Knowledge-Based Systems 125 (2017) 116–135

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis

Md Shad Akhtar, Deepak Gupta, Asif Ekbal*, Pushpak Bhattacharyya

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

ARTICLE INFO

Article history:
Received 21 September 2016
Revised 15 February 2017
Accepted 25 March 2017
Available online 27 March 2017

Keywords:
Sentiment analysis
Aspect term extraction
Feature selection
Ensemble
Conditional random field
Support vector machine
Maximum entropy
Particle swarm optimization

ABSTRACT

In this paper we present a cascaded framework of feature selection and classifier ensemble using particle swarm optimization (PSO) for aspect based sentiment analysis. Aspect based sentiment analysis is performed in two steps, *viz.* aspect term extraction and sentiment classification. The pruned, compact set of features performs better compared to the baseline model that makes use of the complete set of features for aspect term extraction and sentiment classification. We further construct an ensemble based on PSO, and put it in cascade after the feature selection module. We use the features that are identified based on the properties of different classifiers and domains. As base learning algorithms we use three classifiers, namely Maximum Entropy (ME), Conditional Random Field (CRF) and Support Vector Machine (SVM). Experiments for aspect term extraction and sentiment analysis on two different kinds of domains show the effectiveness of our proposed approach.

1. Introduction

Recent past has witnessed a phenomenal growth of internet users globally, and the third world countries like India, China etc. are not the exceptions. Use of social media and messaging applications grew 203% year-on-year in 2013, with overall application users rising 115% over the same period, as reported by Statista, citing data from Flurry Analytics. This growth means that 1.61 billion people are now active in social media around the world and this is expected to advance to 2 billion users in 2016, led by India. The research shows that consumers are now spending daily approximately 8 h on digital media including social media and mobile internet usages. This has completely changed the lifestyles of people. Before buying any product or acquiring any service, customers or users nowadays depend heavily on the web-based information available through several shopping portals, online sites, blogs, tweets etc. They want to make sure that the products that they buy or the services that they acquire are of high quality. Be it buying a product from an e-commerce website or going for a dinner/lunch to a restaurant or planning for watching a movie at the theater, they always seek other users' opinions about the product and/or services before experiencing themselves. The unprecedented volume and variety of user-generated contents make it difficult to go through all the information manually. But, in order to get an unbiased opinion of a product or service one has to extract and read all the reviews. Unfortunately this is not an easy task to perform considering the huge amount of time and efforts involved. Therefore, there is a need to develop tools and techniques that will aid users in mining the desired information from the collection of reviews.

Sentiment Analysis [1] is a well-established task that targets to determine the polarity of opinion expressed in a given user's review. In general, polarity can either be positive, negative or neutral depending on the sentiment expressed in a review [2,3]. These information not only help an individual seeking for information, but also facilitate the process of refining various business decisions to improve the quality of products or services. There have been quite a significant number of existing methods for sentiment analysis in various domains [4–8]. It is to be noted that most of these existing works focus on discovering sentiments at the document [4] or sentence [5] level. However, the techniques focused on such a coarse-level analysis (i.e. *document level* or *sentence level*) do not always satisfy the users' needs as per their expectations. They often require more finer information from a review in terms of specific *aspects* (or, *features* or *attributes*) of a product or service. For example, an user may be interested in only specific aspects such as the 'design', 'battery life' or 'display screen' of a laptop. Different opinions may have been expressed by different users on each of these

* Corresponding author.
E-mail addresses: asif.ekbal@gmail.com, asif@iitp.ac.in (A. Ekbal).

http://dx.doi.org/10.1016/j.knosys.2017.03.020
0950-7051/

Table 1
Example of user's review, aspect terms and their sentiment polarities.

#	Reviews	Aspect term	Polarity
1.	The fried rice is amazing here.	fried rice	positive
2.	But the staff was so horrible to us.	staff	negative
3.	The menu is limited but almost all of the dishes are excellent.	menu, dishes	negative, positive
4.	The food was delicious but do not come here on a empty stomach.	food	conflict
5.	I grew up eating Dosa and have yet to find a place in NY to satisfy my taste buds.	dosa	neutral

aspects. Therefore, it is important to detect the sentiments with respect to different aspects, and this process is known as 'Aspect based Sentiment Analysis'. The term "aspect" refers to an attribute or a component of the product/service that has been commented on in a review. In aspect level sentiment analysis, we are interested in determining the polarity orientation towards each aspect term, which appears in a review sentence. Polarity values of the aspect terms in a sentence could be same or different. Aspect term extraction is a sub-problem of the aspect level sentiment analysis. It only concerns with detecting the aspect terms present in a sentence. If the aspect term is known beforehand then there is no need to perform aspect term extraction. Once the aspect terms are known/identified, the second sub-problem (i.e. sentiment classification) deals with assigning a polarity value (i.e. positive, negative, neutral and conflict) to each aspect term. Aspect term extraction and opinion target extraction are the related terms. The term "Aspect term extraction" was introduced in SemEval 2014 shared task [9] and the term "opinion term extraction" was coined a year later in SemEval 2015 shared task [10]. Since we are using SemEval 2014 datasets for experiments we use the term "aspect term extraction" in the paper. In Table 1 we present some examples that describe the aspects and their corresponding polarities.

The first review contains only one aspect term i.e. *fried rice* and the sentiment expressed towards it is *positive*. Similarly, review 2 expresses *negative* sentiment for the aspect term *staff*. Third review contains two aspect terms, namely *menu* and *dishes* which carry opposite sentiments, i.e. *negative* for *menu* and *positive* for *dishes*. The reviewer has made both *positive* and *negative* comments on an aspect term *food* in the fourth review, therefore, its polarity is marked as *conflict*. In the last review, aspect term is *dosa* and the sentiment is neither *positive* nor *negative*, and hence it is *neutral*.

The primary focus of aspect based sentiment analysis can be thought of as the processes of extracting aspects and then determining the sentiments that are expressed in the review [2,3]. Aspect term extraction can be modeled as a sequence labeling problem since it depends heavily on the structural and contextual information. For example, in the first review sentence of Table 1, token "*fried*" can not be termed as part of an aspect term if the context "*rice*" is not provided. However, in the presence of contextual information the multi-word token "*fried rice*" together forms an aspect term. The sentiment can be classified either at the coarse-grained level denoting such *positive* and *negative* or at the more fine-grained level that corresponds to *positive, negative, neutral* or *conflict* [9]. Also, aspect terms can influence sentiment polarity within a single domain. As an example, for the restaurant domain, *cheap* is usually positive with respect to *food*, but it denotes a negative polarity when discussing *decor* or *ambiance*[11]. In contrast, sentiment analysis at aspect level can guide users to gain more insights on the sentiments of various aspects of the target entity. Hence, the decision taken thereafter becomes more informative and practical.

According to [12], opinions are personal interpretation of information whereas sentiment refers to an expression constrained on social expectation. Therefore, in light of above definitions the term "opinion mining" would be the more suitable in the work. However, to make it consistent with the SemEval 2014 shared task on aspect based sentiment analysis we choose to use the term "sentiment analysis" throughout the paper. Also, literature shows the evidence that "sentiment analysis" and "opinion mining" are often used inter-changeably to refer the study of polarity orientation in a user written text.

Literature survey shows that the concept of aspect based sentiment analysis has recently drawn the attention of researchers worldwide. Earlier approaches to aspect extraction are based on the frequently used nouns and noun phrases [2,13,14]. Such approaches work well when many aspects are strongly associated with certain categories of words (such as nouns), but often fail when many low frequency terms are used as the aspects. Nowadays, with the emergence of few labeled datasets, supervised learning approaches [7,8] are predominantly being used. Some other approaches include the techniques, such as those that define aspect terms using a manually specified subset of Wikipedia category [15] hierarchy, unsupervised clustering [13] and semantically motivated technique [4]. In 2014 a shared task, SemEval-2014 [9] was organized for addressing the challenges of aspect based sentiment analysis and to provide a common benchmark setup. Participation to this particular task were quite overwhelming. The best performing model as reported in [16] was based on CRF that uses lexical and syntactic features. The performance of machine learner was further boosted with a set of hand-crafted heuristics. For sentiment classification the best system was reported by Wagner et al.[17]. A two-step method was proposed for the task. First, a rule-based method using sentiment lexicons (e.g. BingLiu, SentiWordNet, MPQA etc.) was applied to find the polarity of an aspect term. Subsequently, output of the rule-based system is combined with bag-of-*n*-gram features to train SVM classifier.

In [18], an application of recurrent neural network (RNN) has been discussed for the aspect term extraction task. Further, it was shown that LSTM-RNN based system with the assistance of extra set of features performs better than a feature-rich system based on CRF. In [19], a deep convolutional neural network (CNN) based architecture has been proposed for aspect term extraction task. The authors employed two different word embeddings (pre-trained embedding trained on Google News corpus and Amazon word embedding trained on 4.7-billion-word corpus from Amazon) along with Part-of-Speech (PoS) tag information and few linguistic patterns for training a CNN. Recently, tree-kernel based approach [20] has been proposed for capturing the lexical and syntactic information for identifying the polarity orientation towards the aspect terms.

Existing literature on sentiment analysis does not show much efforts for systematic feature selection. Most of the existing systems rely on heuristic based methods for selecting the most relevant set of features or classifier candidates for constructing an ensemble. The process is time-consuming because different combinations of features/classifiers should be exhaustively tried to finally fix a model. Another crucial issue is domain adaptation where the system, developed targeting a specific domain, often fails to perform reasonably when the domain is altered. The set of features or classifiers which show acceptable performance for a domain may not be equally effective for the other. Thus, careful feature selection plays an important role. An effective usage of Z-score and Information gain for feature selection in sentiment analysis has been reported in [21]. A computationally efficient feature selection technique is proposed in [22] that is based on document frequency for sentiment analysis. However, this is shown to be weaker than the information gain based feature selection in terms of reported accuracy. In [23], a PSO based method has been proposed for effective selection of optimal parameter values for SVM. Three different techniques of ensemble (fixed rules, weighted com-

bination and meta-classifiers) were used in [24] for sentiment classification. They conclude that weighted combination based ensemble method performs better over the other two. In [25], it has been shown that the problem of sentiment classification can be overcome by classifier ensemble. A bootstrap ensemble framework for twitter sentiment classification is proposed in [26]. An ensemble based approach for Chinese sentiment analysis by incorporating English sentiment resource was proposed in [27]. Wang et al. [28] conducted their experiments on ten publicly available datasets to prove the effectiveness of ensemble learning in sentiment analysis. However, it is to be noted that none of these existing works focused on determining sentiment at such a fine-grained level.

In the first part of the paper we propose a technique for feature selection that automatically determines the most relevant set of features for aspect term extraction and sentiment classification. Our algorithm is based on Particle Swarm Optimization (PSO) [29]. Feature selection, also known as variable subset selection or dimensionality reduction, is a technique that selects the most relevant features for the target problem. By removing the most irrelevant and redundant features from the data, feature selection aids to improve the performance of a classifier. We use Conditional Random Field (CRF)[30], Support Vector Machine (SVM) [31] and Maximum Entropy (ME) model [32] as the learning algorithms. We implement a diverse set of features for each of the tasks, i.e. aspect term extraction and sentiment classification. One appealing characteristics of these features being the fact that we limited ourselves in not using much domain-dependent information for the spirit of their easy adaptability to new domains and applications. Most of the features used for aspect term extraction or sentiment classification exploit lexical, syntactic or semantic level features as discussed in Section 3. Initially, we perform a series of experiments with the various combination of features. The best configuration, thus obtained, is used as the baseline model on which PSO based feature selection technique is applied. Each classifier, when subjected to PSO based feature selection, yields a set of solutions. Each solution represents a particular feature combination. The classifiers are then trained, tested and evaluated on these feature combinations. Next we select the most promising models (based on F-measure or accuracy) for each of the classifiers. In order to further improve the performance, in the second part of our algorithm we propose a PSO based ensemble learning method. The most promising models are selected and combined using a majority or weighted voting.

Experiments on the benchmark setups of SemEval-2014 show that our proposed techniques achieve state-of-the-art performance for aspect term extraction and sentiment classification. Evaluation shows that systematic method of feature selection can produce improved performance, even with a much reduced set of features. In our earlier attempt we have proposed a feature selection technique for aspect based sentiment analysis in [33]. The present research differs from our previous works with respect to the following points: current work is more extensive as we substantially extend our algorithmic view by developing more models based on the classifiers, which are heterogeneous in nature; developed feature selection technique for three different classifiers, namely CRF, SVM and ME; developed a PSO based ensemble selection method for combining the most promising models in order to further improve the performance. The contributions of the present work can be summarized as follows: (i). use of a very diverse and rich feature set for aspect term extraction and sentiment classification; (ii). efficient feature selection technique based on PSO that shows superior performance with a compact and pruned feature set for aspect term extraction and sentiment classification both; (iii). efficient ensemble construction techniques based on PSO for aspect term extraction and sentiment classification; (iv). proposal of a generalized technique that attains state-of-the-art performance for aspect based sentiment analysis in two different domains, *viz.* laptop and restaurant reviews.

The rest of the paper is structured as follows. A brief introduction to PSO is presented in Section 2. Features used for aspect term extraction and sentiment classification are presented in Section 3. In Section 4, we present our proposed method for feature selection and ensemble construction. Experimental results with detailed analysis are presented in Section 5. Finally, we conclude in Section 6.

