*Journal of **AI** and **D**ata **M**ining*
Vol 6, No 2, 2018, 341-353
DOI: 10.22044/JADM.2017.1024

MHSubLex: Using Metaheuristic Methods for Subjectivity Classification of Microblogs

H. Keshavarz and M. Saniee Abadeh*

Faculty of Electrical & Computer Engineering, Tarbiat Modares University, Tehran, Iran.

Received 06 May 2016; Revised 21 June 2016; Accepted 17 April 2017
**Corresponding author: saniee@modares.ac.ir (M. Saniee).*

Abstract

In Web 2.0, people are free to share their views, experiences, and opinions. One of the problems that arise in Web 2.0 is the sentiment analysis of texts produced by users in outlets such as Twitter. One of the main tasks of sentiment analysis is subjectivity classification. Our aim is to classify the subjectivity of tweets. To this end, we create subjectivity lexicons, in which the words are classified into the objective and subjective ones. To create these lexicons, we make use of three metaheuristic methods. Two meta-level features are extracted in this method, which show the number of subjective and objective words in tweets according to the lexicons. We then classify the records based upon these two features. By comparing accuracy and f-measure to the baselines, it is shown that the proposed method performs better. In the three metaheuristics, it is observed that the genetic algorithm performs better than the simulated annealing and the asexual reproduction optimization, and its performance is also better than all baselines in two of the three assessed datasets, in terms of accuracy. The lexicons that are created using this method can give an insight about the subjectivity and objectivity of words.

Keywords: *Evolutionary Computation, Genetic Algorithm, Natural Language Processing, Prediction Methods, Sentiment Analysis, Twitter, Web 2.0.*

1. Introduction

People tend to share their opinions about numerous subjects. Every day, they turn to the Web 2.0 outlets to talk about various entities such as products and people [1]. Web 2.0 and online social media have enabled people to express their ideas more easily than before. By the emergence of social media, thousands of people have started to share their views, and processing the text they generate can help us in the decision-making processes [2]. The analyzing and processing opinions people express is a discipline called sentiment analysis or opinion mining [1]. The text these users provide in social media can be processed and used in studying the human behavior [3].

One of the platforms of Web 2.0 is microblogging, in which people share their views in short text. One of the most popular microblogging platforms is Twitter.

The posts of users in Twitter are called tweets. They can have 140 characters at the most. There are more than 310 million monthly active users in Twitter[1]. To understand what people say about different subjects, the subjective tweets should be chosen, and the objective ones must be omitted. Hence, the tweets are divided into two groups: objective tweets that do not contain opinionated text and subjective tweets that contain the personal viewpoints of people towards objects and entities. The subjective posts are positive, negative or mixed. An example of a subjective tweet is “Why is #Siri always down @apple”. In this tweet, the writer has expressed a negative opinion about the Siri feature of apple iPhones. An objective tweet would be “@apple Or @Microsoft Buying Out @RIM ?”, which does

[1] https://about.twitter.com/company

not have a personal viewpoint. The labels of these records (negative, positive or objective) are included in the datasets, which will be discussed later.

According to Liu [1], an objective text provides factual information, while a subjective text has personal feelings, beliefs, and views. To get a better insight about what people say in social media, it is critical to be able to distinguish an opinionated text from an objective text. To determine whether a text is objective or subjective is called subjectivity classification [4]. Subjective sentences show the writer's opinions, evaluations, emotions, beliefs, judgments, etc. [5].

Having tools for subjectivity classification is vital because the volume of data is huge, and it is virtually impossible to manually label the millions of tweets that are posted online, whether for subjectivity or for polarity classification. Because of this, the sentiment analysis methods are used to classify different dimensions of sentiments in tweets, one of which is subjectivity classification [1]. The other dimensions include polarity, strength, and emotion detection. Another aspect of sentiment analysis that has recently become popular is figurative language. It has been one of the tasks of the SemEval competition [6], in which the participants were asked to provide methods for sentiment analysis, where irony, sarcasm, and metaphors are prevalent. Irony and sarcasm are important for our work because they often have a negative impact [6]. Irony detection has been thoroughly studied in [7]. Irony has been discussed as a tool for the users to express their thoughts in a creative manner [7].

An important tool for the sentiment analysis is a sentiment lexicon. Many methods use sentiment lexicons for different sentiment analysis tasks. A sentiment lexicon is a dictionary of words or phrases, which are called sentiment words or phrases. These sentiment phrases and words are either divided into positive and negative groups or numbers are assigned to them that show how much positive, negative or neutral they are or what their emotional values are.

In this work, we develop subjectivity lexicons, which show which word can be considered objective and which one is subjective. The subjectivity or objectivity of words is automatically inferred from the corpora. Then using these lexicons, we define two meta-level features for datasets, and with them, we classify the text into subjective and objective classes using the existing classifiers. These features are called the meta-level features because they are based upon another entity, i.e. lexicons, and we use this term in accordance of [3].

These subjectivity lexicons are created using three methods based on datasets: Genetic Algorithm (GA), Simulated Annealing (SA), and Asexual Reproduction Optimization (ARO), which will be discussed later. Each dataset is used to generate subjectivity lexicons. The assumption in lexicons is that certain words are subjective. In our method, the algorithm decides whether a word is subjective or objective. We also use lexicons generated by one dataset to classify text on the other datasets, and gain favorable results.

Our main contributions in this paper are as follow: (i) we build subjectivity lexicons, in which the words are grouped into the subjective and objective ones by means of three metaheuristic methods; and (ii) we present the subjectivity classification as an optimization problem, and then solve it using the lexicons that were created.

We also show that we have better results than baselines, and in some datasets, our method outperforms baselines by several accuracy and f-measure points. Our method also can capture subtleties in text. The GA section of our method is explained in [8]. This paper is an extended version of [8]. Here, two other metaheuristic methods are used additionally, the method is tested on another dataset, the results and the interpretations are expanded, and the method is explained in more detail.

Our results show that using this optimization problem leads to promising results in most of the assessed datasets. Since we generate lexicons, we compare our results with the results of baselines using the existing lexicons.

The rest of this paper is organized as follows: In Section 2, the works done in the field of subjectivity classification are reviewed, focusing on works on social media. We also introduce the existing lexicons in this section, and assess the usage of metaheuristics in sentiment analysis. In Section 3, we explain the proposed optimization problem, and provide ways to solve it using the three metaheuristics GA, SA, and ARO. In Section 4, the datasets are introduced, the experiments are explained, and the results are compared with the existing lexicons. The paper is concluded in Section 5, in which the future works are presented.

2. Related works

In this section, we study the previous research works on the subjectivity classification, and describe the sentiment lexicons that are used for the subjectivity and polarity classification.

2.1. Previous work on subjectivity classification

Tweets tend to be about products, services, and other entities. The posts in Twitter are short, and are usually right to the point. Hence, the tweets are a valuable resource for the sentiment analysis [3]. Most of the subjectivity classification approaches are based upon supervised learning [3]. In one of the earlier works, a naïve bayes classifier was used with binary features such as word presence of pronouns, adjectives, cardinal numbers, and certain modals and adverbs [9]. The authors have addressed evidentiality in text, and have talked about subjective and objective texts years before the formal introduction of opinion mining.

Pang and Lee [10] and Pang et al. [11] have intuitively used the n-gram features to classify a text into subjective and objective. Their work focused on the subjectivity detection of large text, as opposed to short text as in tweets. In [10], the authors have used a subjectivity detector, which finds out whether a sentence is subjective. Then the subjective text is fed into an opinion polarity classifier. Wiebe has used an unsupervised algorithm for classification of subjectivity [12]. The author used some seed words as subjective ones, and found other similar words and expanded the subjective set of words. Wiebe has used different seed synonyms.

Barbosa and Feng [13] have studied the problem of subjectivity and polarity detection of tweets. Their approach consisted of two steps; first, classification of subjectivity, and then classification of polarity. They used characteristics of writing styles of tweets, and meta-information of the words used in tweets. They also made use of prior subjectivity in polarity classification; they modeled the predicted tweets as weakly or strongly subjective.

Jiang et al. [14] have proposed a three-step algorithm. The first phase of their algorithm was subjectivity classification, and the other two steps were polarity classification and graph-based optimization. They explain why the target-independent sentiment analysis does not yield promising results on Twitter, and say that because people tend to talk about many targets in a tweet, it is important to do the sentiment analysis target-dependent. Pak and Paroubek [15] have also worked on Twitter. They argued that subjective texts are usually written in the first-person or address the second-person and tend to be in simple past tense, while an objective text is often in the third-person and uses past participle.

Wang et al. [16] have used two sets of features for their subjectivity classifier: content features such as unigrams, punctuations, and emoticons, and the sentiment lexicon features. They also used the same features for polarity classification of a text. Liu et al. [17] have used two language models for classification of Twitter data into subjective and objective classes: one model for subjective class and one for objective class. They computed the likelihood of a test tweet to be in either class, and then the classification was based upon it. They assumed that emoticons such as ":)" and ":("" make tweets subjective, and tweets that include objective URL links are objective. They argued that it was hard to have an assumption about objective tweets, though they cited some articles that had tried to make such assumptions.

Yu and Hatzivassiloglou [18] have used the intuition that subjective sentences were more similar to the other subjective sentences than to the objective ones. They measured the similarity of sentences, and used n-grams, parts of speech, and other features in a naïve bayes classifier to solve the problem. They also used the words that were semantically oriented, i.e. positive or negative, to differentiate between a subjective and objective text.

Bravo-Marquez et al. [3] have used the meta-level features from the existing lexicons for subjectivity and polarity classification of big social data. Their meta-level features included the sum of scores of positive words, sum of scores of negative words, and number of words that match the joy, trust, etc. word list from each lexicon, when applicable. They provided the baseline results for each lexicon that we will use here to compare our results with.

Koto and Adriani have analyzed the features for the subjectivity and polarity classification [19]. The features they studied included punctuation features, parts of speech, and emoticons. Mansour et al. [20] have created an ensemble classifier for the subjectivity and polarity classification, and they selected a compact set of features for this task.

2.2. Sentiment Lexicons

In this sub-section, we review the sentiment lexicons that are used for the subjectivity and polarity classification. Most of the sentiment lexicons are sets of words or phrases with scores assigned to them, showing their polarity. However, there are sentiment lexicons, in which the words are simply divided into groups without assigning a score.

AFINN: Bradley and Lang [21] have proposed a lexicon named Affective Norms for English

Words (ANEW). This lexicon belonged to the days before micro-blogging. Nielsen [22] has released AFINN lexicon inspired from ANEW, in which he included the slang words used in the micro-blogging platforms. In AFINN, the score of positive words is from +1 to +5, and the score of negative words is from -1 to -5. AFINN contains 2477 English words. This lexicon was built based on the psychological reaction of people to words.

Bing Liu's Lexicon: This lexicon has been experimentally constructed by Bing Liu. The words in this lexicon are divided into the positive and negative ones. The lexicon includes 2006 positive words and 4783 negative words. He has used it in his works, and it also includes slang and misspelled words.

EmoLex: This lexicon, also called NRC-emotion, includes words annotated with eight emotions that are according to the Plutchik wheel of emotions: joy, trust, sadness, anger, surprise, fear, anticipation, and disgust [23]. The words have been tagged using the Amazon Mechanical Turk crowdsourcing platform.

NRC-hashtag: The team NRC-Canada created this lexicon for the SemEval task [24]. This lexicon is automatically constructed based on 775310 tweets. The tweets have positive or negative hashtags such as #good, #bad, #excellent, and #terrible, and the words are grouped considering the hashtags. The sentiment score for each uni- or bigram ranges from -5 to +5, and is calculated using pointwise mutual information of words and labels of the tweets.

OpinionFinder: This lexicon has been presented in [25] by Wilson et al., and is based upon the Multi-Perspective Question-Answering dataset (MPQA). The entires are labeled as positive or negative. The grouping of words is done manually.

Sentiment140: This is another lexicon created by the NRC-Canada team. It is created like NRC-hashtag lexicon but here, emoticons were used for labeling the tweets as positive or negative, instead of hashtags. This lexicon was created using 1.6 million tweets that had positive and negative emoticons such as ":)" and ":(".

SentiWordNet: SentiWordNet 3.0 has been proposed by Baccianella et al. [26]. This lexicon is based upon WordNet, which groups the words together in synsets. In SentiWordNet 3.0, each word has three scores: positivity, objectivity, and negativity scores. These scores range from 0 to 1, and are calculated based upon the semi-supervised methods.

These lexicons with three methods form our baselines. The results from baselines are taken from [3]. In this paper, these three methods are used as baselines as well: Sentiment140, SentiStrength, and SenticNet. The Sentiment140 method is a web application for classification of tweets, and is based upon the work by Go et al. [27]. The SentiStrength method is focused on the classification of sentences [28]. This method returns two scores: a positive score ranging from 1 (not positive) to 5 (extremely positive), and a negative score ranging from -1 (not negative) to -5 (extremely negative).

SenticNet 2 [29] uses the Semantic Web techniques for the semantic-level analysis, and returns a polarity score and a sentic vector for sentences.

2.3. Metaheuristics in sentiment analysis

In this sub-section, we will address the use of metaheuristics in the sentiment analysis. One of the most relevant works is [30], in which a GA has been proposed for subjectivity detection. The authors use a big set of features, and select the most relevant features by means of a GA. Another important work is [31], in which a hybridized GA has been used for feature selection in opinion classification. Authors in [32] have created a hybrid of particle swarm optimization and support vector machines for classification of movie reviews. Another work is [33], in which tabu search has been incorporated.

An artificial immune system is used in [34], by which the words that should be used in the sentiment classification are chosen. Genetic programming has been incorporated in [35] to create new features based on the existing features. Finally, in [36], a GA has been used to find paradigm words in tweets.

3. The MHSL method

In this section, we describe our method, MHSL (MHSubLex), which is an acronym for **M**eta**H**euristics **S**ubjectivity **L**exicons. We formulated the problem as an optimization problem, and tried to solve it using three metaheuristic techniques. In an overview of our method, we grouped the words into the subjective and objective ones using a metaheuristic method on the training dataset; in this way, we created a subjectivity lexicon, which showed if each word was subjective or objective. We wanted to count the number of subjective and objective words in every record, and decided whether a record in the test dataset was subjective or objective based on these values. After creation of the lexicon, a model was trained on the train data based on these two features (number of subjective and objective words), and then it was used to classify the test data.

In other words, our algorithm has two phases; the first phase is to generate a subjectivity lexicon based on the train data and calculate the features based on this lexicon for the whole dataset, and the second phase is to train a classifier on the train dataset and to apply it on the test dataset. The two features used here are the number of subjective and objective words in each record based upon the created lexicon.

For the first phase of our algorithm, which is the creation of the sentiment lexicons, we incorporated three metaheuristics, discussed as what follow.

3.1. Metaheuristics

First, we introduce the three metaheuristics we use in our method as follows. These metaheuristics are GA, SA, and ARO.

3.1.1. Genetic algorithm (GA)

In GA, a solution is represented as a chromosome. A certain number of chromosomes form a population, and the goal of GA is to gradually make better populations in terms of fitness of their chromosomes. Each individual has a fitness value, which is calculated using the fitness function.

In each iteration of GA, two chromosomes are selected as parents, and with a cross-over operation, they produce new children. The children may also be mutated. Then the children may replace the existing individuals in the population.

3.1.2. Simulated annealing (SA)

Simulated annealing (SA) is a probabilistic metaheuristic that was first used to solve the Travelling Salesman Problem [37]. SA is based upon a metallurgy process named annealing. The SA approach uses a base solution, and tries to improve it. It includes a *Temperature* parameter that is high at the beginning but in each iteration is multiplied by a factor named *Alpha*, which is between 0 and 1 and very close to 1, so that the temperature decreases slowly. When the temperature reaches a certain point named *Epsilon*, the algorithm stops. In each iteration, a new solution is generated in the neighborhood. This new solution is accepted if it has a lower “cost” (and so, it is a good trade) and the algorithm replaces the previous solution with a new one. However, new solutions with higher costs may be accepted. The acceptance of bad trades depends on the temperature and the difference of the costs. A cost function must be defined for solutions to compare them. A new solution is accepted if the criterion in (1) is satisfied,

$$e^{-D/T} > R(0,1) \qquad (1)$$

in which ΔD is the cost difference of the new solution and the previous one (it is negative for a good trade and positive for a bad trade), T is the temperature, and $R(0,1)$ is a random number between 0 and 1.

3.1.3. Asexual reproduction optimization (ARO)

In ARO, proposed by Farasat et al. [38], a single chromosome is considered as a parent. Then the following operations are done on the chromosome. First, a sub-string in this chromosome is chosen randomly and is mutated. The length of this substring, g, is also random. The mutated sub-string is called a larva. Then with a probability, p_c, the parent and the larva do a cross-over, and the result is named as a bud. If the fitness of a bud is better than the fitness of its parent, it replaces the parent. The probability p_c is calculated using (2):

$$p_c = \frac{1}{1+\ln g} \qquad (2)$$

3.2. Lexicon creation (Phase one)

In this phase, the metaheuristics are incorporated for the lexicon creation. Since we want to create subjectivity lexicons, the chromosomes in GA and ARO and the solution in SA are subjectivity lexicons. Each element in a chromosome represents a word in the corpora, and its value shows if the word is subjective or objective. For example, if the corpus contains 1000 words, each chromosome will consist of 1000 cells, corresponding to each word.

A simple chromosome is shown in figure 1:

Average	*Door*	*Great*	*Is*	*Not*	*Awful*	*Bad*
S	**O**	**S**	**O**	**S**	**S**	**S**

Figure 1. Chromosome representation.

In this figure, **S** represents Subjective and **O** represents Objective. Each cell can have two values: subjective and objective. The fitness of each chromosome shows how well it can distinguish between the subjective and objective records. The reason for choosing metaheuristics is that the size of the search space is 2^n, where n is the number of words. Algorithm (1) presents the fitness function of a chromosome k in dataset T. A chromosome classifies a training record by counting the number of subjective and objective words in it. If the classification is correct, the fitness increases by one. If not, the error is the subtraction of subjective and objective count of words in the record, and this error is subtracted from the fitness value as penalty. Our optimization problem is to minimize the penalty and maximize the reward. The SubjCount and ObjCount values are calculated according to (3) and (4).

$$SubjCount(T_i, k) = \sum_{w_j \in T_i} S(w_j, k) \tag{3}$$

$$ObjCount(T_i, k) = \sum_{w_j \in T_i} O(w_j, k) \tag{4}$$

in which, T_i is the ith record of dataset, and the lexicon used is the kth chromosome in the population. $S(w_j, k)$ and $O(w_j, k)$ are calculated as follow: S(w_j, k) = 1 if w_j = S in chromosome k, else it is 0, and O(w_j, k) = 1 if w_j = O in chromosome k, else it is 0.

Figure 2 shows a sample of SubjCount and ObjCount in a chromosome and a record.

```
Fitness(chromosome k, Dataset T)
fitness = 0
class = null
for each record $T_i$ in T
  SubjCount = 0
  ObjCount = 0
  for each word $w_{ij}$ in $T_i$
    if $V(k, w_{ij})$ = S // $V(k, w_{ij})$ is the value of wij in chromosome k
      SubjCount = SubjCount + 1
    if $V(k, w_{ij})$ = O
      ObjCount = ObjCount + 1
  end for
  if (SubjCount > ObjCount) class = Subjective
    else class = Objective
  if (label($T_i$) = class) fitness = fitness + 1
    else fitness = fitness – abs(SubjCount – ObjCount)
end for
return fitness
```

Algorithm 1. Fitness function for a chromosome

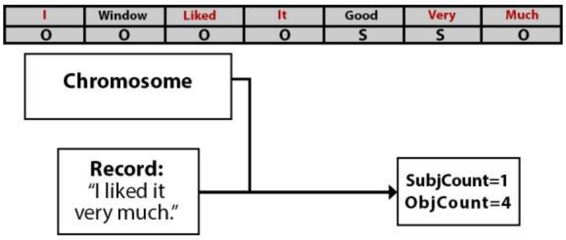

Figure 2. An example of SubjCount and ObjCount for a record [8].

In the SA algorithm, a cost function is used instead of a fitness function. Contrary to a fitness function in which a higher value means a better individual, in a cost function, a lower value indicates a better solution. We use the negative value of fitness function as the cost function in the SA approach, and do not alter the fitness function in other ways.

GA, SA, and ARO gradually converge to a good solution. In GA, after a certain number of iterations are met, the best chromosome in terms of fitness is taken as the subjectivity lexicon. In SA and ARO, the final solution is considered as the subjectivity lexicon. These lexicons can give us a deeper understanding of sentiment words, and may lead us to make better polarity lexicons, in which the words are labeled as positive or negative. Here, when we have the subjectivity lexicon, we use it to classify a text into a subjective and an objective one.

Since each individual is a lexicon, words should be selected for them. We want to distinguish between the objective and subjective words, and hence, the lexicons are built based on all of the words in the datasets. We discard all punctuation marks, and all of the words are converted to lower case. Though the punctuation marks are widely used in the literature for the sentiment analysis, we focus on creating lexicons based on words. The existing lexicons that we will compare our results with them do not include punctuation marks, and hence, we omit them.

3.3. Classification (Phase two)

For the classification of tweets, we need features. In our method, these features for each record are the number of subjective and objective words in a record, according to the subjectivity lexicon. As seen earlier, the number of subjective words of a record in the lexicon is named SubjCount, and the number of objective words is named ObjCount. These two meta-level features are given to a classifier. For validation of our method, we use 10-fold cross-validation. In each iteration, the 9 training folds are used as the training dataset, and a subjectivity lexicon is created for the training dataset. Then the two meta-level features SubjCount and ObjCount are calculated for all of the records in the datasets based upon the created lexicon. A model is built with these features on the 9 training folds, and is applied on the test fold.

4. Experimental evaluation

In this section, we describe our experiments of the proposed method. We introduce the datasets that were used in our experiments, the parameters, and the results. We then discuss the results obtained.

4.1. Datasets

MHSL was run on three datasets. These datasets are comprised of tweets people posted in Twitter.

These three datasets are Stanford Twitter Sentiment (STS) [39][2], Sanders[3], and SemEval-2013 [40]. The SemEval dataset is taken from the task 2 of the 2013 challenge, which is named as the sentiment analysis in Twitter. In these datasets, tweets are generally labeled as positive, negative, and neutral. We consider the neutral class as the objective class, and positive and negative classes as the subjective class. We used these datasets in accordance of [3]. The STS dataset has been introduced by Go et al. [39]. It consists of tweets that are manually annotated. The Sanders dataset consists of tweets with hashtags containing the names of big technology companies. The SemEval-2013 dataset was built for the Twitter sentiment analysis task in SemEval-2013 [40]. The number of objective and subjective records in each dataset is shown in table 1.

Table 1. Labels of dataset records.

	STS	Sanders	SemEval
Subjective tweets	139	1224	5097
Objective tweets	359	2502	4585

We consider all of the words in each dataset. The number of individual words in each dataset is shown in table 2.

Table 2. Number of words in each dataset.

	STS	Sanders	SemEval
Number of words	1880	4409	7363

These datasets were chosen from Twitter. The sentiment analysis of Twitter is a challenging task because the Twitter users have their own culture, tweet about various subjects unlike a specific subject [41], and neutral tweets are more than subjective ones. Also since tweets are short, the sentiment cues from a tweet are very limited [42]. Due to these challenges, we proposed this method, which makes use of all of the words present in the tweets because each word can have an influence on the classification. There are other datasets as well such as the dataset used in [43] but their style of writing is rather straightforward, and is not like complicated tweets.

4.2. Experimental setup

Programs in C# were written for experiments and implementations of GA, SA, and ARO on the datasets. We also used 10-fold cross-validation based on two meta-level features, *SubjCount* and *ObjCount*. The lexicons were created based on the 9 training folds, and then the values for these meta-level features were calculated for all of the records in datasets, and a classifier was trained on these two meta-level features on the training dataset (9 folds) and tested on the test dataset (1 fold). The classifier used was Bayes Network in Weka. The reason we chose this classifier was that it yielded better results, showing that the SubjCount and ObjCount features were relatively independent.

GA, SA, and ARO were run 10 times on each of the runs. The measures were averaged for each of the 100 runs of the algorithms.

For GA, the number of population was set to 2000. The cross-over was uniform, with a probability rate of 0.8. In the uniform cross-over, each gene of the two children was selected either from the first or the second parent.

The mutation rate was 0.05, and in the mutation phase, each of the genes of the chromosomes was changed randomly by a probability of 0.1. The algorithm was run for 250,000 iterations. The method for selecting chromosomes for cross-over was roulette-wheel selection. This method was also used for selecting chromosomes to be replaced. The chromosomes were initialized by random values, i.e. each gene of each chromosome had the value of “S” or “O”, each with a 50% chance at the beginning of the algorithm.

For SA, the initial temperature was set to 4000. The alpha value was set to 0.999, and the epsilon was 1, so the algorithm would run until temperature decreases below 1. In SA, a neighbor solution must be created every time the temperature changes. Creating new neighbors was done by changing the value of each word (Subjective or Objective) to a random value by a probability of 10%.

For ARO, the algorithm was run for 200,000 iterations. This algorithm does not have other parameters.

To get a better understanding of the words, we calculated the objectivity and subjectivity scores for them. The objectivity/subjectivity score for a word is the percentage of times that word has been objective/subjective in the whole 100 runs of GA (product of 10 folds and 10 runs) in the STS dataset. For example, if a word has been objective 97 times and subjective in 3 runs in the 100 runs, its objectivity score is 0.97, and its subjectivity score is 0.03. We chose the STS dataset because its tweets were general, while the tweets in the Sanders dataset were about technology companies.

[2] http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip

[3] http://www.sananalytics.com/lab/twitter-sentiment/

4.3. Results

In this section, we compare our results with the results from baselines. We have taken the baseline results from Bravo-Marquez et al. [3]. Seven of these baselines are sentiment lexicons, and three of them are the results from methods. Table 3 shows the comparison of results of our methods using the three metaheuristics, on three datasets, comparing them with baselines.

The value of f-measure (or as seen in other works, F_1) for the class C_a is calculated as follows:

$$F_1(C_a) = 2.\frac{\text{Precision}(C_a).\text{Recall}(C_a)}{\text{Precision}(C_a) + \text{Recall}(C_a)} \quad (5)$$

The F-measure reported for the methods and baselines in table 3 is the average of F-measure for objective and subjective classes.

In this table, it can be seen that the three methods perform better in terms of accuracy than the f-measure. For example, GA + Bayes Network outperforms all the baselines in terms of accuracy in two datasets but is better than baselines in terms of f-measure in one dataset. Furthermore, the difference of the results of our methods and the average accuracy of other methods is higher than the difference of f-measure of results of our method and the average F-measure of other methods. A higher value for accuracy and F-measure is desired, and in this table, it is shown that our results are higher and thus better in most of the datasets.

The results shown in table 3 show that GA outperforms the other two metaheuristics that were used, namely SA and ARO. Also it is shown that ARO works slightly better in our settings than SA, yielding better results.

We now analyze the lexicons created using GA. Tables 4 and 5 show the 50 most subjective and most objective words in the STS dataset. Since GA was run for 10 times on each of the folds, 100 subjectivity lexicons were generated. The percentage of a word being subjective or objective in these 100 lexicons is considered as its subjectivity or objectivity score, and these tables show the words with the highest scores in subjectivity and objectivity.

Table 3. 10-fold cross-validation subjectivity classification results [3].

Methods	**STS**		**Sanders**		**SemEval**	
	Accuracy	**F_1**	**Accuracy**	**F_1**	**Accuracy**	**F_1**
Sent140	0.688	0.596	0.623	0.574	0.620	0.573
SSPOL	0.769	0.772	0.636	**0.681**	0.683	**0.719**
EmoLex	0.618	0.602	0.599	0.590	0.611	0.610
SenticNet	0.682	0.673	0.611	0.605	0.593	0.594
Bing Liu’s Lexicon	0.750	0.740	**0.664**	0.650	0.663	0.660
NRC-Hashtag	0.631	0.610	0.622	0.620	0.550	0.530
Sent140 Lexicon	0.701	0.680	0.623	0.630	0.608	0.602
AFINN	**0.792**	**0.796**	0.649	0.640	**0.703**	0.700
SWN3	0.742	0.730	0.618	0.620	0.630	0.630
OpinionFinder	0.744	0.740	0.620	0.610	0.613	0.611
MHSL (GA + Bayes Network)	0.841	0.785	0.764	0.713	0.609	0.604
MHSL (SA + Bayes Network)	0.791	0.744	0.726	0.660	0.555	0.518
MHSL (ARO + Bayes Network)	0.783	0.710	0.731	0.670	0.593	0.589

It can be seen in table 5 that most of the objective words are those that are usually omitted from datasets as stop-words. However, there are some stop-words present in table 4. It shows that some of the stop-words are indeed important in the subjectivity classification. In fact, in paper [44], it is argued that it is better to keep all of the stop-words than omitting all of them because of the knowledge they may have in sentiment analysis. Our method shows that when people tweet subjectively or objectively, their choice of words differs, even in using the words that are deemed stop-words. They use the words such as “excellent” and “awful” when posting a subjective tweet but they also tend to use words such as “than”, “go”, and “if” as well.

Table 4. The most subjective words based on subjectivity score.

Word	Subjectivity score	Word	Subjectivity score
Wrong	1	Again	1
Very	1	Great	1
;)	1	Like	1
Am	1	From	1
Awesome	1	Are	1
Lol	1	Lebron	1
Got	1	Obama	1
Hate	1	It’s	1
A	1	Good	1
Reading	1	But	1
My	1	I	1
Fail	1	His	1
More	1	Than	1
Going	1	Place	1
Much	1	So	1
Go	1	Lakers	1
Me	1	Time	1
You	1	Best	1
Back	1	Never	1
It	1	Love	1
Is	1	The	1
Sad	0.99	Can’t	0.99
Last	0.99	If	0.99
Pretty	0.99	Warner	0.99
Clinton	0.99	:)	0.99

Table 5. The most objective words based on objectivity score.

Word	Objectivity score	Word	Objectivity score
2	1	Of	1
Your	1	And	1
With	1	JQuery	1
How	1	About	1
To	1	Was	1
-	1	Weekend	1
Canon	1	On	1
At	1	Stanford	1
See	1	May	1
China	1	Up	1
Movie	1	Saw	1
Here	1	Dentist	1
Safeway	1	Three	1
Joining	1	Found	0.99
40D	0.99	Visa	0.99
Twitter	0.99	Bill	0.99
Years	0.99	Card	0.99
Top	0.99	San	0.99
Francisco	0.99	Check	0.99
Super	0.99	An	0.99
Get	0.99	Playing	0.99
Start	0.99	Did	0.98
Does	0.98	RT	0.97
Before	0.97	School	0.97
NCAA	0.97	Need	0.96

On the other hand, the subjective words are those that imply subjectivity. Sentiment lexicons are created based on the words that are considered subjective. The words that have a high subjectivity score can be considered sentiment words, and their polarity can be determined, either by hand or by an automatic method, which can be similar to MHSL.

If we consider all the words with a subjectivity score of higher than a certain threshold as subjective, we can create polarity lexicons based on each dataset, which will be our future work.

We used the subjectivity lexicons created for each dataset for the subjectivity classification of other datasets. The cross-transfer subjectivity classification performance is shown in table 6. The two meta-level features *SubjCount* and *ObjCount* were calculated for each record, and a 10-fold cross-validation was performed.

Table 6. Performance of cross-transfer subjectivity classification.

	STS dataset		Sanders dataset		SemEval dataset	
	Acc	F_1	Acc	F_1	Acc	F_1
Lexicon Created for STS			56.82	56.74	60.68	59.87
Lexicon Created for Sanders	74.94	67.39			59.29	58.64
Lexicon Created for SemEval	72.09	41.89	53.78	53.65		

Now we compare our method to the existing baselines using the Wilcoxon Signed Rank Test. The results of the test can be seen in tables 7 and 8. These results show that our method is superior to the baselines in terms of accuracy, performing better than most of them.

One of the two close baselines to our method in terms of accuracy is AFINN, which has a performance near MHSL in the STS dataset. However, its performance in the Sanders dataset is significantly worse than MHSL, and its performance in the SemEval dataset is significantly better than MHSubLex.

In terms of f-measure, SSPOL is on the same level of our algorithm. The Z value is the test statistic that shows how well our algorithm performs comparing to other methods based on the results on different datasets.

If its value for one baseline is close to zero, it shows that the performance of our algorithm is close to that baseline. If its value for baseline 1 is more negative than for baseline 2, it shows that our algorithm performs better comparing to baseline 1 than baseline 2. Z is calculated in IBM SPSS.

Table 7. Results of Wilcoxon Signed Rank Test for accuracy values of MHSubLex with GA and baselines.

	Z	Asymp Sig (2-tailed)	Wins for MHSL	Losses for MHSL
Sent140	-1.069	0.285	2	1
SSPOL	-0.535	0.593	2	1
EmoLex	-1.069	0.285	2	1
SenticNet	-1.604	0.109	3	0
BingLiu's Lex	-1.069	0.285	2	1
NRC-hashtag	-1.604	0.109	3	0
Sent140 Lex	-1.604	0.109	3	0
AFINN	-0.535	0.593	2	1
SWN 3	-1.069	0.285	2	1
OpFinder	-1.069	0.285	2	1

Table 8. Results of Wilcoxon Signed Rank Test for F-measure values of MHSubLex with GA and baselines.

	Z	Asymp Sig (2-tailed)	Wins for MHSL	Losses for MHSL
Sent140	-1.604	0.109	3	0
SSPOL	0	1.000	2	1
EmoLex	-1.069	0.285	2	1
SenticNet	-1.604	0.109	3	0
Bing Liu's Lex	-0.535	0.593	2	1
NRC-hashtag	-1.604	0.109	3	0
Sent140 Lex	-1.604	0.109	3	0
AFINN	-0.535	0.593	1	2
SWN 3	-1.069	0.285	2	1
OpFinder	-1.069	0.285	2	1

Overall, MHSubLex performs better than all of the baselines in the STS and Sanders datasets in terms of accuracy. However, in the SemEval dataset, its results are not better than all methods. Several reasons can be cited. One of the reasons is that this dataset is a large and challenging one, which was built for a competition.

The challenge in classifying its tweets can be seen in other baselines as well. As it can be seen in table 3, the other methods perform significantly worse on this dataset than on the other datasets. The other reason that our methods do not perform well in this dataset is that the idea of grouping the words into subjective and objective ones is not very suitable for this dataset. There are many objective tweets in this dataset that use the words that are deemed subjective, and vice versa, and this can cause our method to perform poorer on this dataset.

This problem is also apparent in other baselines, as there are numerous words that are included in the sentiment lexicons that are present in the objective tweets, and there are tweets that are subjective but do not contain subjective words.

For example, the average number of Bing Liu's lexicon words in subjective and objective tweets in SemEval dataset are 1.28 and 0.57, respectively. However, these numbers for the STS dataset are 1.41 and 0.28, respectively. It shows that the presence of subjective words in objective tweets is more prevalent in the SemEval dataset than the STS dataset, and the presence of subjective words in subjective tweets is less prevalent in SemEval than STS.

As an example, the tweet "C'mon Cam and the Panthers! U r the missing link in my quinfecta weekend. Hard to do...App win, UNC win, State loss, Dook loss, Panther win" in SemEval has 6 subjective words, according to the Bing Liu's lexicon. However, it is labeled as objective in this dataset.

The subjectivity score box-plot for words in each of the three lexicons can be seen in figures 3, 4, and 5 for each of the three datasets. They show the first, second, and third quartile of subjectivity scores, from left to right. If the box in the box-plot is inclined to the left side, it shows that most of the words have low subjectivity scores.

As it can be seen in figures 3 to 5, most of the words in the datasets are more objective than subjective because the subjectivity scores in the datasets are relatively low.

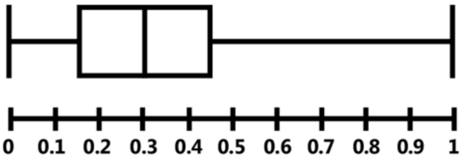

Figure 3. Box plot for subjectivity score of words in STS dataset.

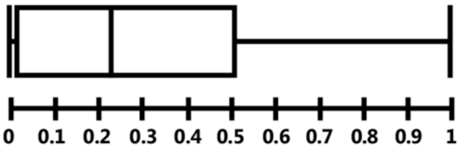

Figure 4. Box plot for subjectivity score of words in Sanders dataset.

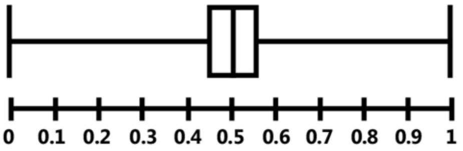

Figure 5. Box plot for subjectivity score of words in SemEval dataset.

Figure 5 shows that 25, 50, and 75 percent of the words in the STS dataset have subjectivity scores less than 0.26, 0.31, and 0.45, respectively. Figure 6 demonstrates that the subjectivity scores of almost 75% of words in the Sanders dataset are below 0.5. This shows that most of the words used in tweets tend to be more objective than subjective.

The subjectivity score of words such as “never”, “am”, “if”, and “but” are significantly high in the datasets. However, these words are not present in lexicons such as AFINN and Bing Liu’s lexicon. Our method shows that by considering these words as subjective words, the accuracy increases.

5. Conclusion and future work

We addressed the problem of subjectivity classification in micro-blogs. The main goal of our work was to improve the accuracy of classification of the tweets into subjective and objective classes, and meanwhile, find out what the underlying concept in subjectivity was. We hypothesized that the words could be grouped into the subjective and objective ones, and by counting the number of subjective and objective words in a tweet, it could be inferred whether the tweet was subjective or objective.

The results obtained demonstrate that our hypothesis is true because we have examined it on three datasets, and have gained high accuracy and f-measure values. Moreover, due to the stochastic nature of metaheuristics, we have run the algorithms 10 times for each dataset. The classification using our method is significantly better than the baselines.

Our work was focused on finding the objective and subjective words in order to generate a subjectivity lexicon. The lexicon was created using three evolutionary methods: GA, SA, and ARO. GA outperformed the other two metaheuristics in terms of accuracy and F-measure.

Using these algorithms, we could build subjectivity lexicons based on training datasets. For each tweet, two meta-level features were extracted, the count of objective words and the count of subjective words. These features were used to classify tweets into subjective and objective.. A model was trained on the training dataset using these two meta-level features, and then the model was applied on the test dataset to calculate the accuracy and f-measure values. This process was repeated ten times in a 10-fold cross-validation scheme.

Once a subjectivity lexicon is built, its uses are two-fold: (i) it can be used for classifying the text. The number of subjective words and the number of objective words can be calculated using them, and these two features can be used for classification; and (ii) it can show the content providers that which words are subjective and which words are objective; hence, they can choose words for their tweets wisely to send opinionated or objective messages.

Our method outperformed baselines on accuracy on at least two of the three datasets assessed. Since the fitness function was based upon accuracy, the method prevailed baselines in several accuracy points. The f-measure of our method, though still higher than most baselines, was closer to them and this is because of the unbalanced datasets and fitness function.

Our work can contribute to build a sentiment lexicon using subjective words, in which these words are divided into positive and negative words. In this case, we incorporate the words that are considered as subjective in our work, and try to group them into positive and negative words.

In our future works, we try to build a sentiment lexicon based on the subjectivity lexicon that we have created. We also will explore the notion of subjectivity in words. We also want to explore other metaheuristics such as the firefly algorithm, which has been successfully incorporated in [45]. Also, other feature extracting methods, such as [46] will be explored.

References

[1] Liu, B. (2015). Sentiment analysis and opinion mining, Morgan & Claypool Publishers.

[2] Fersini, E., Messina, E. & Pozzi, F. A. (2014). Sentiment analysis: Bayesian Ensemble Learning, Decision Support Systems, vol. 68, pp. 26-38.

[3] Bravo-Marquez, F., Mendoza, M. & Poblete, B. (2014). Meta-level sentiment models for big social data analysis, Knowledge-Based Systems, vol. 69, pp. 86-99.

[4] Ritter, A., Clark, S. & Etzioni, O. (2011). Named entity recognition in tweets: an experimental study, Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP'11, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1524–1534.

[5] Wiebe, J., Bruce, R. F. & O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications, in Proceedings of the Association for Computational Linguistics (ACL-1999), pp. 246-253.

[6] Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., & Reyes, A. (2015). SemEval 2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 470-478.

[7] Hernańdez Farías, D., Patti, V. & Rosso, P. (2016). Irony Detection in Twitter: The Role of Affective Content. In: ACM Trans. Internet Technol. vol. 16, no. 3, pp. 1-24.

[8] Keshavarz, H. & Abadeh, M. S. (2016). SubLex: Generating Subjectivity Lexicons Using Genetic Algorithm for Subjectivity Classification of Big Social Data, in The 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC2016), IEEE, pp. 136-141.

[9] Wiebe, J. & Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts, Computational Linguistics and Intelligent Text Processing, pp. 486-297.

[10] Pang, B. & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in Proceedings of the ACL, pp. 271, 2004.

[11] Pang, B. Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques, in Proceedings of the ACL, Association for Computational Linguistics, pp. 79-86.

[12] Wiebe, J. (2000). Learning subjective adjectives from corpora, in Proceedings of National Conf. on Artificial Intelligence (AAAI-2000), pp. 735-740.

[13] Barbosa, L. & Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data, in Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, pp. 36-44.

[14] Jiang, L., Yu, M., Zhou, M., Liu, X. & Zhao, T. (2011). Target-dependent twitter sentiment classification, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, pp. 151-160.

[15] Pak, A. & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining, In LREC, vol. 10, pp. 1320-1326.

[16] Wang, X., Wei, F., Liu, X., Zhou, M. & Zhang, M. (2011). Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach, in Proceedings of the 20th ACM international conference on Information and knowledge management, pp. 1031-1040.

[17] Liu, K. L., Wu-Jun, L. & Guo, M. (2012). Emoticon Smoothed Language Models for Twitter Sentiment Analysis, In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, pp. 1678-1684.

[18] Yu, H. & Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences, in Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2003), pp. 129-136.

[19] Koto, F. & Adriani, M. (2015). A Comparative Study on Twitter Sentiment Analysis: Which Features are Good?, In Natural Language Processing and Information Systems, Springer International Publishing, pp. 453-457.

[20] Mansour, R., Hady, M. F. A., Hosam, E., Amr, H. & Ashour, A. (2015). Feature Selection for Twitter Sentiment Analysis: An Experimental Study, In Computational Linguistics and Intelligent Text Processing, Springer International Publishing, pp. 92-103.

[21] Bradley, M. M. & Lang, P. J. (2009). Affective Norms for English Words (ANEW) Instruction Manual and Affective Ratings, Technical Report C-1, The Center for Research in Psychophysiology University of Florida, 2009.

[22] Nielsen, F. (2011). A new anew: evaluation of a word list for sentiment analysis in microblogs, in Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages.

[23] Mohammad, S. M. & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon, Computational Intelligence, vol. 29, no. 3, pp. 436-465.

[24] Mohammad, S. M., Kiritchenko, S. & Zhu, X. (2013). Nrc-canada: building the state-of-theart in sentiment analysis of tweets, in Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013), pp. 321-327.

[25] Wilson, T., Wiebe, J. & Hoffmann, P. (2005). Recognizing contextual polarity in phraselevel sentiment analysis, in Proceedings of Human Language Technologies Conference/Conference on Empirical

Methods in Natural Language Processing (HLT/EMNLP 2005), Canada, pp. 347-354.

[26] Baccianella, S., Esuli, A. & Sebastiani, F. (2010). 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, Malta, pp. 2200-2204.

[27] Go, A., Bhayani, R. & Huang, L. (2010). Twitter sentiment classification using distant supervision, Technical report, Stanford University.

[28] Thelwall, M., Buckley, K. & Paltoglou, G. (2012). Sentiment strength detection for the social web, Journal of the American Society for Information Science and Technology, vol. 63, no. 1, pp. 163-173.

[29] Cambria, E., Havasi, C. & Hussain, A. (2012). SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis, In FLAIRS conference, pp. 202-207.

[30] Das, A. & Bandyopadhyay, S. (2010). Subjectivity detection using Genetic Algorithm, the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA10), Lisbon, Portugal.

[31] Abbasi, A., Chen, H. & Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums, ACM Transactions on Information Systems, vol. 26, no. 3, pp. 12.

[32] Basari, A. S. H., Hussin, B., Ananta, I. G. P., & Zeniarja, J. (2013). Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization, Procedia Engineering 53, pp. 453-462.

[33] Bai, X., Padman, R. & Airoldi, E. (2004). Sentiment extraction from unstructured text using tabu search-enhanced markov blanket, Carnegie Mellon University, School of Computer Science,[@Institute for Software Research International].

[34] Baldominos Gómez, A., Mingueza, N. L. & García del Pozo, M. C. (2015). OpinAIS: An Artificial Immune System-based Framework for Opinion Mining, International Journal of Artificial Intelligence and Interactive Multimedia, vol. 3, no. 3, pp. 25-34.

[35] Arora, S., Mayfield, E., Penstein-Rosé, C. & Nyberg, E. (2010). Sentiment classification using automatically extracted subgraph features, in Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, Association for Computational Linguistics, pp. 131-139.

[36] Carvalho, J., