	94.18
NEWGENE1			
FULLY Correct	01.35	33.33	02.60
Correct LEFT boundary	01.35	33.33	02.60
Correct RIGHT boundary	02.70	66.67	05.19

The obtained results show that PSO outperforms RFE for all the four datasets and GA for three datasets (GENIA, GENETAG, BC-II) in terms of F-score and the number of feature selected. On AIMed dataset, GA-based feature selection technique performs comparable to PSO in terms of F-score.

Table 12 Detailed assessment of the proposed technique on AIMed dataset

Class	Recall	Precision	F-measure
Overall			
FULLY Correct	90.73	89.46	90.09
Correct LEFT boundary	95.10	93.76	94.42
Correct RIGHT boundary	91.89	90.60	91.24

We also observe that PSO was more efficient than other two feature selection techniques in identifying the small subset of features (Tables 15, 16).

6.7 Comparisons with the existing systems

This section presents the comparison of our system with the other state-of-art systems that have used the same datasets to perform experiments. In our proposed approach, we did not use heavy domain-specific resources and/or tools except PoS and chunk information. Our system is able to outperform the best system on the GENIA, BC-II datasets. On GENIA dataset, we achieve the F-measure value of 75.26%. The F-measure value on GENETAG dataset is 93.05%, on AIMed the obtained F-measure value was 90.09% and on BC-II dataset, we achieve the F-measure value of 89.11%. Tables 13, 14, 15 and 16 show the comparison with the existing technique on GENIA, AIMed, GENETAG and BC-II dataset, respectively. We observed that the results obtained on AIMed dataset were not able to beat (Ekbal et al. 2013). In context to that we would like to mention that the (Ekbal et al. 2013) used genetic algorithm using the population size: 200, while in PSO, we set the population size:20 which is significantly very less. However, when we set GA with the same population size as PSO, we obtained comparable F-score value as shown in Table 17. It should be noted that in case of AIMed dataset, majority of the work is carried to extract the protein interaction pair assuming the entity (protein) is already identified. As our basic aim is to develop a generic system that can perform well across several biomedical domains, we were motivated to use AIMed dataset to validate our approach.

7 Error analysis

Our close investigation to the obtained results during the experimentation shows that the model developed for biomedical entity extraction suffers due to the implicit representation of target tokens. Proper boundary identification of NE often creates a problem.

We perform error analysis in terms of confusion matrices as shown in Tables 18, 19, 20 and 21 for AIMed, GENETAG, GENIA and BC-II datasets, respectively.

For the AIMed dataset, a large number of 'B-protein' (Begin of protein term) and 'I-protein' (Intermediate tokens of protein term) were wrongly classified as 'O' (Others). A sum of 171 instances were wrongly classified for these two classes. It was observed that majority of the mis-classifications was due to incorrect prediction of non-NE terms by NEs. This case was mostly between 'O' with 'B-NEWGENE' or 'I-NEWGENE'.

In the GENETAG dataset, it was investigated that only 'B-NEWGENE' was incorrectly predicted as 'O' while it was also observed that most mis-classifications were due to the fact that all the other classes were wrongly predicted as 'I-NEWGENE'. The possible reason behind this anomaly was due to the occurrence of 'I-NEWGENE' in most of the times in the training data. Our system was unable to predict the 'O' tags. A total of 743 instances were wrongly predicted as 'I-NEWGENE' and 'B-NEWGENE'.

For the GENIA dataset, the 'I-cell_line' and 'I-cell_type' were not predicted correctly. About 284 instances of 'I-cell_line' were incorrectly predicted as 'I-cell_type' and 73 instances of 'I-cell_type' were predicted as 'I-cell_line'. We observe quite similar behavior for the classes, 'B-cell_type' and 'B-cell_line'. A total of 3,275 instances were mis-classified for these two classes. A significant number of NE instances are also predicted as non-NE, which might have caused low recall.

In the case of BC-II dataset, the major cause of the error was due to wrong entity classification. From the confusion matrix, it was observed that 427 instances that should be classified as 'B-GENE' were wrongly classified as the 'I-GENE' and 416 instances from 'I-GENE' were classified as the 'B-GENE'. We also observed error where there was a case of missing entity. A total of 424 instances were missed and were predicted as 'O'. A case of over prediction was also observed where 'O' was predicted by entity. A total of 402 instances were reported such cases. We further analyzed the output for each dataset. We observed that our system lacks in correctly classifying the instances which includes parentheses. Example of such an instance is reported in Table 22. In Table 22, system showed acceptable performance in identifying 'B-protein', but was unable to predict the boundary of the instances as it was unable to tag the parentheses correctly. It is observed that short word (length 3 or less) started with capital letter is predicted as NE. This might be due to the capitalization feature that we defined. Example of such an instance is reported in Table 23. In training data, there are enough such instances. It is also observed that the system makes many errors in identifying the boundary of long NE. This may be because of the appearance of many symbols and/or common words inside the NE. Contextual informa-

Table 13 Comparisons between existing approaches: GENIA dataset

System	Classification methodology	Domain-specific information	F- Measure
Our developed system	PSO-based feature selection (CRF and PSO)	PoS, phrase	75.26
Ekbal and Saha (2013)	Feature selection (CRF and GA)	PoS, phrase	74.90
GuoDong and Jian (2004) final	HMM and SVM	Name duplication, cascaded NEs dictionary, PoS, phrase	72.55
GuoDong and Jian (2004)	HMM and SVM	PoS, phrase	64.1
Kim et al. (2005)	Two-phase model with ME and CRF	PoS, phrase, rule-based component	71.19
Park et al. (2006)	ME	PoS , phrase, domain-salient words using WSJ, morphological patterns,collocations from MEDLINE	66.91
Finkel (2004)	ME	Gazetteers, web querying, surrounding abstracts, abbreviation handling, BNC corpus, POS	70.06
Settles (2004)	CRF	PoS, semantic knowledge sources of 17 lexicons	70.00
Saha et al. (2009)	ME	PoS,phrase	67.41
Song et al. (2004) final	SVM,CRF	PoS, phrase,Virtual Sample	66.28
Song et al. (2004) base	SVM	PoS, phrase	63.85
Ponomareva et al. (2007)	HMM	PoS	65.7

Table 14 Comparisons between existing approaches: AIMed dataset

System	Classification methodology	Domain-specific information	F-measure
Our developed system	PSO-based feature selection (CRF and PSO)	PoS, phrase	90.09
Ekbal et al. (2013)	Feature selection (SVM and GA)	PoS, phrase	93.60

Table 15 Comparison of PSO with genetic algorithm (GA) and recursive feature elimination (RFE)-based feature selection

	Objective dataset	GENIA	GENETAG	AIMed	BC-II
GA	No. of features	34	37	28	29
	F-score	73.84	91.94	90.35	88.76
RFE	No. of features	23	37	25	25
	F-score	72.70	89.77	88.04	85.99
CRF[PSO]	No. of features	28	29	25	22
	F-score	75.26	93.05	90.09	89.11

tion, in many cases, does not provide enough information to predict some of the penultimate or last word of the NE. Such examples are shown in Tables 24 and 25 for the GENETAG dataset. We observe similar behavior for the AIMed datasets. Boundary detection problem was also reported in the case of BC-II dataset. Examples have been shown in Table 29. In Table 30, the entity was not correctly classified and is predicted as 'O'. Example of such an instance is reported in Table 23. Our system was unable to properly predict the same instance if it appears more than once at two different times in a sentence. In Table 26, e.g., both instances of 'NF-kappaB' were classified as B-protein in the test data, but the second instance of NF-kappaB was actually not a protein. It was also seen that for most of the NEs our system was able to identify the left boundary but due to some punctuation symbols (",","(",")","."") sometimes our system fails to identify the whole boundary of the NE. Example of such an instances is reported in Table 27. This leads to some drop in the overall accuracy of our system. For 'cell_type' NE, our system could not perform well as compared to other NEs because our sys-

Table 16 Comparisons between the existing approaches: GENETAG

System	Classification methodology	Domain-specific information	F-score
Our system	PSO-based feature selection (CRF and PSO)	PoS, phrase	93.05
Ekbal et al. (2013)	GA-based ensemble (CRF and SVM)	PoS, phrase	93.95
Song et al. (2004)	SVM	–	66.7
Bickel et al. (2004)	SVM	A dictionary	72.1
Kinoshita et al. (2005)	TnT (Brants 2000), the Trigrams Tags	Dictionary-based postprocessing HMM-based part-of-speech tagger	80.9
Mitsumori et al. (2005)	SVM	Gene/protein name dictionary	78.09
Finkel et al. (2005)	ME+ postprocessing		82.2
McDonald and Pereira (2005)	CRF		82.4
Wang et al. (2008)	HMM, SVM, Ensemble technique	Postprocessing	82.58

Table 17 Comparisons between the existing approaches: BC-II

System	Classification methodology	Domain-specific information	F-measure
Our System	CRF+ PSO-based feature selection	POS, phrase	89.11
Ando (2007)	Semi-supervised learning Alternating Structure Optimization (ASO)	Word strings and character types of the current and neighboring words, domain lexicon lookup	87.21
Kuo et al. (2007)	CRF	POS, phrase abbreviations of biological chemical compounds	86.83
Huang et al. (2007)	SVM + CRF	POS, phrase, 123,503 predicates to characterize each word	86.57
Klinger et al. (2007)	CRF	PoS, phrase output of a normalizing tagger, ProMiner	86.33
Ganchev et al. (2007)	CRF + Greedy-based feature selection	Word features based on distributional clustering	86.28
Liu et al. (2007)	CRF	POS, Token shape, Suffix, Dictionary-lookup (BioThesaurus and UMLS Metathesurus)	85.89
Grover et al. (2007)	CRF + bidirectional maximum entropy Markov Model (BMEMM)	Contextual feature, POS, orthographic features, head noun, features derived from the abbreviation matcher and from in-house protein gazetteer derived from RefSeq	84.70
Struble et al. (2007)	CRF	POS, phrase	82.85
Vlachos (2007)	CRF	POS, phrase	82.84
Baumgartner Jr et al. (2007)	Combining the output of multiple gene mention identification systems	Phrase	80.95

Table 18 Confusion matrix of proposed model on AIMed dataset

Reference	Predicted		
	B-protien	I-protien	O
B-protein	1074	3	45
I-protein	26	178	126
O	50	63	1746

Table 21 Confusion matrix of the proposed model on BC-II dataset

Reference	Predicted		
	B-GENE	I-GENE	O
B-GENE	5492	427	117
I-GENE	416	4631	307
O	219	183	123021

tem was unable to fully detect the proper boundary which leads to the mis-classification. Example of such an instance is reported in Table 28.

7.1 Computational complexity of the system

The cost of computations required for a complete run of the proposed feature selection approach is the product of the computations required for PSO and the training time of CRF. The total cost to run PSO is the sum of the computation costs required to calculate the cost of a candidate solution and the computations required to update each particle's position and velocity. Let us assume that we have N = no. of particles and I = no. of iterations, F_{avg} is the average number of bits selected from a given particle. L is the size of label set, T is total number of features and S is the size of average training samples. Mathematically, it can be described as:

$$\begin{aligned} &\text{Cost of a candidate solution} = \mathcal{O}(F_{\text{avg}}) \\ &\text{Cost to update each particle's position} = \mathcal{O}(N) \\ &\text{Cost to update each particle's velocity} = \mathcal{O}(N) \end{aligned} \tag{13}$$

$$\begin{aligned} &\text{Total cost of PSO for a single iteration} \\ &\quad = \mathcal{O}(F_{\text{avg}} * N) \\ &\text{Total cost to train CRF on active feature set} \\ &\quad = \mathcal{O}(T * L^2 * S^2) \\ &\text{Total cost to model for a single iteration} \\ &\quad = \mathcal{O}(F_{\text{avg}} * N)(T * L^2 * S^2) \\ &\text{Total cost of model for } I \text{ number of iterations} \end{aligned} \tag{14}$$

Table 19 Confusion matrix of proposed model on GENETAG dataset

Reference	Predicted				
	B-NEWGENE	B-NEWGENE1	I-NEWGENE	I-NEWGENE1	O
B-NEWGENE	5764	0	24	0	44
B-NEWGENE1	2	1	70	0	0
I-NEWGENE	4	0	6393	0	2
I-NEWGENE1	0	0	67	1	0
O	510	0	233	0	123442

Table 20 Confusion matrix of the proposed model on GENIA dataset

Reference	Predicted										
	B-cell_line	B-cell_type	B-DNA	B-protein	B-RNA	I-cell_line	I-cell_type	I-DNA	I-protein	I-RNA	O
B-cell_line	212	33	0	14	0	29	2	0	0	0	49
B-cell_type	93	795	4	56	0	12	36	2	0	0	129
B-DNA	1	0	477	63	0	0	0	15	1	0	59
B-protein	10	9	36	2622	1	4	0	6	70	0	232
B-RNA	0	0	0	9	53	0	0	0	0	2	7
I-cell_line	21	2	0	0	0	482	73	0	2	0	78
I-cell_type	6	35	0	8	0	284	1290	5	42	0	178
I-DNA	0	0	23	10	0	2	0	872	37	0	103
I-protein	0	2	0	144	0	9	10	56	2230	1	352
I-RNA	0	0	0	3	4	0	0	0	6	99	14
O	37	82	76	459	10	74	70	153	264	6	50227

Table 22 An example of mis-classification on AIMed dataset

Instances	ET(	A	)	and	ET	(	B	)	receptors	in	human	myocardial	trabeculae	.
Actual	B-protien	I-protien	I-protien	I-protien	O	B-protien	I-protien	I-protien	I-protien	O	O	O	O	O
Predicted	B-protien	O	O	O	O	B-protien	O	O	O	O	O	O	O	O

Table 23 An example of mis-classification on AIMed dataset

Instance	We	have	therefore	tested	whether	other	CC	chemokines	could	bind	to	and	activate	CCR5	.
Actual	O	O	O	O	O	O	O	O	O	O	O	O	O	B-protien	O
Predicted	O	O	O	O	O	O	B-protien	O	O	O	O	O	O	B-protien	O

Table 24 An example of mis-classification on GENETAG dataset

Instances	includes	the	mammalian	RGS	proteins	RGS6	,	RGS7	.
Actual	O	O	B-NEWGENE	I-NEWGENE	I-NEWGENE	B-NEWGENE1	O	B-NEWGENE	O
Predicted	O	O	B-NEWGENE	I-NEWGENE	I-NEWGENE	I-NEWGENE	O	B-NEWGENE	O

Table 25 An example of mis-classification on GENETAG data

Instances	RGS6	,	RGS7	,	RGS9	,	and	RGS11	.
Actual	B-NEWGENE1	O	B-NEWGENE	O	B-NEWGENE	O	O	B-NEWGENE	O
Predicted	I-NEWGENE	O	B-NEWGENE	O	B-NEWGENE	O	O	B-NEWGENE	O

Table 26 An example of mis-classification on GENIA dataset

Instances	Cl-1	treatment	had	no	detected	effect	on	NF-kappaB	activation	in	lung	tissue	.
Actual	O	O	O	O	O	O	O	O	O	O	O	O	O
Predicted	O	O	O	O	O	O	O	B-protein	O	O	O	O	O

Table 27 An example of mis-classification on GENIA data

Instances	accessibility	of	V	,	D	,	and	J	gene	segments
Actual	O	O	O	O	O	O	O	B-DNA	I-DNA	I-DNA
Predicted	O	O	B-DNA	I-DNA	I-DNA	I-DNA	I-DNA	I-DNA	I-DNA	I-DNA

Table 28 An example of mis-classification on GENIA data

Instances	In	distinction	to	drug-free	maturing	dendritic	cells	,	2.5	micromol/L
Actual	O	O	O	B-cell_type	I-cell_type	I-cell_type	I-cell_type	O	O	O
Predicted O	O	O	O	O	B-cell_type	I-cell_type	O	O	O	O

Table 29 An example of mis-classification on BC-II data

Instances	The	7.2	kb	EcoRI	fragment	of	AfMNPV	was	cloned
Actual	O	B-GENE	I-GENE	I-GENE	I-GENE	O	O	O	O
Predicted	O	B-GENE	I-GENE	O	O	O	O	O	O

Table 30 An example of mis-classification on BC-II data

Instances	test	for	Borrelia	burgdorferi	serum	antibodies	had	positive	results
Actual	O	O	B-GENE	I-GENE	I-GENE	I-GENE	O	O	O
Predicted	O	O	O	O	B-GENE	I-GENE	O	O	O

$$= \mathcal{O}(I(F_{\text{avg}} * N)(T * L^2 * S^2))$$

We have also calculated the job execution times of PSO model for performing feature selection. In addition to that, we have also made comparison with GA-based feature selection (as both are randomized wrapper models) and found that computational complexity of PSO is less compared to GA. We observed that, for the swarm size of 10 particles, PSO took 5316 s in order to complete a single iteration, while GA-based approach reported execution time of 7200 s in order to complete a single generation with population size of 10 chromosomes as shown in Fig. 4. This proves that PSO is much faster than GA-based approach for solving the problem of feature selection (Tables 29, 30). In terms of computational complexity, PSO is less complicated and still achieves better performance compared to GA. Thus with very limited number of executions we can achieve good accuracy with the use of PSO. This proves the utility of PSO-based approach in feature selection. It not only achieves better performance compared to GA-based approach but also converges faster.

8 Conclusions and future works

In this paper, we propose a PSO-based feature selection technique for entity extraction in multiple biomedical domain corpora. The proposed method makes use of a diverse feature set, which was implemented without using much domain-specific resources and/or tools. We have evaluated our approach on benchmark datasets, namely GENIA, GENETAG, AIMed and BioCreative II (BC-II) gene mention recognition datasets. We observe that classifier performs better with a reduced feature set in comparison with the model developed with all features. As a classifier we have used CRF. Evaluation results indicate that the use of binary PSO for feature selection tends to improve the accuracy and reliability of a classification model on the biomedical dataset. We have carried out a thorough sensitivity analysis of different parameters of PSO and have shown their impacts on the overall system performance. Evaluation results for the GENIA dataset show the F-measure value of 75.26% using the pruned feature set compared to 68.72% F-measure obtained when the classifier is trained with the complete set of features. Because of feature selection, we also observed significant performance increment for the other three benchmark datasets. For BC-II dataset, the system was able to improve the F-measure value by 6.96, 2.11% on AIMed dataset and 5.41% on GENETAG dataset. Detailed comparative studies with the other existing techniques also prove the efficacy of our proposed technique.

In the present work, we develop feature selection technique based on single objective optimization(SOO) where we determined the most relevant set of features with respect to the F-measure value. In future, we would like to develop the feature selection technique based on multi-objective optimization(MOO) that would be able to optimize more than one classification quality measures simultaneously. We would also like to measure the effectiveness of the proposed approach for the other kinds of datasets.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Aghdam MH, Heidari S (2015) Feature selection using particle swarm optimization in text categorization. J Artif Intell Soft Comput Res 5(4):231–238

Alatas B, Akin E (2008) Rough particle swarm optimization and its applications in data mining. Soft Comput 12(12):1205–1218

Ando RK (2007) Biocreative II gene mention tagging system at IBM Watson. In: Proceedings of the second biocreative challenge evaluation workshop, vol 23, pp 101–103

Baumgartner Jr WA, Lu Z, Johnson HL, Caporaso JG, Paquette J, Lindemann A (2007) An integrated approach to concept recognition in biomedical text. In: Proceedings of the second biocreative challenge evaluation workshop, vol 23, pp 257–271

Berger AL, Pietra VJD, Pietra SAD (1996) A maximum entropy approach to natural language processing. Comput Linguist 22(1):39–71

Bickel S, Brefeld U, Faulstich L, Hakenberg J, Leser U, Plake C (2004) A support vector machine classifier for gene name recognition. In: Embo workshop: a critical assessment of text mining methods in molecular biology, Granada, Spain

Cagnina LC, Errecalde ML, Ingaramo DA, Rosso P (2008) A discrete particle swarm optimizer for clustering short-text corpora. In: Proceedings of bioinspired optimization methods and their applications, BIOMA-2008, Ljubljana, Slovenia

Chen W-N, Zhang J, Lin Y, Chen N, Zhan Z-H, Chung HS-H (2013) Particle swarm optimization with an aging leader and challengers. IEEE Trans Evol Comput 17(2):241–258

Chinnaswamy A, Srinivasan R (2016) Hybrid feature selection using correlation coefficient and particle swarm optimization on microarray gene expression data. In: Innovations in bio-inspired computing and applications. Springer, Berlin, pp 229–239

Chuang L-Y, Chang H-W, Tu C-J, Yang C-H (2008) Improved binary PSO for feature selection using gene expression data. Comput Biol Chem 32(1):29–38

Correa ES, Freitas AA, Johnson CG (2006) A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, pp 35–42

Das S (2001) Filters, wrappers and a boosting-based hybrid for feature selection. In: ICML, vol 1, pp 74–81

Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 3(02):185–205

Eberhart RC, Shi Y (1998) Comparison between genetic algorithms and particle swarm optimization. In: International conference on evolutionary programming, pp 611–616

Ekbal A, Saha S (2013) Stacked ensemble coupled with feature selection for biomedical entity extraction. Knowl Based Syst 46:22–32
Ekbal A, Saha S, Sikdar UK (2013) Biomedical named entity extraction: some issues of corpus compatibilities. SpringerPlus 2(1):1
Finkel J, Dingare S, Manning CD, Nissim M, Alex B, Grover C (2005) Exploring the boundaries: gene and protein identification in biomedical text. BMC Bioinform 6(Suppl 1):S5
Finkel J, Dingare S, Nguyen H, Nissim M, Manning C, Sinclair G (2004) Exploiting context for biomedical entity recognition: from syntax to the web. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, pp 88–91
Ganchev K, Crammer K, Pereira F, Mann G, Bellare K, McCallum A, Carroll S, Jin Y, White P (2007) Penn/umass/chop biocreative II systems. In: Proceedings of the second biocreative challenge evaluation workshop, vol 23. pp 119–124
Ghamisi P, Benediktsson JA (2015) Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci Remote Sens Lett 12(2):309–313
Grover C, Haddow B, Klein E, Matthews M, Nielsen LA, Tobin R (2007) Adapting a relation extraction pipeline for the biocreative II task. In: Proceedings of the biocreative II workshop, vol 2
GuoDong Z, Jian S (2004) Exploring deep knowledge resources in biomedical name recognition. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, pp 96–99
Gupta DK, Reddy KS, Ekbal A (2015) Pso-asent: feature selection using particle swarm optimization for aspect based sentiment analysis. In: International conference on applications of natural language to information systems, pp 220–233
Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422
Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge, MA, USA
Hsieh S-T, Sun T-Y, Liu C-C, Tsai S-J (2009) Efficient population utilization strategy for particle swarm optimizer. IEEE Trans Syst Man Cybern Part B (Cybern) 39(2):444–456
Huang H-S, Lin Y-S, Lin K-T, Kuo C-J, Chang Y-M, Yang B-H (2007) High-recall gene mention recognition by unification of multiple backward parsing models. In: Proceedings of the second biocreative challenge evaluation workshop, vol 23, pp 109–111
Juang C-F (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern Part B (Cybern) 34(2):997–1006
Kao Y-T, Zahara E (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 8(2):849–857
Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: 1997 IEEE international conference on systems, man, and cybernetics, 1997. Computational cybernetics and simulation, vol 5, pp 4104–4108
Kim J-D, Ohta T, Tsuruoka Y, Tateisi Y, Collier N (2004) Introduction to the bio-entity recognition task at JNLPBA. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, pp 70–75
Kim S, Yoon J, Park K-M, Rim H-C (2005) Two-phase biomedical named entity recognition using a hybrid method. In: Natural language processing—IJCNLP 2005. Springer, Berlin, pp 646–657
Kinoshita S, Cohen KB, Ogren PV, Hunter L (2005) Biocreative Task1A: entity identification with a stochastic tagger. BMC Bioinform 6(Suppl 1):S4
Kittler J (1978) Feature set search algorithms. In: Pattern recognition and signal processing, pp 41–60
Klinger R, Friedrich CM, Fluck J, Hofmann-Apitius M (2007) Named entity recognition with combinations of conditional random fields. In: Proceedings of the second biocreative challenge evaluation workshop
Krisshna NA, Deepak VK, Manikantan K, Ramachandran S (2014) Face recognition using transform domain feature extraction and pso-based feature selection. Appl Soft Comput 22:141–161
Kumar A, Patidar V, Khazanchi D, Saini P (2016) Optimizing feature selection using particle swarm optimization and utilizing ventral sides of leaves for plant leaf classification. Procedia Comput Sci 89:324–332
Kuo C-J, Chang Y-M, Huang H-S, Lin K-T, Yang B-H, Lin Y-S (2007) Rich feature set, unification of bidirectional parsing and dictionary filtering for high f-score gene mention tagging. In: Proceedings of the second biocreative challenge evaluation workshop, vol 23, pp 105–107
Lafferty JD, McCallum A, Pereia FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, pp 282–289. http://dl.acm.org/citation.cfm?id=645530.655813
Lin S-W, Lee Z-J, Chen S-C, Tseng T-Y (2008) Parameter determination of support vector machine and feature selection using simulated annealing approach. Appl Soft Comput 8(4):1505–1512
Lin S-W, Ying K-C, Chen S-C, Lee Z-J (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst Appl 35(4):1817–1824
Liu H, Torii M, Hu Z, Wu C (2007) Gene mention and gene normalization based on machine learning and online resources. In: Proceedings of the second biocreative challenge workshop, pp 135–140
Liu Y, Wang G, Chen H, Dong H, Zhu X, Wang S (2011) An improved particle swarm optimization for feature selection. J Bionic Eng 8(2):191–200
Liu Z, Liu S, Liu L, Sun J, Peng X, Wang T (2016) Sentiment recognition of online course reviews using multi-swarm optimization-based selected features. Neurocomputing 185:11–20
Lu Y, Liang M, Ye Z, Cao L (2015) Improved particle swarm optimization algorithm and its application in text feature selection. Appl Soft Comput 35:629–636
McDonald R, Pereira F (2005) Identifying gene and protein mentions in text using conditional random fields. BMC Bioinform 6(Suppl 1):S6
Merwe D, Van der Engelbrecht AP (2003) Data clustering using particle swarm optimization. In: The 2003 Congress on evolutionary computation, 2003. CEC'03, vol 1, pp 215–220
Mitsumori T, Fation S, Murata M, Doi K, Doi H (2005) Gene/protein name recognition based on support vector machine using dictionary as features. BMC Bioinform 6(Suppl 1):S8
Park K-M, Kim S-H, Rim H-C, Hwang Y-S (2006) ME-based biomedical named entity recognition using lexical knowledge. ACM Trans Asian Lang Inf Process (TALIP) 5(1):4–21
Pedersen MEH (2010) Good parameters for particle swarm optimization. Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001
Peram T, Veeramachaneni K, Mohan CK (2003) Fitness-distance-ratio based particle swarm optimization. In: Proceedings of the 2003 IEEE Swarm intelligence symposium, 2003. SIS'03
Ponomareva N, Pla F, Molina A, Rosso P (2007) Biomedical named entity recognition: a poor knowledge hmm-based approach. In: Natural language processing and information systems. Springer, Berlin, pp 382–387
Rabiner L, Juang B-H (1993) Fundamentals of speech recognition. Prentice-Hall, Inc., NJ, USA

Ramadan RM, Abdel-Kader RF (2009) Face recognition using particle swarm optimization-based selected features. Int J Signal Process Image Process Pattern Recognit 2(2):51–65

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

Samadzadegan F, Saeedi S (2009) Clustering of lidar data using particle swarm optimization algorithm in urban area. Laserscanning 09(38):334–339

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

Shang L, Zhou Z, Liu X (2016) Particle swarm optimization-based feature selection in sentiment classification. Soft Comput 20(10):1–14. doi:10.1007/s00500-016-2093-2

Shanno DF (1970) Conditioning of quasi-Newton methods for function minimization. Math Comput 24(111):647–656

Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). Biometrika 52(3/4):591–611

Sheikhpour R, Sarram MA, Sheikhpour R (2016) Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer. Appl Soft Comput 40:113–131

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: The 1998 IEEE international conference on evolutionary computation proceedings, 1998. IEEE World Congress on computational intelligence, pp 69–73

Shi Y, Eberhart RC (2001) Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 Congress on evolutionary computation, vol 1, pp 101–106

Skalak DB (1994) Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Proceedings of the eleventh international conference on machine learning, pp 293–301

Song Y, Kim E, Lee GG, Yi B-k (2004) POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, pp 100–103

Struble CA, Povinelli RJ, Johnson MT, Berchanskiy D, Tao J, Trawicki M (2007) Combined conditional random fields and n-gram language models for gene mention recognition. In: Proceedings of the second biocreative challenge evaluation workshop; 23–25 April 2007; Madrid, Spain, pp 81–83

Tran B, Xue B, Zhang M (2014) Overview of particle swarm optimisation for feature selection in classification. In: Asia-Pacific conference on simulated evolution and learning, pp 605–617

Vlachos A (2007) Tackling the biocreative2 gene mention task with conditional random fields and syntactic parsing. In: Proceedings of the second biocreative challenge evaluation workshop; 23–25 April 2007; Madrid, Spain, pp 85–87

Wang H, Zhao T, Tan H, Zhang S (2008) Biomedical named entity recognition based on classifiers ensemble. IJCSA 5(2):1–11

Xi M-L, Sun J, Wu Y (2010) Quantum-behaved particle swarm optimization with binary encoding. Control Decis 1:019

Yan X, Wu Q, Liu H, Huang W (2013) An improved particle swarm optimization algorithm and its application. Int J Comput Sci Issues (IJCSI) 10(1):316–324

Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5:1205–1224

Zhang J-R, Zhang J, Lok T-M, Lyu MR (2007) A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training. Appl Math Comput 185(2):1026–1037