2. Brief introduction to particle swarm optimization

Particle Swarm Optimization (PSO) [29] is a population based stochastic optimization method which is founded on the behavior of bird flocking. PSO starts with a set of random solutions and searches for the global optima by updating the generations. In PSO, the potential solutions of the given problem are called as particles and denoted as $\overrightarrow{X}(k) = (x_{(k,1)}, x_{(k,2)}, \ldots\ldots\ldots, x_{(k,n)})$ in an n-dimensional search space. Each co-ordinate $x_{(k,\,d)}$ of these particles can change with some rate, known as the velocity $v_{(k,\,d)}$ d=1,2,...,n. Every particle keeps a record of the best position that it has ever visited. Such a record is called the particle's previous best position and denoted by $\overrightarrow{B}(k)$. The global best position attained by any particle so far is also recorded and stored in a particle denoted by $\overrightarrow{G}$. An iteration comprises evaluation of each particle, then stochastic adjustment of $v_{(k,\,d)}$ in the direction of particle $\overrightarrow{X}(k)$'s previous best position and the previous best position of any particle in the neighborhood [29]. Entire process of PSO is governed by three operations, namely *evaluate, compare* and *imitate*. The *evaluation* phase measures how well each particle, i.e. the candidate solution solves the problem at hand. The *comparison* process attempts to identify the best particle by comparing different solutions. The *imitation* process produces new particles based on some of the best particles found so far. These three processes are repeated until a given stopping criterion is met. The objective is to find the particle that best solves the target problem. Velocity and neighborhood are the two important concepts in PSO. Every particle $\overrightarrow{X}(k)$ is associated with a velocity vector. The velocity vector is updated at every generation, and used to generate a new particle $\overrightarrow{X}(k)$. The neighborhood defines how other particles in the swarm, such as $\overrightarrow{B}(k)$ and $\overrightarrow{G}$, interact with $\overrightarrow{X}(k)$ to modify its respective velocity vector and position.

There are other popular approaches like the well-known genetic algorithm (GA) [34] and simulated annealing (SA)[35]. In order to solve the global optimization problems these techniques are widely used to find the good set of solutions in the search space. While SA is a probabilistic meta-heuristic approach, GA relies on the concept of survival of the fittest. Unlike GA, PSO does not retain only the good solutions. It allows the particles to move in the search space on the basis of the number of cases, and it generates good set of possible solutions without eliminating any weak particle.

3. Feature set

In this section we describe the features that we use for aspect term extraction and sentiment classification. Most of the lexical and syntactic features that we use are domain-independent and generic in nature, and therefore may be used for the applications of similar nature. We restrict ourselves in not using much domain-dependent external resources. Few of the features e.g. word cluster, WordNet, NER, head word, dependency relation, semantic orientation, lexicons feature etc. separately and/or collectively have been proved to be efficient for many different NLP problems. So, we have rigorously studied our datasets and defined a common set of features for both the restaurant and laptop domains. Some

of the features such as Word cluster, Semantic orientation, BingLiu, BingLiu Direct etc. have been re-implemented in a better and representative way to exploit generalization across multiple domains. The feature such as prefix and suffix of fixed length character sequences have not been used for aspect term extraction as such.

3.1. Features for aspect term extraction

We use the following features for aspect term extraction from the reviews of both the domains, *viz.* restaurant and laptop.

1. **Words:** Surface forms and their converted lower-cased versions are used as two separate features.
2. **Local context information:** Local contextual information plays an important role to properly identify aspect terms. For context information, for each token a window is defined that constitutes the preceding and following few tokens. Here, we use the context of size 5, i.e. 2 words to the left and 2 words to the right. For the example shown below, local context for the token w_i (*was*) contains four tokens, i.e. w_{i-2} (*the*), w_{i-1} (*staff*), w_{i+1} (*so*) and w_{i+2} (*horrible*) as its contexts.

$$\begin{array}{cccccccc} & & w_{i-2} & w_{i-1} & w_i & w_{i+1} & w_{i+2} & \\ But & \boxed{the} & \boxed{staff} & \boxed{\textbf{was}} & \boxed{so} & \boxed{horrible} & & to\ us. \end{array}$$

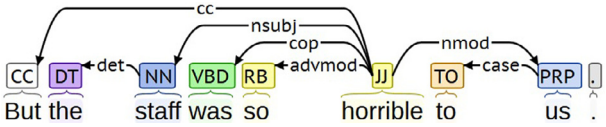
3. **Part-of-Speech (PoS) tag:** PoS information of the token provides useful evidence to identify aspect term. The potential aspect candidates correspond mostly to noun, adjective, verb or adverb. Here, we use PoS tags of the current and the surrounding tokens as feature.
4. **Head word:** We observe that significant percentage of constituent words that belong to the noun phrases have the chances of being an aspect term. We identify and implement a binary feature that fires if the current token is a head word of the noun phrase. A '0' value is assigned for the words that do not belong to a noun phrase. For example, in the following review *spicy tuna roll* and *asian salad* are noun phrases, whereas *roll* and *salad* denote the two head words, respectively.
   - Review: Best ***spicy tuna roll***, great ***asian salad***.
   - HeadWord: 0 0 0 1 0 0 0 1 0
5. **Head word PoS:** PoS of the head word is used as a feature of the model.
6. **Chunk information:** Many aspect terms are multiword in nature. For example, *"battery life", "spicy tuna rolls"* etc. We use chunk information that provides useful evidence to identify the boundaries of aspect terms. An example is shown in the following:

Review:	Boot	time	is	super	fast	.
Aspect:	B-ASP	I-ASP	O	O	O	O
Chunk:	NP		VP	ADJP		O

   Most of the tokens of an aspect term belong to noun phrases. However, some of the constituents in an aspect term could denote instances that belong to the phrases like VP, PP etc. In the example shown below, "getting a table" is an aspect term and the corresponding chunk information is VP (getting) and NP (a table), respectively.

Review:	Good	luck	getting	a	table	.
Aspect:	O	O	B-ASP	I-ASP	I-ASP	O
Chunk:	NP		VP	NP		O

7. **Lemma:** We use lemmas of the words as features. For example, the words like *serve, serves, served* and *serving* in restaurant domain can be identified as different inflectional forms of the word *serve*.
8. **Stop word:** In general, stop words cannot be the aspect terms (e.g., *the, is, at* etc.). A feature is defined that takes the value equal to 1 or 0 depending upon whether it is a stop word or not.
9. **Word length:** Length of a token may be effective in identifying the aspect terms. We observe that aspect terms are generally longer in length. We define a binary-valued feature that is set to high if the length of the candidate token exceeds a predetermined threshold. In our case we assume the token to be an aspect term if its length is more than 5 characters.
10. **Prefix and suffix:** Prefix and suffix of fixed-length character sequences are stripped from each token and used as the features of classifier. Here, we use prefixes and suffixes of length up to three characters of the current word as features.
11. **Frequent aspect term:** We extract and complie a list of frequent aspect terms from the training dataset. A binary-valued feature is then defined that fires if and only if the current token appears in this list. Here, the threshold is set to 5.
12. **Dependency relation:** Grammatical relationship among the words in a sentence can be represented by the dependency relations. We define two different dependency relation features in our work. One denotes the relation in case the current token is the *governor* (i.e head of the relation), while the other represents the relation if it is the *dependent*. For the first feature we look for the dependency relations of types: 'amod'(adjectival modifier), 'nsubj'(nominal subject) and 'dep'(dependent). The second feature corresponds to the relation types: 'nsubj'(nominal subject), 'dobj'(direct object) and 'dep'(dependent). Let us consider the following example review.



   It contains an aspect term *staff*. The token *staff* is dependent on *horrible* via relation 'nsubj'. No other above mentioned relations (neither governor nor dependent) are present for the token *staff*. Therefore, only the feature that corresponds to *dependent* 'nsub' will fire. Stanford dependency parser[1] is used to extract the dependency relations from a sentence. These features are defined in line with [36].
13. **WordNet:** In WordNet [37], different words that are semantically similar (or synonymous to each other) are categorized into synsets. Synset information as a feature enables the model to group tokens with identical senses. For example, the tokens *lunch* and *dinner* are related as the homonyms of *meal* in the WordNet hierarchy. This feature is particularly very crucial in identifying an unseen aspect term whose synonyms are present in the training set. We consider only the noun synsets. We define this feature following the one as mentioned in [36].
14. **Named entity information:** As per definition, only attribute of a product can be tagged as aspect term and not the product itself. Therefore, a named entity (NE) is not treated as an aspect term. Also, some tokens (which are normally a part of an aspect term) can not be considered as an instance of aspect term if they belong to any NE. For example in Table 2, a token *'sushi'* is present in both review sentences. It is an aspect term in the first case (***an attribute of a restaurant***). However, in second sentence it is not treated as an aspect term because it belongs to a NE *'Go Sushi'* (***a restaurant name***). We, therefore, define and use a binary-valued feature, which fires when a token is part of a NE.

[1] http://nlp.stanford.edu/software/lex-parser.shtml.

Table 2
Named entity information and aspect term relation.

Review: Certainly not the best ***sushi*** in New York.												
NE:	0	0	0	0	0	0	0	0	0			
Aspect terms:	O	O	O	O	B-ASP	O	O	O	O			
Review: I trust the people at Go ***Sushi***, it never disappoints.												
NE:	0	0	0	0	0	1	1	0	0	0	0	0
Aspect term:	O	O	O	O	O	O	O	O	O	O	O	O

15. **Character n-grams:** Character n-gram is a contiguous sequence of n characters extracted from a given word by striping off few characters from the beginning and/or end positions. We extract character n-grams by varying n in the range of 1–5 and use these as features of the classifier.
16. **Aspect term list:** For each domain (i.e. restaurant & laptop) we compile two different lists of aspect terms from the respective training set.
    (a) The first list contains the aspect terms that appear more than a predefined threshold count (here, f_1) in the training set.
    (b) The second list is created in order to handle the multi-word aspect terms. At first we compile single-word aspect terms whose counts are above a predefined threshold f_2 in the training data. Then, a probability p is computed in line with [36] for each word in the collection. The list is then compiled by selecting only those aspect terms whose corresponding probabilities are above a certain threshold (say, θ).

    Two binary-valued features are defined that take the value of 0 and 1 depending upon whether the current word appears in the compiled lists or not. As a result of cross validation we set f_1, f_2 & θ as 5, 5 & 0.7 respectively.
17. **Word cluster:** Brown clustering [38] is a form of hierarchical clustering technique which assigns words to a cluster based on the contexts in which they appear. We employ Brown clustering algorithm on two separate in-domain unlabeled datasets, namely Yelp[2] for the restaurant domain and Amazon[3] reviews for the laptop domain. Yelp dataset consists of 990,605 reviews, whereas Amazon dataset comprises of 88,414 reviews. A feature vector of length five (binary-valued) is defined that denotes the cluster identifier. An example is depicted below:

Review:	But	the	staff	was	so	horrible	to	us	.
ClusterId:	00101	1010	11010	01100	00101	11101	01110	01111	00111

18. **Semantic orientation (SO) score:** For each word, sentiment orientation (SO) score [39] is computed that measures how much it is associated with the positive or negative sentiments. Point-wise Mutual Information (PMI), a measure of association of token t with respect to positive or negative review, is used to determine the sentiment score as follow:

$$SO(t) = PMI(t, posRev) - PMI(t, negRev)$$

and

$$PMI(t, negRev) = \log \frac{freq(t, negRev) * N}{freq(t) * M}$$

where $freq(t, negRev)$ is the frequency of word t in negative review, $freq(t)$ is frequency of t in the corpus, M is the number of tokens in negative review and N is the number of tokens in the corpus. Similarly, $PMI(t, posRev)$ is the PMI scores with respect to the positive review. A positive SO score implies that the token is more inclined to the positive than negative reviews. We compute SO scores on the training sets of the respective domains, and used them as features in our work.
19. **Orthographic features:** We define two features based on the constructions of words. These check whether the token starts with a capitalized letter or starts with a digit. We observe that many aspect terms are capitalized and contains numeric symbols.

3.2. Sentiment classification

User's opinion expressed in a review are classified into the following semantic classes i.e. *positive, negative, neutral* and *conflict*. For sentiment classification we directly adapt few features used for aspect term extraction (e.g., local context, sentiment orientation score etc.). Along with these we define and implement some other problem specific features, as listed below, for the task at hand.

1. **Aspect term and its context**: Surface form of an aspect term along with its lower case form are used as features. As sentiment of a review heavily depends on the context where an aspect term appears, we use five tokens to the left and right as the local context information (i.e. a context word window of size 11 including the current one).
2. **Sentiment lexicon**: Sentiment lexicons are the important and useful sources for analyzing the opinions. Based on the following lexicons we extract few features:
    (a) **MPQA lexicon**: MPQA subjectivity lexicon [40] categorizes word sentiment into positive, negative and neutral classes. It contains 8,000 words along with its corresponding sentiment. For each token we define the polarity score to be 1, −1, 0 for a positive, negative and neutral token. respectively. For the token that does not appear in this lexicon a score of 2 is assigned. An integer-valued feature is then defined that computes the sum of polarity scores of all the terms that appear in the context window of size five.
    (b) **Bing Liu lexicon**: Bing Liu lexicon [41] is a list of positive and negative sentiment words. This lexicon contains approximately 6,800 words. Similar to the process used in MPQA lexicon we assign the following scores: 1 for a positive token, −1 for negative token and 2 for the token that does not appear in the lexicon. Based on this configuration, we define the following two features:
        (1) **Bing:** Polarity score of all the tokens that fall into the context window of a target aspect term are summed up and used as a feature. Size of the context window is set to five.
        (2) **Bing Direct:** In this feature we compute the sum of the polarity scores of only those words which have a *direct dependency relation* with the target aspect term.
    (c) **SentiWordNet lexicon**: This is one of the most popularly used lexicons for sentiment analysis. SentiWordNet[4] [42] lexicon is based on WordNet that assigns sentiment

[2] http://www.yelp.com/.
[3] http://snap.stanford.edu/data/other.html.
[4] http://sentiwordnet.isti.cnr.it/.

score of positivity and negativity to each synset. The sentiment scores of all the words that appear in the surrounding context of previous five and next five words of the target aspect term are retrieved and summed up. The value obtained as a result of this is used as a feature.

3. **Domain-specific words**: All the above lexicons are generic in nature and do not cover many domain specific words that express specific sentiments. Some of these examples are *'mouth watering'*, *'yummy'* and *'over cooked'* for the restaurant domain. We manually compile a lexicon of such words from the web[5] and from the general intuition. Following the same scoring method (1: positive, −1: negative and 2: words that don't appear), we compute sum of all the scores for the words that appear in the context of size 5.

4. Proposed method

In this section we describe our proposed method of PSO based feature selection and ensemble construction. At first we determine the best fitting feature sets for aspect term extraction and sentiment classification. As base classifiers we use three classification models, namely CRF, SVM and ME. The proposed method of ensemble selection automatically determines the best set of classifiers, that when combined together using a PSO based ensemble, improves the classification performance most. The entire process can be outlined as a sequence of three steps as follows:

1. Identification and Implementation of a diverse and rich feature sets for aspect term extraction and sentiment classification;
2. PSO based feature selection that yields a set of solutions for each classifier; and
3. Ensemble construction using a PSO to combine the outputs of classifiers.

These steps are generic in nature and can be adapted to any other application domain. In our current paper we evaluate our proposed techniques for two different problems, namely aspect term extraction and sentiment classification. For each of these two problems we use the reviews from two domains, namely restaurant and laptop. The schematic diagram of the proposed method is depicted in Fig. 1. The features that we use for our tasks cover lexical, syntactic as well as semantic level information. The PSO based feature selection is single objective optimization (SOO) in nature, where we optimize only one function. We choose the most relevant features in such a way, that when the classifiers are trained with these particular combinations, maximize the objective functions. For aspect term extraction we optimize F-measure, whereas for sentiment classification we optimize the classification accuracy. Output of this process is a set of vectors, each of which corresponds to a particular feature combination. We choose a set of promising models based on their effectiveness (i.e. based on F-measure, recall and precision, or accuracy values). We select the best models in two different ways: 3N effective classifiers (N each from a particular classification technique) are selected based on F-measure values; and then 3N models are selected in such a way that half of these are promising with respect to recall and half are with respect to precision. Our proposed method of ensemble construction operates in two steps: first step of which selects the most eligible candidate models (out of N as described above) for combination; and in the second step these are combined using majority or weighted voting approach. The best ensemble is obtained by optimizing F-measure or accuracy depending upon whether the problem deals with aspect term extraction or sentiment classification. Entire scheme is represented in the form of an algorithm, as mentioned below. In subsequent sections we present more detailed discussions on these steps.

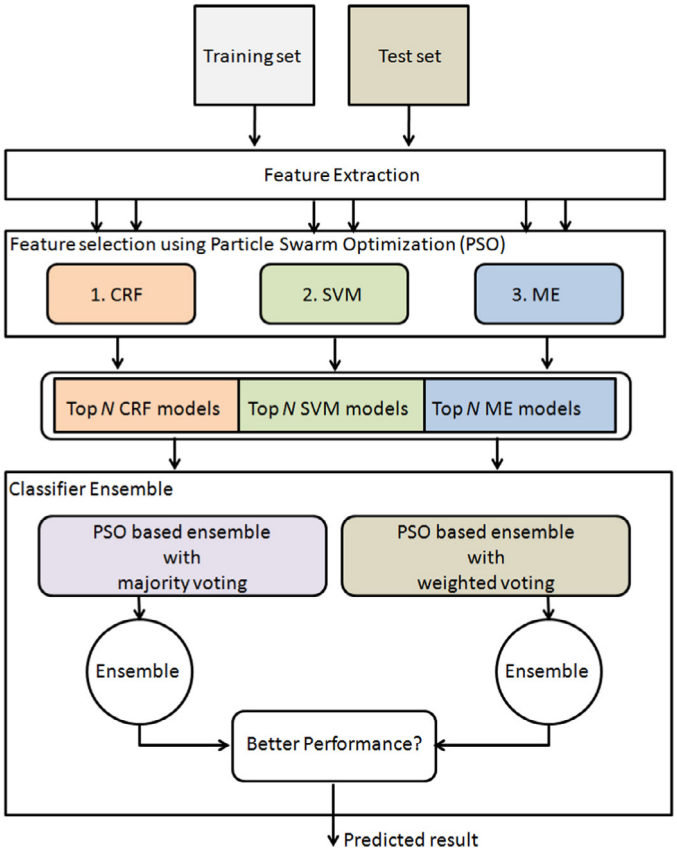


Fig. 1. Proposed methodology.

Algorithm 1 $Classified_{Test}$ = PSO_Asent($Feature_{Train}$, $Feature_{Dev}$, $Feature_{Test}$, c_1, c_2, c_3, N)

Input: $Feature_{Train}$, $Feature_{Dev}$, $Feature_{Test}$ - Feature files, c_1, c_2, c_3 - Three classifiers corresponds to CRF, SVM & ME, N - Top N models.

Output: $Classified_{Test}$ - Final classified output.

For each classifiers c_i **do**
  $Models_{c_i} \leftarrow$ PSO($Feature_{Train}$, $Feature_{Dev}$, c_i)
  $Top_{c_i} \leftarrow$ GetTopModels($Models_{c_i}$, N)
end for
$Candidates_{Classifier} \leftarrow$ FeatureVector(Top_{c_1}, Top_{c_2}, Top_{c_3}).
$BestCandidates_M \leftarrow$ PSO($Candidates_{Classifier}$, $MajorityVoting$)
$BestCandidates_W \leftarrow$ PSO($Candidates_{Classifier}$, $WeightedVoting$)
$EnsembleModel_M \leftarrow$ BuildEnsembleModel($BestCandidates_M$)
$EnsembleModel_W \leftarrow$ BuildEnsembleModel($BestCandidates_W$)
$Model_{Ensemble} \leftarrow$ BetterPerformance($EnsembleModel_M$, $EnsembleModel_W$)
$Classified_{Test} \leftarrow$ PredictClass($Model_{Ensemble}$, $Feature_{Test}$)

4.1. Feature selection using PSO

We develop our feature selection technique using a binary version of PSO. Basic steps of the proposed approach are described as below:

4.1.1. Particle encoding and initialization:

Potential solutions $\vec{X}(i)$, *particles* in PSO, are initialized with a fixed-length binary valued-string of 0's & 1's. Mathematically, *par-*

[5] http://world-food-and-wine.com/describing-food.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10} ($n = 10$)
$Particle_1$	0	0	1	1	0	1	0	1	1	0
$Particle_2$	1	0	1	1	0	0	0	1	0	1
$Particle_3$	1	1	0	0	1	1	1	0	1	1
$Particle_4$	0	0	1	0	1	0	1	1	0	0

($N = 4$)

Fig. 2. Particle encoding.

ticles can be formulated as:

$$\vec{X}(i) = (x_{(i,1)}, x_{(i,2)}, ..., x_{(i,n)})$$

where $x_{(i,j)} \in \{0, 1\}$, $i = 1, 2, ..., N$, N is the number of particles. The length of each particle (n) is equal to the number of features present. Each bit position of a particle corresponds to a feature. The value of this bit denotes whether the respective feature participates in classifier's training or not. A feature f_j is used for classifier's training if and only if its corresponding bit position j contains a value of '1', otherwise, it is not considered for training. Fig. 2 represents an example of particle representation for $N = 4$ and $n = 10$. In $Particle_1$, only five features i.e. f_3, f_4, f_6 f_8 and f_9, have the bit values 1 and hence only these participate in classifier's training. Similarly $Particle_2$ encodes a feature combination where f_1, f_3, f_4, f_8 and f_{10} are considered for classifiers' training.

At the beginning, a fixed number of N particles are initialized and encoded in the swarm. Encoding process is described as follows:

$$x_{(i,j)} = \begin{cases} 1, & \text{if} \quad \mu \geq 0.5 \\ 0, & \text{Otherwise} \end{cases}$$

A uniform random number μ on the interval (0, 1) is generated for each bit position $x_{(i,j)}$ of $\vec{X}(i)$ and if μ is greater than a threshold of 0.5 we choose the bit value as '1', otherwise it is initialized with '0'.

4.1.2. Updating the global and best position value:

At the very beginning, previous best position of the particle $\vec{X}(i)$, represented by $\vec{B}(i)$, is set to null. As the initial particle is generated, we set the value of $\vec{B}(i)$ to the position vector of the particle $\vec{X}(i)$. At each iteration the best position vector $\vec{B}(i)$ is updated based on the fitness function (or, objective function). In our case the fitness function is the F-measure value of the classifier trained using the feature combination as represented in the particle $\vec{X}(i)$. If the current position vector $\vec{P}(i)$ is better than its best position vector $\vec{B}(i)$ with respect to the fitness value, we update the best position by setting it to the current position (representing the best position, the particle has seen so far). Otherwise, it remains unaltered. This means that if $f(\vec{P}(i)) > f(\vec{B}(i))$, update the value of $\vec{P}(i)$. Here, the fitness value is computed to be equal to the F-measure value of the classifier. For the global best position, i.e. for $\vec{G}$ we also follow the same process for updating. Initially, the global best position is set to empty. The update only happens after all the values of $\vec{B}(i)$ are determined. The value of $\vec{G}$ is set to the fittest $\vec{B}(i)$ found so far. It is updated only when the fittest solution represented by $f(\vec{B}(i))$ in the swarm is superior to $f(\vec{G})$.

4.1.3. Updating the velocities:

Every particle $\vec{X}(i)$ in the swarm has an associated velocity vector $V(i) = (v_{(i,1)}, v_{(i,2)}, ..., v_{(i,n)})$. Rate of change of $x_{(i,j)}$ in $\vec{X}(i)$ is governed by the corresponding element $v_{(i,j)}$ in $V(i)$. Each element $v_{(i,d)}$ in $V(i)$ is updated following the process as mentioned below [43]:

$$v_{(i,j)} = w * v_{(i,j)} + \mu_1(b_{i,j} - x_{(i,j)}) + \mu_2(g_j - x_{(i,j)})$$

where $w(0 < w < 1)$, μ_1 and μ_2 are known as inertia weight, cognitive and social scaling parameters, respectively. The $b_{(i,j)}$, $x_{(i,j)}$, $g_{(j)}$ denote the j^{th} components of $\vec{B}(i)$, $\vec{X}(i)$ and $\vec{G}$, respectively. The concept of inertia weight was not there in the original version of PSO [44]. It was first introduced by Shi et al. [45] to better control the exploration and exploitation of the particles.

4.1.4. Generating new particles:

For each particle $\vec{X}(i)$ and each bit position $x_{(i,j)}$ in $\vec{X}(i)$ can be either 0 or 1 based on the following criteria:

$$x_{(i,j)} = \begin{cases} 1 & \text{if } r > \xi(v_{(i,j)}) \\ 0 & \text{otherwise} \end{cases}$$

where $0 \leq r \leq 1$ is a uniform random number and

$$\xi(v_{(i,j)}) = \frac{1}{1 + \exp\left(-\vec{v_{(i,d)}}\right)}$$

4.2. PSO based ensemble construction

It is a fact that a single learning algorithm is not always sufficient to produce the acceptable performance whenever the application domain is altered. If a learner L shows good accuracy for any domain m, it is not guaranteed that it will show similar performance for any other domain n as well. In classifier ensemble, several classifiers are combined together to produce the final output. For many applications the approaches were proved to be useful. The feature selection step described earlier yields a set of solutions for each of the learning algorithms, i.e. MEMM, CRF and SVM. Each solution represents a particular feature combination, which is used to construct a classification model. Our proposed PSO based ensemble selection method finds a good subset of models, and combine them together using majority and weighed voting techniques. The basic steps that we describe earlier in Section 4.1 also apply here. The notable difference is in the step of problem encoding, where each particle represents a set of individual classifiers in this case. As an example, for a given set of classifiers, $C = (c_1, c_2, c_3, c_4, c_5)$ and N=3, the swarm can be represented as follows:

$$\vec{X}(1) = (1, 0, 1, 0, 1)$$
$$\vec{X}(2) = (1, 0, 1, 1, 0)$$
$$\vec{X}(3) = (1, 1, 1, 0, 0)$$

Here for the first particle, three classifiers, namely c_1, c_3 and c_5 are chosen as the candidates for constructing the ensemble as only these positions have the values of 1.

Once the eligible classifiers are identified, these are combined using either majority or weighted voting. In majority voting we analyse the outputs of several classifiers, and finally assign the class label that has the maximum occurrences. Thus, we always choose the particular class for which majority of the classifiers agree. In case of weighted voting, rather than uniform weights we assign different weights depending upon the strengths or weaknesses of the classifiers. For an instance, we add the weights of those classifiers that predict the same class. Finally, we assign the

Table 3
Classifier ensemble using majority and weighted voting.

Classifier (C_i) & Weight (f_i)					Ensemble	
C_1	C_2	C_3	C_4	C_5	Majority	Weighted
$f_1 = 0.60$	$f_2 = 0.60$	$f_3 = 0.90$	$f_4 = 0.92$	$f_5 = 0.61$		$max(\Sigma^5_{i=1,x} f_i, \Sigma^5_{i=1,y} f_i)$
x	x	y	y	x	x	y
x	x	x	y	x	x	x
y	y	x	y	y	y	y
x	y	x	x	x	x	x
y	y	y	x	x	y	y

class that has the highest weighted vote. The weight is the F-measure or accuracy of the classifier. It is to be noted that F-measure or accuracy corresponds to the fitness value of the classifier (feature selection) or ensemble.

In Table 3, we provide an example for two kinds of voting schemes. Suppose, for a two-class ('x' and 'y') problem we have the five classifiers denoted by C_i, i = 1...5, along with their respective weights f_i. Predicted class labels of each classifier are shown in the respective columns. For the first instance, predicted class labels are 'x', 'x', 'y', 'y' & 'x' respectively. Three classifiers i.e. C_1, C_2 & C_5 have predicted 'x' whereas two classifiers C_3 & C_4 have predicted 'y' as the class labels. Since the count of class 'x' is higher than the count of 'y', majority voting technique picks 'x' as the final class label. However, in case of weighted voting technique, class label 'y' is chosen because weighted sum of 'y' class ($f_3 + f_4 = 1.82$) is greater than the weighted sum of class 'x' ($f_1 + f_2 + f_5 = 1.81$).

4.2.1. Discussion on optimization using PSO:

After discussing various aspects of PSO, we try to realize the processes that it follows to optimize the feature set and the candidates for classifier ensemble.

Feature Selection: Every token in the training or test dataset is represented by a vector of length n, which is equal to the length of a particle in PSO, i.e. each bit position of a particle corresponds to exactly one feature. Once a particle is initialized, as defined in Section 4.1.1, it represents a subset of features whose corresponding bit positions are '1'. We consider this feature subset as an input for classifier's training and its evaluation. Hence, each particle has an associated fitness value, say d_i^t for i_{th} particle, at iteration t. In the next iteration $t+1$, a new set of particles is generated and evaluated. At the end of each iteration t, PSO selects a particle p (a.k.a global solution) that has the highest fitness value seen up to the t^{th} iterations. Feature set represented by particle p is the optimized feature set up to the tth iterations. We continue this process up to the maximum number of iterations, and on termination PSO yields a set of near-(optimal) solutions. Table 4 shows one such example. It depicts 5 particles, showing the encoding of two iterations (t & $t+1$) along with their corresponding fitness values. Particle P_1 at iteration t selects *word, next, stop* & *PoS* as elements of its feature set. We only use these 4 features for training the models. Evaluation of the model yields the fitness value, i.e. $d_1 = 69.78\%$. Similarly, P_2 at iteration t makes use of *prev, next* & *PoS* as its feature set. Fittest particle and optimized feature set at the end of each iteration, i.e. t: P_4 and $t+1 : P_2$, are listed at the bottom of Table 4.

Classifier Ensemble: For classifier ensemble, length of a particle in the swarm is set equal to the number of classifiers that participate in the ensemble process. In contrast to feature selection, each bit position of the particle encodes a classifier. Particles are initialized following the same technique as discussed in Section 4.1.1. Presence of bit '1' in the particle selects the corresponding classifier's predicted output for ensemble process. Subsequently, outputs of the selected classifiers are combined and evaluated using majority or weighted voting techniques. Rest of the processes are similar to the PSO based feature set optimization. An example of PSO based classifier ensemble for sentiment classification is depicted in Table 5. At iteration t, five particles are initialized and combined using the selected candidates (corresponding to the bit positions having values of 1). Output of the combined model is then evaluated and assigned a fitness value d_i to each particle p_i. As similar to the feature selection technique, PSO finds out a set of near-(optimal) solutions at the end of each iteration.

5. Datasets, experiments and analysis

In this section we first provide a brief description of the datasets that we use for our experiments, and then report the experimental results along with necessary analysis.

5.1. Datasets

We use the benchmark datasets of SemEval-2014 shared task for our experiments. The dataset consists of user generated reviews from two domains, namely restaurant and laptop. The restaurant dataset comprises of 3,044 user reviews in the training set while gold test set contains 800 review instances. There are 3,045 & 800 user reviews present in the training and gold test datasets, respectively, for the laptop domain. Number of aspect terms present in the two training sets counts to 3,699 and 2,358, respectively. Brief statistics of the datasets are shown in Table 6.

5.2. Preprocessing:

At first the datasets are pre-processed to remove the XML tags, and extract the relevant bits of information. For this Stanford CoreNLP[6] suite is used to tokenize the reviews and extract various basic information such as lemma, Part-of-Speech (PoS) and named entity (NE) information. The aspect term can also be a multi-word token. In order to properly denote the boundaries of aspect term we use a IOB encoding scheme, where B, I and O denote the beginning, internal and outside tokens of aspect term.

5.3. Experimental results

As mentioned earlier, we use three robust classifiers, *viz.* CRF, SVM & ME, as our base learners. For implementation of these algorithms we use CRF++,[7] Yamcha[8] and Stanford ME classifier,[9] respectively. Evaluation of all the systems are performed following the SemEval-2014 evaluation script. Since context information provides crucial information in aspect based sentiment analysis, we define four baseline systems, i.e. $Baseline_1$, $Baseline_2$, $Baseline_3$ and

[6] http://nlp.stanford.edu/software/corenlp.shtml.
[7] http://taku910.github.io/crfpp/.
[8] http://chasen.org/~taku/software/yamcha/.
[9] http://nlp.stanford.edu/software/classifier.shtml.

Table 4
Representative feature pruning scenario; features whose values are '0' are pruned; fitness values are hypothetical.

			Features						Fitness Value
			Word	Prev	Next	WordLen	Stop	POS	d_i
Sample feature file			The	NULL	food	0	1	DT	–
			food	The	was	0	0	NN	
			was	food	good	0	1	VBD	
			good	was	but	0	0	JJ	
			but	good	service	0	1	CC	
			service	but	was	1	0	NN	
			was	service	poor	0	1	VBD	
			poor	was	.	0	0	JJ	
			.	poor	NULL	0	0	.	
Particle Swarm Optimization (PSO) based feature selection									
Particle P_i	$Iteration_t$	P_1	1	0	1	0	1	1	69.78
		P_2	0	1	1	0	0	1	70.68
		P_3	1	0	0	1	1	0	70.23
		P_4	1	0	1	0	0	1	71.24
		P_5	1	1	0	1	0	0	67.94
	$Iteration_{t+1}$	P_1	1	1	0	0	0	1	69.54
		P_2	1	0	1	0	1	1	72.87
		P_3	1	1	0	1	0	0	71.43
		P_4	0	1	1	0	1	0	67.98
		P_5	1	0	1	1	1	1	68.76
Fittest particle and optimized feature set at $Iteration_t$									
P_4			1	0	1	0	0	1	71.24
Features for P_4			The		food			DT	71.24
			food		was			NN	
			was		good			VBD	
			good		but			JJ	
			but		service			CC	
			service		was			NN	
			was		poor			VBD	
			poor		.			JJ	
			.		NULL			.	
Fittest particle and optimized feature set at $Iteration_{t+1}$									
P_2			1	0	1	0	1	1	72.87
Feature for P_2			The		food		1	DT	72.87
			food		was		0	NN	
			was		good		1	VBD	
			good		but		0	JJ	
			but		service		1	CC	
			service		was		0	NN	
			was		poor		1	VBD	
			poor		.		0	JJ	
			.		NULL		0	.	

$Baseline_{SemEval}$ for comparative analysis. In $Baseline_1$, we fix the context window as w_{i-2}^{i+2} (i.e. previous two, current and next two instances) and select all the features in that window for training. Similarly, $Baseline_2$ takes all the features in context window of w_{i-3}^{i+3} for training. $Baseline_3$ is based on PSO optimization. It is similar to $Baseline_1$ and $Baseline_2$ except context information was obtained from PSO. The baseline system, $Baseline_{SemEval}$ was defined by the organizers of the SemEval 2014 task. Below, we describe the baseline systems:

1. $Baseline_1$: This baseline system is developed by training with all the features mentioned in Section 3 within the local context window of w_{i-2}^{i+2} : all features of previous two (−2, −1), current (0) and next two (+1, +2) instances.
2. $Baseline_2$: This is similar to $Baseline_1$ *except* the context window w_{i-3}^{i+3} is considered i.e. all features of previous three (−3, −2, −1), current (0) and next three (+1, +2, +3) instances.
3. $Baseline_3$: This model is dependent on the best model obtained after we execute PSO. Context information, obtained through PSO based feature selection, is considered along with all the other features.
4. $Baseline_{SemEval}$: This fourth baseline is derived from SemEval-2014 shared task for both aspect term extraction and sentiment classification. These are defined as below:
   a. A list containing all the aspect terms of the training set is generated. A sequence of tokens is tagged as aspect term in a test sentence, if it appears in the list.
   b. For each aspect term of the test set, all the training sentences containing this aspect term are retrieved. The final sentiment class is then determined based on the most frequent class.

Results of the baseline models for each domain are reported in Table 7.

Parameter settings of PSO:

Different parameters of PSO guide its behavior and efficacy in optimizing the problem at hand. It has been shown in [46,47] that we can achieve satisfactory performance with PSO if we tune its parameters properly. Several ways for assigning the parameters have been discussed in [48]. We perform various experiments in order to choose the best fitting parameters for our problem. As a result, we found four different parameter settings that were (near)-optimal for the different tasks at hand. In order to maintain the uniformity we make use of all the four settings for the separate runs of PSO. Table 8 shows our choice of parameters for the ex-

Table 5
Representative classifier pruning scenario for the sentiment classification task; Classifier whose values are '0' are pruned; Fitness values are hypothetical.

Aspect terms			Classifier's Output					Ensemble Output	Fitness Value
			C_1	C_2	C_2	C_4	C_5	(Majority)	d_i
Boot time			positive	positive	neutral	positive	positive	positive	70.68
screen			neutral	negative	negative	neutral	negative	negative	
price			positive	neutral	negative	positive	positive	positive	
Windows 8			positive	positive	neutral	neutral	neutral	neutral	
battery life			negative	conflict	conflict	negative	conflict	conflict	
weight			negative	neutral	negative	neutral	neutral	neutral	
Particle Swarm Optimization (PSO) based classifier ensemble									
Particles P_i	$Iteration_t$	P_1	1	0	1	0	1		69.78
		P_2	1	0	1	1	0		71.24
		P_3	0	1	1	0	0		70.68
		P_4	1	0	0	1	1		70.23
		P_5	1	1	0	1	0		67.94
	$Iteration_{t+1}$	P_1	1	1	0	0	0		69.54
		P_2	1	1	0	1	0		71.43
		P_3	1	0	1	0	1		72.87
		P_4	0	1	1	0	1		67.98
		P_5	1	0	1	1	1		68.76
Fittest particle and optimized ensemble candidates at $Iteration_t$									
P_2			1	0	1	1	0		71.24
Boot time			positive		neutral	positive		positive	71.24
screen			neutral		negative	neutral		neutral	
price			positive		negative	positive		positive	
Windows 8			positive		neutral	neutral		neutral	
battery life			negative		conflict	negative		negative	
weight			negative		negative	neutral		negative	
Fittest particle and optimized ensemble candidates at $Iteration_{t+1}$									
P_3			1	0	1	0	1		72.87
Boot time			positive		neutral		positive	positive	72.87
screen			neutral		negative		negative	negative	
price			positive		negative		positive	positive	
Windows 8			positive		neutral		neutral	neutral	
battery life			negative		conflict		conflict	conflict	
weight			negative		negative		neutral	negative	

Table 6
Statistics of datasets.

Domain	Dataset	#Reviews	#Aspect	Sentiment Class			
				#Pos	#Neg	#Neu	#Conf
Restaurant	training	3044	3699	2164	805	633	91
	test	800	1134	728	196	196	14
Laptop	training	3045	2358	987	866	460	45
	test	800	654	341	128	169	16

periment. We denote these runs as: PSO_{Run_1}, PSO_{Run_2}, PSO_{Run_3} and PSO_{Run_4}. For each run we fix the number of particles and iterations as 50 and 100, respectively.

For each pair of instance and output label, feature vectors are generated. The classifiers are trained with these vectors, and PSO based feature selection technique is employed to find the most relevant set of features. In order to optimize the feature selection model, part of the training set was used as a development set in the work. Classifiers were trained on 90% of the training set while the development set, which constitutes of 10% of training set, was used for evaluating the performance of the trained models. Based on different parameter combinations, we execute PSO for four different runs as mentioned above. For a particular domain-classifier combination, we combine all these runs. Thereafter we select the top-most N models based on the F-measure measure (for aspect term extraction) or accuracy (for sentiment classification) value. Here we set the value of N as 20. It is also to be noted that, for aspect term extraction we select the models that show either good recall or precision values. This is done in order to choose the diverse classifiers (i.e. complimentary in nature), so that when they are combined, produces higher performance. The underlying assumption, being that the classifiers which perform good with respect to recall may not (in most cases) have the similar characteristics to those, performing good for precision. We assign these extracted models an unique name X_{Y_i} where $i = 1...N$, $Y \in \{p(recision), r(ecall), f(-measure)\}$ and $X \in \{M(EM), C(RF), S(VM)\}$. For example, C_{f_1} and C_{f_2} represent two CRF based models with highest F-measure values.

Here we show the results of only the top 5 systems, which were selected based on F-measure or accuracy value. Results of these models are reported in Table 9. Comparisons between the results of baselines and the models selected through PSO based feature selection (c.f. Table 7 and Table 9) show that we can achieve better performance if we are able to find out the most appropriate set of features. For aspect term extraction in the restaurant domain, best model of ME, M_{f_1}, obtains a F-measure of 72.86%. This is higher compared to all the other baseline models. For CRF and SVM we obtain the F-measures of 83.11% (C_{f_1}) and 81.76% (S_{f_1}), respectively, which are above all the baseline models. For sentiment classification we obtain the accuracies of 74.95%, 78.65% and 77.24% for ME,

Table 7
Results of baseline systems. Here, f_n is the total number of features for respective domain/classifiers.

	Baselines	Restaurant						Laptop					
		Aspect term				Sentiment		Aspect term				Sentiment	
		P	R	F	f_n	Acc	f_n	P	R	F	f_n	Acc	f_n
ME	$Baseline_1$	68.84	73.28	70.99	68	74.16	38	57.62	54.89	56.22	83	65.90	22
	$Baseline_2$	69.04	73.33	71.14		73.72		57.66	55.19	56.40		64.06	
	$Baseline_3$	70.24	73.89	72.02		73.63		57.44	54.89	56.13		65.13	
CRF	$Baseline_1$	79.94	76.19	78.02	68	77.33	38	78.40	60.59	68.35	83	69.57	22
	$Baseline_2$	80.33	76.64	78.44		76.71		78.86	59.97	68.13		69.11	
	$Baseline_3$	79.80	76.19	77.95		77.64		79.47	60.43	68.66		69.57	
SVM	$Baseline_1$	82.26	75.66	78.82	66	69.92	25	80.28	61.00	69.33	67	57.18	20
	$Baseline_2$	83.28	78.65	80.90		71.25		80.75	62.23	70.29		58.86	
	$Baseline_3$	83.13	78.65	80.83		74.71		81.78	64.52	72.13		64.67	
SemEval	$Baseline_{SemEval}$	–	–	47.15	–	64.28	–	–	–	35.64	–	51.37	–

Table 8
Various parameter settings of PSO.

Run	# Particle	# Iteration	Inertia weight(w)	μ_1	μ_2
PSO_{Run_1}	50	100	0.3593	−0.7238	2.0289
PSO_{Run_2}			0.7298	1.49618	1.49618
PSO_{Run_3}			−0.3699	−0.1207	3.3657
PSO_{Run_4}			−0.4349	−0.6504	2.2073

CRF and SVM, respectively, for the restaurant domain. In case of laptop domain the system yields the F-measure values of 59.39%, 72.75% and 72.78% for aspect term extraction. The system for sentiment classification on this dataset yields the accuracies of 66.81%, 72.17% and 66.97%, for ME, CRF and SVM, respectively. In Table 9, we also exhibit the number of features f_n that participates in classifier's training. It shows how we can achieve better performance even with the use of a much reduced sets of features. As an instance, the PSO based feature selection model only makes use of 35 (C_{f_1}) and 27 (S_{f_1}) features for the training of CRF while the domains are restaurant and laptop, respectively. This proves that, indeed, feature selection helps to obtain better performance with a less complex model that utilizes less number of features.

We show the experimental results in Table 10, where 10 best models are selected based on the good recall and precision values (5 each).

In Tables 11 and 12 we show the optimized feature sets as determined by PSO for aspect term extraction for the restaurant and laptop domains, respectively. It shows that some of the features, e.g. context information, PoS tag, chunk, word cluster, WordNet synsets, dependency relation, named entity etc., have greater impact than the others. In Table 13, we present the optimized feature sets for sentiment classification, which clearly suggests that the use of sentiment lexicons is the primary cause in achieving good accuracy.

5.4. Results of classifier ensemble

Models extracted in the previous subsection are chosen as the potential candidates for constructing classifier ensemble. In order to find the best candidates for ensemble construction we employ the PSO based method that we described in the earlier section. Similar to feature selection approach we keep the same set of parameter combinations. The best set of classifiers obtained are combined using both majority voting and weighted voting techniques. Evaluation results are reported in Table 14 that shows the effectiveness of the weighted voting technique ($En_{Weighted}$) over majority voting ($En_{Majority}$) for all the cases. The proposed ensemble achieves the F-measure scores of 84.52% and 74.93% for aspect term extraction for restaurant and laptop domains, respectively. For sentiment classification we obtain the accuracies of 80.07% and 75.22% for restaurant and laptop domains, respectively. It is clearly evident that ensemble has been effective with considerable performance improvement (approximately 2% increase on an average) as a result of PSO based classifier selection process. In Table 15 we present the optimal subsets of classifier candidates which are used in final ensemble construction. Results show that ME based classifiers have the least contributions in comparison to CRF or SVM for all the

Table 9
Results of top 5 models obtained through PSO based feature selection (w.r.t F-measure).

Classifiers	Models	Restaurant						Laptop					
		Aspect term				Sentiment		Aspect term				Sentiment	
		P	R	F	f_n	Acc	f_n	P	R	F	f_n	Acc	f_n
ME	M_{f_1}	71.44	74.33	**72.86**	38	**74.95**	20	61.43	57.49	**59.39**	41	**66.81**	13
	M_{f_2}	70.78	74.77	72.72	41	74.69	20	60.25	57.95	59.08	39	66.20	12
	M_{f_3}	70.24	74.95	72.52	31	74.60	17	61.38	56.88	59.04	46	66.05	14
	M_{f_4}	70.53	74.51	72.47	41	74.51	18	60.48	57.33	58.86	48	65.90	16
	M_{f_5}	71.17	73.80	72.46	37	74.42	23	60.61	57.18	58.85	46	65.74	13
CRF	C_{f_1}	85.39	80.95	**83.11**	35	**78.65**	16	83.39	64.52	**72.75**	44	**72.17**	11
	C_{f_2}	85.63	80.42	82.94	39	78.48	18	83.83	64.22	72.72	42	72.01	11
	C_{f_3}	85.76	80.46	82.91	31	78.39	20	83.23	64.52	72.69	40	71.71	10
	C_{f_4}	85.03	82.68	82.80	32	78.21	23	83.49	64.22	72.60	44	71.55	17
	C_{f_5}	84.68	80.95	82.77	34	78.13	16	83.20	64.37	72.58	46	71.40	13
SVM	S_{f_1}	83.53	80.07	**81.76**	29	**77.24**	11	81.99	65.44	**72.78**	27	**66.97**	11
	S_{f_2}	83.54	79.71	81.58	30	76.89	11	82.17	64.83	72.47	32	66.81	13
	S_{f_3}	83.41	79.80	81.56	37	76.80	10	83.90	63.76	72.45	30	66.66	09
	S_{f_4}	83.87	79.36	81.55	43	76.71	18	82.29	64.67	72.43	31	66.51	09
	S_{f_5}	83.25	79.80	81.49	35	76.63	12	82.01	64.83	72.41	34	66.36	11

Table 10
Result of top 5 models obtained through PSO based feature selection (w.r.t precision and recall).

Classifiers	Metrics	Models	Aspect Term							
			Restaurant				Laptop			
			P	R	F	f_n	P	R	F	f_n
ME	Precision	M_{p_1}	**71.44**	74.33	72.86	38	**61.43**	57.49	59.39	41
		M_{p_2}	71.17	73.80	72.46	37	61.38	56.88	59.04	39
		M_{p_3}	70.90	73.72	72.28	34	61.01	56.72	58.79	45
		M_{p_4}	70.78	74.77	72.72	41	60.82	56.26	58.45	46
		M_{p_5}	70.69	73.80	72.21	37	60.75	56.57	58.59	44
	Recall	M_{r_1}	70.24	**74.95**	72.52	30	59.46	**58.10**	58.77	43
		M_{r_2}	70.78	74.77	72.72	41	60.25	57.95	59.08	46
		M_{r_3}	70.03	74.60	72.24	37	61.43	57.49	59.39	41
		M_{r_4}	70.53	74.51	72.47	41	60.48	57.33	58.86	47
		M_{r_5}	70.09	74.42	72.19	39	60.61	57.18	58.85	48
CRF	Precision	C_{p_1}	**85.90**	79.54	82.60	32	**84.24**	63.76	72.58	46
		C_{p_2}	85.76	80.24	82.91	31	83.83	64.22	72.72	42
		C_{p_3}	85.63	80.42	82.94	39	83.70	63.60	72.28	50
		C_{p_4}	85.57	80.07	82.73	43	83.49	64.22	72.60	44
		C_{p_5}	85.49	80.07	82.69	35	83.40	63.76	72.27	44
	Recall	C_{r_1}	85.29	**80.95**	83.11	35	82.29	**64.67**	72.43	35
		C_{r_2}	85.03	80.68	82.80	32	83.23	64.52	72.69	40
		C_{r_3}	84.61	80.51	82.51	38	83.20	64.37	72.58	46
		C_{r_4}	85.63	80.42	82.94	39	83.49	64.22	72.60	44
		C_{r_5}	85.76	80.24	82.91	31	82.48	64.06	72.11	45
SVM	Precision	S_{r_1}	**84.10**	78.83	81.38	28	**83.90**	63.76	72.45	30
		S_{r_2}	83.99	78.65	81.23	33	83.83	63.45	72.23	35
		S_{r_3}	83.97	79.01	81.41	33	83.06	63.76	72.14	30
		S_{r_4}	83.94	78.83	81.30	38	82.71	64.37	72.39	35
		S_{r_5}	83.91	79.10	81.43	36	82.44	63.91	72.00	37
	Recall	S_{r_1}	83.53	**80.07**	81.76	29	81.99	**65.44**	72.78	26
		S_{r_2}	82.96	78.89	81.40	37	80.83	65.13	72.14	36
		S_{r_3}	83.25	79.80	81.49	35	82.17	64.83	72.47	32
		S_{r_4}	83.54	78.71	81.58	30	82.29	64.67	72.43	31
		S_{r_5}	83.36	79.54	81.41	33	82.10	64.52	72.26	34

problems. Contribution of SVM and CRF based models are comparable to each other.

5.5. Comparisons with the existing systems

Our proposed system performs convincingly better with respect to the baseline models that we defined. In order to further establish the effectiveness of our proposed approach we provide comparisons with some other state-of-the-art models as well. Comparisons with the existing systems are reported in Table 16. In SemEval-2014 shared task, the best performing systems for aspect term extraction exhibit the F-measure values of 84.01% for the restaurant domain [49] and 74.55% for the laptop domain [16]. Both of these systems were developed based on CRF. This is lower in comparison to what we obtain,i.e. 84.52% and 74.91% for the restaurant and laptop domains, respectively. It is to be noted that the CRF based model developed in [49] makes use of additional external resources and large amount of unlabeled data to generate word clusters, which were used as the features. The CRF based system of [16] incorporates more extensive additional resources and a rule-based sentiment analysis module to improve the performance. The most distinctive characteristics of our present work compared to these two is that we don't make use of any heavy domain-specific resources and/ or tools. However, due to the use of systematic approaches for feature selection and classifier ensemble, we obtain better accuracies with much reduced sets of features. In comparison to the system of Liu et al. [18] for aspect term extraction, our proposed model achieves more than 2% increase in F-measure value for the restaurant domain. For laptop domain, [18] reported 75.00% F-measure value using Long-Short-Term-Memory (LSTM) along with an extra set of features. In sentiment classification our system for aspect term extraction obtains F-measure values of 80.07% and 74.46% for the restaurant and laptop domains, respectively. For sentiment classification, the best system [17] of the shared task reports the accuracies of 80.95% and 70.49% for these two domains. However, this is to be noted that the method proposed in [17] was based on SVM that made use of some extra features extracted from the bag-of-word concept, rule-based system, and combined lexicons (BingLiu, SentiWordNet, MPQA). Like aspect term extraction task we also achieve better performance with less number of features. Our system also performs convincingly better as compared to Kaljahi et al. [20] who uses tree-kernels based technique for the sentiment classifications.

The current work is an extension of our earlier work reported in [33], where a feature selection approach is developed for CRF classifier. The system proposed in the current work clearly performs better than our previous systems proposed in [33]. The reasons behind this better performance are due to better re-implementation of some of the previous features, implementation of some new features, use of three different classification techniques and the PSO based classifier ensemble technique. We also present detailed experiments, thorough analysis of the results and error analysis. It should be noted that our system uses considerably less number of features and external resources as compared to the existing systems. Hence, the complexity of our proposed model is lower compared to the others.

We also perform Analysis of variance (ANOVA) [50] to measure the statistical significance of the results obtained. For this the algorithm was executed 10 times. It was observed that differences (proposed model vs. existing state-of-the-art systems) in mean F-measure are statistically significant as p value is less than 0.05 in each case.

If $P =$ #particles, $I =$ #iterations, $F_v =$ *#1's in a particle on avg* and $\xi =$ *model training time*, then the time complexity of the proposed algorithm, i.e. PSO based feature selection and

Table 11
Optimized feature sets for aspect term extraction task in restaurant domain.

Models	Features																	
	Word & context	PoS	Head Word	Chunk	Lemma	Stop Word	Word Length	Prefix	Suffix	Frequent aspect term	Dependency Relation	WordNet	Named Entity	Char n-grams (1,2,3,4,5)gram	Word cluster	Aspect term list	Sentiment Orientation	Orthographic c: IsCap, d: InitDigit
M_{f_1}	−2..+3	√	√	√	–	√	–	–	√	–	√	√	√	1,3,4	√	√	√	–
M_{f_2}	−2..+2	√	√	–	–	–	–	–	–	–	–	√	√	2,3,4,5	√	–	√	–
M_{f_3}	−1..+2	√	–	–	–	–	–	√	√	–	–	√	–	3,4,5	√	–	–	–
M_{f_4}	−3..+2	√	–	√	–	√	–	–	√	–	√	√	√	1,3,4,5	√	√	–	–
M_{f_5}	−3..+2	√	√	√	–	–	–	√	√	√	–	–	√	1,2,5	–	√	–	–
C_{f_1}	−2..0	–	√	–	–	√	–	–	√	–	√	√	√	2,5	√	–	√	–
C_{f_2}	−2..+3	–	√	–	–	–	–	√	√	√	√	√	√	2	√	–	√	–
C_{f_3}	−2..+3	√	–	√	–	√	–	–	–	–	–	√	–	3,5	√	–	–	–
C_{f_4}	−2..0	√	√	√	–	–	–	–	√	–	–	√	–	3,4,5	√	–	√	–
C_{f_5}	−2..+2	√	–	–	–	–	–	–	√	–	√	√	–	3	√	–	√	–
S_{f_1}	−1..+1	–	√	–	–	–	√	–	–	–	√	√	√	1,5	√	√	–	–
S_{f_2}	−1..+2	√	–	√	–	√	–	√	√	√	√	–	√	1	√	√	–	–
S_{f_3}	−1..+2	√	√	√	–	√	–	√	√	√	–	√	√	1,2,4	√	–	√	–
S_{f_4}	−1..+2	–	√	√	–	√	√	√	–	√	√	–	√	√	√	–	–	–
S_{f_5}	−1..+1	√	√	–	–	–	√	–	–	√	√	√	√	1,3,4	√	√	√	–
M_{p_1}	−2..+3	√	√	√	–	√	–	–	√	–	√	√	√	1,3,4	√	√	√	–
M_{p_2}	−3..+2	√	√	√	–	–	–	√	√	√	–	–	√	1,2,5	–	√	–	–
M_{p_3}	−3..+2	√	√	√	–	√	–	√	√	√	√	–	–	2,3,5	√	√	√	–
M_{p_4}	−3..+2	√	–	√	–	√	–	–	√	–	√	√	√	1,3,4,5	√	√	–	–
M_{p_5}	−2..+3	√	√	–	–	–	–	–	√	–	√	√	√	1	√	–	√	–
M_{r_1}	−1..+2	√	–	–	–	–	–	√	√	–	–	√	–	3,4,5	√	–	–	–
M_{r_2}	−2..+2	√	√	–	–	–	–	–	–	–	–	√	√	2,3,4,5	√	–	√	–
M_{r_3}	−1..+2	√	√	√	–	√	–	–	√	–	√	√	–	1,5	√	–	√	–
M_{r_4}	−3..+2	√	–	√	–	√	–	–	√	–	√	√	√	1,3,4,5	√	√	–	–
M_{r_5}	−3..+2	√	√	–	–	–	–	√	√	–	√	√	√	2,5	√	–	√	–
C_{p_1}	0	√	√	–	–	√	–	√	√	√	√	√	√	1,2	√	√	–	–
C_{p_2}	−2..+3	√	–	√	–	√	–	–	–	–	–	√	–	3,5	√	–	–	–
C_{p_3}	−2..+3	–	√	–	–	–	–	√	√	√	√	√	√	2	√	–	√	–
C_{p_4}	−3..+3	√	√	√	–	√	–	√	√	√	√	√	√	2,3,4	√	√	–	–
C_{p_5}	0..+1	√	–	–	–	–	–	√	√	√	–	√	–	1,4	√	–	√	–
C_{r_1}	−2..0	–	√	–	–	√	–	–	√	–	√	√	√	2,5	√	–	√	–
C_{r_2}	−2..0	√	√	√	–	–	–	–	√	–	–	√	–	3,4,5	√	–	√	–
C_{r_3}	−3..+3	√	√	√	–	√	–	√	–	–	√	√	√	4,5	√	√	–	–
C_{r_4}	−2..+3	–	√	–	–	–	–	√	√	√	√	√	√	2	√	–	√	–
C_{r_5}	−2..+3	√	–	√	–	√	–	–	–	–	–	√	–	3,5	√	–	–	–
S_{p_1}	−1..+1	√	√	–	–	√	√	–	–	√	√	–	√	2	√	√	√	–
S_{p_2}	−1..+1	√	√	-	–	√	√	–	–	√	√	–	√	2,3	√	√	√	–
S_{p_3}	−2..+1	√	√	–	–	√	√	–	–	√	√	–	–	3,5	√	√	√	–
S_{p_4}	−2..+2	√	√	–	–	√	–	–	√	√	√	–	√	3,4	√	√	√	–
S_{p_5}	−2..+1	√	–	√	–	–	√	–	–	√	–	√	–	1,2,3,5	√	√	√	–
S_{r_1}	−1..+1	–	√	–	–	–	√	–	–	–	√	√	√	1,5	√	√	–	–
S_{r_2}	−1..+1	√	√	–	–	√	√	√	–	√	√	√	√	1,3	√	√	√	–
S_{r_3}	−1..+1	√	√	–	–	–	√	–	–	√	√	√	√	1,3,4	√	√	√	–
S_{r_4}	−1..+2	√	–	√	–	√	–	√	√	√	√	–	√	1	√	√	–	–
S_{r_5}	−1..+2	√	–	–	–	√	√	–	–	√	√	√	√	2,5	√	√	–	–

Table 12
Optimized feature sets for aspect term extraction task in laptop domain.

Models	Features																	
	Word & context	PoS	Head Word	Chunk	Lemma	Stop Word	Word Length	Prefix	Suffix	Frequent aspect term	Dependency Relation	WordNet	Named Entity	Char n-grams (1,2,3,4,5)gram	Word cluster	Aspect term list	Sentiment Orientation	Orthographic c: IsCap, d: InitDigit
M_{f_1}	−2..+3	√	√	–	–	–	–	√	√	–	√	√	–	2,4	√	–	√	–
M_{f_2}	0..+1	√	–	–	–	√	–	–	–	–	–	√	–	1,4,5	√	–	√	–
M_{f_3}	−1..+1	√	√	–	–	–	–	√	√	–	–	–	–	2,3,4	√	–	–	–
M_{f_4}	−1..+1	√	–	–	√	√	–	√	√	–	√	–	–	4,5	√	–	√	–
M_{f_5}	−1..+1	√	–	√	–	–	–	–	–	–	√	–	–	3,4	√	–	√	–
C_{f_1}	−2..0	√	–	√	√	√	–	–	√	√	–	√	√	2,3,5	√	√	√	d
C_{f_2}	−1..+1	√	–	√	√	√	–	√	√	√	√	√	√	1,4	√	–	–	c
C_{f_3}	−1..+1	√	√	√	–	–	–	–	–	–	√	–	√	1,3,5	√	√	–	√
C_{f_4}	−2..+2	√	–	√	√	–	–	√	√	–	–	√	√	2,3	√	√	√	√
C_{f_5}	0..+2	√	√	√	√	√	–	√	√	√	–	√	–	4,5	√	√	√	d
S_{f_1}	−1..+1	√	√	√	–	√	√	√	–	–	–	√	–	4	√	–	–	c
S_{f_2}	−2..+1	√	√	–	√	√	–	–	–	–	–	√	√	1,4,5	√	–	–	d
S_{f_3}	−2..+1	√	–	–	–	–	–	√	√	–	–	√	–	4,5	√	–	–	c
S_{f_4}	−1..+1	–	–	√	–	√	–	–	√	√	–	√	–	2,3,4	√	√	–	–
S_{f_5}	−1..+2	√	√	–	–	√	√	–	√	√	–	√	–	–	√	–	–	d
M_{p_1}	−2..+3	√	√	–	–	–	–	√	√	–	√	√	–	2,4	√	–	√	–
M_{p_2}	−1..+1	√	√	–	–	–	–	√	√	–	–	–	–	2,3,4	√	–	–	–
M_{p_3}	−3..+2	√	√	–	–	–	–	√	√	–	√	–	–	1,3,5	√	–	√	–
M_{p_4}	−3..+2	√	√	–	√	–	–	–	–	–	√	√	–	1	√	–	√	√
M_{p_5}	−3..+2	√	√	–	–	–	–	–	–	–	√	√	–	1,2,4	√	–	√	–
M_{r_1}	−3..+3	√	√	√	–	–	–	√	–	–	√	√	–	2,3	√	–	√	–
M_{r_2}	−2..+2	√	√	–	–	–	–	–	–	–	–	√	√	2,3,4,5	√	–	√	–
M_{r_3}	−2..+3	√	√	√	–	√	–	–	√	–	√	√	√	1,3,4	√	√	√	–
M_{r_4}	−3..+2	√	–	√	–	√	–	–	√	–	√	√	√	1,3,4,5	√	√	–	–
M_{r_5}	−3..+2	√	√	√	–	–	–	√	√	√	–	–	√	1,2,5	–	√	–	–
C_{p_1}	0..+2	√	√	√	√	√	–	√	√	√	–	√	–	4,5	√	√	√	d
C_{p_2}	−1..+1	√	–	√	√	√	–	√	√	√	√	√	√	1,4	√	–	–	c
C_{p_3}	−3..+3	–	√	√	√	√	–	–	–	√	√	–	–	5	√	√	–	–
C_{p_4}	−2..+2	√	–	√	√	–	–	√	√	–	–	√	√	2,3	√	√	√	√
C_{p_5}	−2..0	√	–	–	√	–	–	–	–	√	√	√	√	1,3	√	√	–	–
C_{r_1}	−1..+2	√	√	√	–	√	–	√	√	√	–	–	√	3,4	√	√	√	–
C_{r_2}	−1..+1	√	√	√	–	–	–	–	–	–	√	–	√	1,3,5	√	√	–	√
C_{r_3}	0..+2	√	√	√	√	√	–	√	√	√	–	√	–	4,5	√	√	√	d
C_{r_4}	−2..+2	√	–	√	√	–	–	√	√	–	–	√	√	2,3	√	√	√	√
C_{r_5}	0	√	√	–	–	√	–	–	–	√	√	√	√	1,3	√	–	–	–
S_{p_1}	−2..+1	√	–	–	–	–	–	√	√	–	–	√	–	4,5	√	–	–	c
S_{p_2}	−2..+1	–	√	–	–	–	–	–	√	–	–	√	–	1,3,4	√	–	√	√
S_{p_3}	−2..+2	√	–	–	–	–	–	–	–	–	√	–	√	1	√	–	√	√
S_{p_4}	−2..+1	√	–	–	–	√	–	√	–	–	√	√	√	2,3,4	√	–	–	d
S_{p_5}	−2..+1	√	–	–	√	√	√	–	–	–	–	√	√	√	√	–	–	√
S_{r_1}	−1..+1	√	√	√	–	√	√	√	–	–	–	√	–	4	√	–	–	c
S_{r_2}	−2..+1	√	–	–	√	√	√	–	√	–	√	–	√	1,3,4	√	–	–	√
S_{r_3}	−2..+1	√	√	–	√	√	–	–	–	–	–	√	√	1,4,5	√	–	–	d
S_{r_4}	−1..+1	–	–	√	–	√	–	–	√	√	–	√	–	2,3,4	√	√	–	–
S_{r_5}	−1..+1	√	√	–	√	–	–	–	√	–	√	√	√	1,2,3,4	√	–	√	d

Table 13
PSO based optimized feature sets for sentiment classification.

Models	Features													
	Restaurant							Laptop						
	Word & context	Word Bigram	MPQA	Bing	Bing Direct	SentiWordNet	PMI	Word & context	Word Bigram	MPQA	Bing	Bing Direct	SentiWordNet	PMI
M_{f_1}	−2..+3	−1..+2	–	–	–	√	√	−2..+3	–	–	–	√	√	–
M_{f_2}	−2..+1	0..+2	–	√	–	√	√	0..+3	–	–	–	√	–	–
M_{f_3}	−2..+1	−2..+1	–	–	–	–	√	0..+3	0..+1	–	–	√	√	–
M_{f_4}	−2..+2	−4..+1	–	√	–	√	√	−4..+3	–	–	√	–	√	–
M_{f_5}	−2..+5	–	√	–	√	–	–	−2..+1	0..+2	–	–	√	√	–
C_{f_1}	−1..+1	0..+2	√	–	√	–	–	0	−2..+1	–	√	–	√	–
C_{f_2}	−1..+1	−3..+2	√	√	√	√	–	0	−2..0	–	√	–	–	–
C_{f_3}	−4...0	0..+2	–	–	√	√	√	0	−1..+1	–	√	–	√	–
C_{f_4}	0..+1	−4..0	√	–	√	√	–	−1..0	−2..+2	–	√	√	√	–
C_{f_5}	−2..0	0..+2	√	–	√	√	–	0	−1..+1	–	√	√	√	–
S_{f_1}	0..+3	−1..0	–	√	√	–	–	−1..+1	−1..+2	√	√	√	√	–
S_{f_2}	0..+1	0..+3	–	√	√	–	–	−1..+1	–	–	√	√	√	–
S_{f_3}	0..+2	–	–	√	√	–	–	0..+2	−1..0	√	√	√	–	–
S_{f_4}	0	−4..0	√	√	√	–	–	−1..+2	–	√	√	√	–	–
S_{f_5}	0..+2	–	–	√	√	–	–	−1..+1	−1..0	√	√	√	–	–

Table 14
Results of PSO based classifier ensemble w.r.t majority ($En_{Majority}$) & weighted ($En_{Weighted}$) techniques.

Objectives	Method	Restaurant				Laptop			
		Aspect term			Sentiment	Aspect term			Sentiment
		P	R	F	Acc	P	R	F	Acc
Accuracy	$En_{Majority}$	–	–	–	79.98	–	–	–	74.00
	$En_{Weighted}$	–	–	–	80.07	–	–	–	75.22
F–measure	$En_{Majority}$	86.58	81.39	83.90	–	83.98	65.75	73.75	–
	$En_{Weighted}$	86.27	82.01	84.09	–	84.70	66.05	74.22	–
Precision &	$En_{Majority}$	87.07	82.01	84.46	–	84.99	66.67	74.72	–
Recall	$En_{Weighted}$	87.09	82.10	84.52	–	85.49	66.7	74.93	–
Proposed Model		87.09	82.10	84.52	80.07	85.49	66.7	74.93	75.22

Table 15
Classifiers selected through PSO based classifier ensemble.

Domain	Task	Metrics	Method	Candidate Models
Restaurant	Aspect term	F-measure	$En_{Majority}$	C_{f_5}, C_{f_7}, $C_{f_{11}}$, $C_{f_{15}}$, $C_{f_{18}}$, S_{f_4} & S_{f_9}
			$En_{Weighted}$	$M_{f_{16}}$, $C_{f_{20}}$, C_{f_2}, C_{f_3}, C_{f_4}, C_{f_5}, $C_{f_{10}}$, $C_{f_{13}}$, $C_{f_{14}}$, $C_{f_{15}}$, $C_{f_{18}}$, $C_{f_{19}}$, $C_{f_{20}}$, S_{f_1}, S_{f_3}, S_{f_4}, $S_{f_{11}}$, $S_{f_{12}}$, $S_{f_{17}}$ & $S_{f_{19}}$
		Pre & Rec	$En_{Majority}$	C_{p_3}, C_{p_5}, C_{r_2}, C_{r_3}, C_{p_5}, C_{p_3}, S_{p_1} & S_{r_2}
			$En_{Weighted}$	C_{p_3}, C_{p_4}, C_{p_5}, C_{r_1}, C_{r_3}, S_{p_2}, S_{r_1} & S_{r_5}
	Sentiment	Accuracy	$En_{Majority}$	$M_{f_{11}}$, C_{f_1}, C_{f_2}, C_{f_4}, C_{f_5}, C_{f_8}, $C_{f_{11}}$, $C_{f_{12}}$, $C_{f_{15}}$, $C_{f_{19}}$, S_{f_3}, S_{f_5}, S_{f_6}, S_{f_7}, S_{f_8}, $S_{f_{14}}$, $S_{f_{18}}$, $S_{f_{19}}$ & $S_{f_{20}}$
			$En_{Weighted}$	M_{f_8}, $M_{f_{13}}$, $M_{f_{14}}$, C_{f_1}, C_{f_4}, C_{f_5}, C_{f_6}, $C_{f_{15}}$, $C_{f_{16}}$, $C_{f_{18}}$, $C_{f_{19}}$, S_{f_1}, S_{f_8}, S_{f_9}, $S_{f_{10}}$, $S_{f_{13}}$, $S_{f_{14}}$, $S_{f_{18}}$ & $S_{f_{20}}$
Laptop	Aspect term	F-measure	$En_{Majority}$	$M_{f_{14}}$, C_{f_3}, C_{f_9}, $C_{f_{10}}$, $C_{f_{13}}$, $C_{f_{15}}$, $C_{f_{16}}$, $C_{f_{17}}$, $C_{f_{18}}$, $C_{f_{19}}$, $C_{f_{20}}$, S_{f_4}, S_{f_5}, S_{f_7}, $S_{f_{10}}$, $S_{f_{16}}$ & $S_{f_{17}}$
			$En_{Weighted}$	$M_{f_{10}}$, C_{f_5}, C_{f_7}, C_{f_9}, $C_{f_{12}}$, S_{f_1}, S_{f_6}, S_{f_9}, $S_{f_{10}}$ & $S_{f_{18}}$
		Pre & Rec	$En_{Majority}$	C_{p_4}, C_{r_1}, C_{r_2}, S_{p_1}, S_{p_5}, S_{r_1} & S_{r_5}
			$En_{Weighted}$	M_{p_1}, C_{p_3}, C_{r_2}, C_{r_4}, S_{p_1}, S_{p_2}, S_{r_1} & S_{r_5}
	Sentiment	Accuracy	$En_{Majority}$	M_{f_6}, $M_{f_{10}}$, $M_{f_{15}}$, C_{f_1}, C_{f_3}, C_{f_5}, C_{f_6}, C_{f_7}, $C_{f_{12}}$, $C_{f_{15}}$, $C_{f_{19}}$, S_{f_4}, S_{f_8}, $S_{f_{14}}$ & $S_{f_{17}}$
			$En_{Weighted}$	M_{f_1}, M_{f_8}, $M_{f_{14}}$, $M_{f_{15}}$, C_{f_3}, C_{f_4}, C_{f_6}, C_{f_7}, C_{f_8}, $C_{f_{17}}$, S_{f_1}, $S_{f_{15}}$ & $S_{f_{16}}$

Table 16
Comparisons with existing systems. DLIREC [49], IHS_RD [16] and DCU [17] are the best performing systems at SemEval-2014.

Method	Restaurant		Laptop	
	Aspect term	Sentiment	Aspect term	Sentiment
Baseline	80.90	77.57	72.13	69.57
System [33]	81.91	78.48	72.42	71.25
DLIREC [49]	84.01	–	–	–
IHS_RD [16]	–	–	74.55	–
DCU [17]	–	80.95	–	70.49
System [18]	82.06	–	75.00	–
System [20]	–	76.46	–	68.50
Proposed model	84.52	80.07	74.93	75.22

classifier ensemble, would be $O((P*I*F_v*\xi)+(P*I*F_v)) \approx O(P*I*F_v*(\xi+1))$.

The key characteristics of the current work are as follows: (i). proposal of a two-step, first step of which performs feature selection and the second step performs ensemble learning for aspect based sentiment analysis; (ii). use of extensive feature sets for aspect term extraction and sentiment classification; (iii). proposal of an aspect term extractor and sentiment analyzer that yield state-of-the-art performance on benchmark datasets; (iv). finding that small set of relevant features can actually improve the classifier's performance; and (v). determining proper subsets of classifiers further improves the performance.

5.6. Comparison among feature selection techniques: PSO, PCA and Information gain

As a comparison to PSO based feature selection, we exploits PCA and Information gain for reducing the dimension of feature space for our problems. Principal Component Analysis (PCA) [51] is a useful statistical technique to compress the data by reducing the number of dimensions. It finds the patterns in data of high dimension and transforms it into lower dimension by leaving out redundant information. PCA starts its processing by calculating the eigen values and eigen vectors of the covariance matrix. These vectors provide information about the patterns in the data. Eigen vector corresponds to the highest eigen value contains the most important pattern, where as vector corresponds to next highest eigen value contains relatively lesser information than the first but more than others. Thus, by keeping only top k vectors, we can ignore/remove $n-k$ redundant features from the datasets. We then train, test and evaluate a model using the reduced subset of feature sets. We perform the following steps for PCA based dimensionality reduction:

1. Convert training and test datasets into numerical form.
2. Find covariance matrix of datasets.
3. Calculate eigen values and Eileen vector of the covariance matrix.
4. Sort eigen vectors w.r.t non-increasing eigen values.
5. Keep the top k vectors.
6. Train, test and evaluate the reduced datasets.

Information gain corresponds to the gain obtained due to the reduction in entropy when the data is distributed among the different classes with respect to a particular feature. We calculate information gain[10] for each feature and sort them in ascending order. Top few features are then selected and used for the training of classifier. We report the result of PCA and Information gain based feature space reduction techniques in Table 17 along with the result of PSO. It also reports the value of f_n/k (i.e. number of features) that were required for the respective models. Results show that PSO based methods achieve better result as compared to both PCA and Information gain for all the cases. Also, it should be noted that PSO based model requires relatively less number of features than the other two techniques.

[10] We used WEKA 3.6 for implementation.

Table 17
Comparison between PSO, PCA and Information gain.

Classifiers	Method	Restaurant						Laptop					
		Aspect term				Sentiment		Aspect term				Sentiment	
		P	R	F	f_n/k	Acc	f_n/k	P	R	F	f_n/k	Acc	f_n/k
ME	PSO	71.44	74.33	**72.86**	38	**74.95**	20	61.43	57.49	**59.39**	41	**66.81**	13
	PCA	69.97	72.75	71.33	48	68.78	24	56.63	51.52	53.96	50	63.60	18
	InfoGain	–	–	–	–	72.66	25	–	–	–	–	62.99	17
CRF	PSO	85.39	80.95	**83.11**	35	**78.65**	16	83.39	64.52	**72.75**	44	**72.17**	11
	PCA	84.69	78.57	81.51	49	67.27	25	80.71	62.07	70.18	52	65.96	17
	InfoGain	–	–	–	–	75.22	25	–	–	–	–	67.58	17
SVM	PSO	83.53	80.07	**81.76**	29	**77.24**	11	81.99	65.44	**72.78**	27	**66.97**	11
	PCA	75.37	76.10	75.73	57	74.51	25	68.09	58.40	62.88	64	64.22	17
	InfoGain	–	–	–	–	74.86	25	–	–	–	–	62.84	17

Table 18
Comparisons with existing ensemble techniques.

Method	Restaurant				Laptop			
	Aspect term			Sentiment	Aspect term			Sentiment
	P	R	F	Acc	P	R	F	Acc
Proposed method	85.39	80.95	**84.52**	**80.07**	81.99	65.44	**74.93**	**75.22**
Bagging	67.54	63.93	65.69	73.28	41.28	44.04	42.62	62.84
AdaBoost	67.54	63.93	65.69	70.01	41.89	43.76	42.81	59.02
Stacking	64.55	62.40	63.45	73.98	39.44	42.78	41.05	63.60
Voting	66.04	62.36	64.15	71.64	44.34	43.02	43.67	62.07

5.7. Comparison among ensemble techniques: Bagging, Boosting, Stacking and Voting

We also perform experiments with some of the well-known ensemble techniques such as bagging [52], AdaBoost [53], stacking [54] and voting [55]. We choose sequential minimal optimization (SMO) algorithm for support vector machine as a base classifier for bagging and AdaBoost. For stacking and voting we use logistics regression, SMO and naive Bayes as our base classifiers. In addition, we opt for SMO as our meta classifier in stacking and majority class technique for voting. For implementation of these algorithms we used WEKA 3.6[11].

We observe that, in all cases, our proposed approach yields better results compared to these existing methods. Results are reported in Table 18. Performance improvements were also shown to be statistically significant.

5.8. Error analysis

In this section we present an analysis of the errors encountered by our proposed system. We present both quantitative as well as qualitative analysis with respect to the across-domain and across-classifier phenomenon of feature selection.

5.8.1. Quantitative analysis

Error analysis is performed in terms of confusion matrix as shown in Fig. 3. For aspect term extraction in restaurant domain, quite a good number of 'B-ASP' (Beginning of an aspect term) and 'I-ASP' (Intermediate tokens of an aspect term) are wrongly predicted as 'O' (others). A total of 347 instances were misclassified for these two classes. Our system also faces the same problem for the laptop domain as there are 400 misclassified instances for 'B-ASP' and 'I-ASP' classes. Possible reason behind this anomaly could be the presence of relatively large number of 'O' (other than aspect term) tokens in the training set. Also, few instances of 'I-ASP' and 'O' were misclassified as 'B-ASP' which further hampers the system's performance. In sentiment classification task, majority of the 'positive' instances were correctly predicted for both the domains. However, the system misclassifies most of the time for the 'neutral' instances. For the restaurant domain it misclassifies 116 instances, while only 80 instances are classified correctly. Similarly in laptop domain, proposed system performs slightly better than the restaurant domain for the 'neutral' class with 100 correct classifications and 69 misclassifications. For 'conflict' class our system fails to correctly predict all the instances. This could be because the proposed model was trained with a very fewer number of instances of 'conflict' class. We hope the system could do better with a greater number of training instances.

5.8.2. Qualitative analysis

We analyze the outputs of our proposed system and found that it mainly lags behind in the following scenarios:

Aspect Term Extraction:

- System fails in correctly identifying an aspect term which include braces. For example, our system predicted *installation disk* as an aspect term but fails to tag the braces (including inner text i.e. 'DVD') as 'I-ASP'. This results in incorrect identification of the term in question.

Review:	No	*installation*	*disk*	*(*	*DVD*	*)*	is	included	.
Actual:	O	B-ASP	I-ASP	I-ASP	I-ASP	I-ASP	O	O	O
Predicted:	O	B-ASP	I-ASP	O	O	O	O	O	O

- Aspect terms which are made up of words and digits are shown to have been misclassified. For example, aspect terms like '*Windows 7*' and '*i5*' are not predicted correctly by our proposed system. In the first case our system predicts only '*Windows*' as aspect leaving '*7*' alone. However, in the second case it does not predict it at all.
- We found few cases where the last word of any sentence, even being a part of an aspect term, is misclassified. This may be attributed to the fact that due to the lack of right contextual information, the system has not been able to identify these properly.

[11] http://www.cs.waikato.ac.nz/ml/weka/.

	B-ASP	I-ASP	O
B-ASP	941	9	180
I-ASP	56	334	167
O	55	21	11658

(a) Restaurant Aspect Term Extraction

	B-ASP	I-ASP	O
B-ASP	429	12	206
I-ASP	28	203	202
O	37	19	11467

(b) Laptop Aspect Term Extraction

	positive	negative	neutral	conflict
positive	700	14	13	1
negative	59	128	9	0
neutral	96	20	80	0
conflict	8	5	1	0

(c) Restaurant Sentiment Classification

	positive	negative	neutral	conflict
positive	295	24	22	0
negative	16	92	20	0
neutral	39	30	100	0
conflict	11	4	1	0

(d) Laptop Sentiment Classification

Fig. 3. Confusion matrix for different problems.

Review:	Apple	is	unmatched	in	...	and	*customer*	*service*	.
Actual:	O	O	O	O	...	O	B-ASP	I-ASP	O
Predicted:	O	O	O	O	...	O	B-ASP	O	O

Sentiment Classification:

- Negative words close to the aspect term bias the sentiment towards it. For the example given below, *installation disk (DVD)* was an aspect term and the actual polarity towards it was *neutral*, but the presence of *No* changes the polarity of installation disk (DVD) to 'negative'.
  Review: No *installation disk (DVD)* is included.
  Aspect Term: *installation disk (DVD)*
  Actual sentiment: neutral **Predicted sentiment:** negative
- Our system fails to correctly classify sentiment in smaller sentences. An example is given below:
  Review: The *sangria*'s - watered down.
  Aspect Term: *sangria*
  Actual sentiment: negative **Predicted sentiment:** positive

Few of the features that we have implemented in our proposed model have a higher degree of relevance against others, irrespective of the classification algorithms we used. For example, in aspect term extraction task 'context information', 'word cluster', 'PoS tag' and 'WordNet' are the most prominent features in most of the models (for at least 11 out of 15 best performing models). Here, we closely analyze these features and present the possible explanations behind their selection by majority of the models. Sometimes the combination of features vary because of the random behavior of PSO. In PSO, each bit position of the feature vector randomly oscillates between 0 and 1 depending upon the random number generated.

Since the task of aspect term extraction can be seen as the sequence labeling problem, and therefore word and its context information are unarguably required, thus justifying its selection in the model construction. Word cluster feature tries to group different words that are semantically similar in nature. Therefore, it helps in grouping the tokens that have similar characteristics, for example, all the nouns and noun groups have the tendency of being in the same cluster(s). We induce 21 and 20 clusters for the restaurant and laptop domains, respectively. We observe that all the aspect terms in the test dataset lie in the same training clusters for each domain.

Part of Speech (PoS) tag feature is selected in most of the cases. Aspect terms generally refer to the entities that belong to the noun and noun groups. The use of PoS tag as a feature encourages the model to better capture the properties of aspect terms. There are approximately 84% and 82% noun aspect terms present in the training and gold dataset for restaurant domain, respectively. Similarly in laptop domain, training and gold datasets contain approximately 80% and 74% aspect terms that denote the noun PoS tag. Therefore, it is evident that PoS information of a token does play an important role in correctly identifying the aspect terms. Majority of the aspect terms belong to the noun PoS categories. For the restaurant and laptop domains, these are approximately 89% and 82%. Ou of these 81% and 72% have been correctly identified. The WordNet related feature has been very useful for handling the unseen aspect terms. There are 353 (31%) and 273 (41.74%) instances of unseen aspect terms in the restaurant and laptop domain, respectively. For example, an unseen aspect term 'hotdogs' is correctly predicted by our model whose synset element 'food' was present in the training dataset.

In sentiment classification task, three lexicons features i.e. Bing, Bing Direct and SentiwordNet along with the word and context features are chosen most of the time as candidates for training

and testing in both the domains. However, Bing and Bing direct features have higher impact in laptop domain as compared to the other lexicon features. This could be because Bing Liu opinion mining lexicon [41] itself was created from the electronics corpus.

Now, we analyze the results of feature selection across the various classifiers. Closer observations to the various experimental setups suggest that selection of features heavily depends on the classification algorithm and the PSO. Due to the nature of randomness of PSO, the feature vectors do change over the iterations, and this has significant effect on the decision behind a features' inclusion or exclusion from the final feature set. Feature vectors change depending upon the velocity, which is mainly controlled by three important parameters of PSO, *viz.* inertia weight, social scaling and cognitive parameters. Hence the performance could vary from one iteration to the other iteration. Again due to randomness of the search process, selection of features could vary even within the different models, built using the same classification algorithm. Our experiments show that among the top-performing models there are features which are not always selected in all the models of CRF, SVM or ME.

We also try to analyze the classifier's behavior by adding or deleting a feature to the optimized feature set in model training. We choose 'chunk' feature for this analysis. We add 'chunk' feature to the optimized feature sets of the top models (i.e. M_{f_1}, C_{f_1} and S_{f_1}) if it was not selected by PSO.

Restaurant domain: From Table 11 it can be observed that 'chunk' feature was selected for the ME model, M_{f_1}, but not for the models C_{f_1} and S_{f_1}. So, we drop 'chunk' feature from the first model and add it to the last two models. The model based on ME (i.e. M_{f_1}) reports an F-measure of 72.64% (without chunk) as compared to 72.86% (with chunk). Similarly, C_{f_1} and S_{f_1} with chunk information yield 82.77% and 81.38% F-measure, respectively as compared to 83.11% and 81.76% (i.e. without chunk). Hence the adding (deleting) any feature to (or from) the feature set actually hurts the systems' performance. This entails that our feature set as determined by PSO is optimized in nature. It is also be noted that although this feature is not selected in the best models of SVM or CRF, this is included in the other top models. The reason behind this is the random behavior of PSO. As discussed earlier, chunk information is useful mainly for detecting the multiword aspect terms. However, the restaurant dataset contains relatively lesser percentage (24%) of multiword aspect terms, which could be a possible justification for the absence of chunk information from the optimized feature sets of few models.

Laptop domain: Similarly, we perform the above analysis for the laptop domain. Chunk information was absent from M_{f_1} but present in both C_{f_1} and S_{f_1} models. This suggest that 'chunk' feature has a relatively higher degree of relevance in laptop dataset which has approximately 45% multiwords aspect terms. The other cause is again the random behavior of PSO.

Closer analysis to the various experimental setups suggests that selection of features heavily depends on two important aspects, viz. classification algorithm used and the PSO. Due to randomness of PSO, particles do change over the iterations, and this has significant effect on the decision behind a feature's/classifier's inclusion or exclusion into/from the final optimized set. Any change in the particle depends upon the velocity, which is mainly controlled by three important parameters of PSO, *viz.* inertia weight, social scaling and cognitive parameters. Hence, the performance could vary from one iteration to the other. We found that PSO indeed produces the optimized feature sets, and any perturbation to these led to lower performance. Our experiments show that the use of ensemble techniques in our proposed method did improve the result by a good margin, which is in line with [28] who proved the effectiveness of ensemble learning is sentiment analysis.

6. Conclusion

In this paper we have presented an efficient method for feature selection and ensemble learning for aspect based sentiment analysis. The algorithm is based on single objective PSO. As base learning algorithms we use CRF, SVM and ME. In the first step we determine the best feature sets for aspect term extraction and sentiment classification. This yields a set of solutions, each of which represents a particular feature combination. Based on certain criteria we choose the most promising solutions from the final population of PSO. The models developed with these feature combinations are combined together using a PSO based ensemble technique. The ensemble learner finds out the most eligible models, that when combined together, maximizes some classification quality measures like F-measure (for aspect term extraction) or accuracy (for sentiment classification). As the base learning algorithms we use three classifiers, namely ME, CRF and SVM. We have identified and implemented various lexical, syntactic or semantic level features for solving the problems. Experiments on the benchmark datasets of SemEval-2014 show our proposed techniques attain state-of-the-art performance for both aspect term extraction and sentiment classification. We compare the performance with the best performing systems that were developed using the same setups, several baseline models and the existing systems. In all the settings our proposed methods showed the effectiveness with reasonable performance increments. The key contributions of the current work can be summarized as below: (i). proposal of a two-step process for feature selection and ensemble learning using PSO; (ii). developing a PSO based feature selection and ensemble learning technique for the application like sentiment analysis; (iii). building domain-independent models for aspect based sentiment analysis that achieves state-of-the-art performance; (iv). finding how efficiently we can improve the classifiers' performance if it is trained with the most relevant set of features (particularly for sentiment analysis).

The current work focuses on single objective optimization technique, where we deal with only one objective function at a time. In future we would like to explore how multiobjective optimization that deals with the optimization of more than one objective function be effective for solving the problems. Future work also includes the studies of how the proposed system works for sentiment analysis in other domains and languages.

References

[1] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retrieval 2 (1-2) (2008) 1–135.
[2] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the 10th KDD, Seattle, WAs, 2004, pp. 168--177.
[3] B. Liu, Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2012.
[4] P.D. Turney, Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews, in: Proceedings of the 40th ACL, 2002, pp. 417–424.
[5] S.-M. Kim, E. Hovy, Determining the sentiment of opinions, in: Proceedings of the 20th international conference on Computational Linguistics, Association for Computational Linguistics, 2004, p. 1367.
[6] V. Jagtap, K. Pawar, Analysis of different approaches to sentence-level sentiment classification, Int. J. Scient. Eng. Technol. (ISSN: 2277-1581) 2 (2013) 164–170.
[7] L. Zhuang, F. Jing, X.-Y. Zhu, Movie review mining and summarization, in: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, in: CIKM '06, 2006.
[8] A. Mukherjee, B. Liu, Aspect extraction through semi-supervised modeling, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, in: ACL '12, 2012, pp. 339–348.
[9] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 2014.

[10] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 486–495.
[11] S. Brody, N. Elhadad, An unsupervised aspect-sentiment model for online reviews, in: Proceedings of NAACL, Los Angeles, CA, 2010, pp. 804--812.
[12] M.D. Munezero, C.S. Montero, E. Sutinen, J. Pajunen, Are they different? affect, feeling, emotion, sentiment, and opinion detection in text, IEEE Trans. Affective Comput. 5 (2) (2014) 101–111, doi:10.1109/TAFFC.2014.2317187.
[13] A.-M. Popescu, O. Etzionir, Extracting product features and opinions from reviews, in: Proceedings of the Conference on HLT/EMNLP, 2005, pp. 339–346.
[14] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G.A. Reis, J. Reynar, Building a sentiment summarizer for local service reviews, WWW Workshop on NLP in the Information Explosion Era, 14, 2008.
[15] A. Fahrni, M. Klenner, Old wine or warm beer: target-specic sentiment analysis of adjectives, in: Symsposium on Affective Language in Human and Machine, The Society for the Study of Artificial Intelligence and Simulation of Behavior (AISB), 2008, pp. 60–63.
[16] M. Chernyshevich, IHS R&D belarus: cross-domain extraction of product features using conditional random fields, 2014, pp. 309–313.
[17] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, L. Tounsi, Dcu: aspect-based polarity classification for semeval task 4, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 223–229.
[18] P. Liu, S. Joty, H. Meng, Fine-grained opinion mining with recurrent neural networks and word embeddings, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1433–1443.
[19] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowl.-Based Syst. 108 (2016) 42–49.
[20] R. Kaljahi, J. Foster, Detecting opinion polarities using kernel methods, in: Proceedings of the Workshop on Computational Modelling of People's Opinions, Personality, and Emotions in Social Media, Osaka, Japan, 2016, pp. 60--69.
[21] O. Kummer, J. Savoy, Feature selection in sentiment analysis., in: CORIA, 2012, pp. 273–284.
[22] P. Koncz, J. Paralic, An approach to feature selection for sentiment analysis, in: 2011 15th IEEE International Conference on Intelligent Engineering Systems, IEEE, 2011, pp. 357–362.
[23] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Syst. Appl. 35 (4) (2008) 1817–1824.
[24] R. Xia, C. Zong, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification, Inf. Sci. 181 (6) (2011) 1138–1152.
[25] A. Aue, M. Gamon, Customizing sentiment classifiers to new domains: A case study, in: Proceedings of recent advances in natural language processing (RANLP), 1, 2005. 2–1
[26] A. Hassan, A. Abbasi, D. Zeng, Twitter sentiment analysis: a bootstrap ensemble framework, in: Social Computing (SocialCom), 2013 International Conference on, IEEE, 2013, pp. 357–364.
[27] X. Wan, Using bilingual knowledge and ensemble techniques for unsupervised chinese sentiment analysis, in: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, 2008, pp. 553–561.
[28] G. Wang, J. Sun, J. Ma, K. Xu, J. Gu, Sentiment classification: the contribution of ensemble learning, Decis. Supp. Syst. 57 (2014) 77–93.
[29] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.
[30] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: ICML, 2001, pp. 282–289.
[31] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, doi:10.1023/A:1022627411411.
[32] K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.
[33] D.K. Gupta, K.S. Reddy, A. Ekbal, et al., Pso-asent: feature selection using particle swarm optimization for aspect based sentiment analysis, in: Natural Language Processing and Information Systems, Springer, 2015, pp. 220–233.
[34] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[35] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[36] Z. Toh, W. Wang, DLIREC: aspect term extraction and term polarity classification system, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, p. 235â;;240.
[37] G.A. Miller, WordNet: a lexical database for english, Commun. ACM 38 (11) (1995) 39–41.
[38] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, J.C. Lai, Class-based N-gram models of natural language, Computat. Ling. 18 (4) (1992) 467–479.
[39] V. Hatzivassiloglou, K.R. McKeown, Predicting the semantic orientation of adjectives, in: Proceedings of the ACL/EACL, 1997, pp. 174–181.
[40] J. Wiebe, R. Mihalcea, Word sense and subjectivity, in: Proceedings of the COLING/ACL, 2006, pp. 1065–1072.
[41] X. Ding, B. Liu, P.S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, in: WSDM '08, 2008.
[42] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), European Language Resources Association (ELRA), Valletta, Malta, 2010.
[43] J. Kennedy, J.F. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, 2001.
[44] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[45] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: IEEE World Congress on Computational Intelligence, IEEE, 1998, pp. 69–73.
[46] M.E.H. Pedersen, Tuning & simplifying heuristical optimization, University of Southampton, Ph.D. thesis, 2010.
[47] M.E.H. Pedersen, A.J. Chipperfield, Simplifying particle swarm optimization, Appl. Soft Comput. 10 (2) (2010) 618–628.
[48] M.E.H. Pedersen, Good parameters for particle swarm optimization, Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001 (2010).
[49] Z. Toh, W. Wang, Dlirec: aspect term extraction and term polarity classification system, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 235–240.
[50] T.W. Anderson, S. Scolve, Introduction to the Statistical Analysis of Data, Houghton Mifflin, 1978.
[51] I. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.
[52] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[53] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Thirteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 1996, pp. 148–156.
[54] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (1992) 241–259.
[55] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239, doi:10.1109/34.667881.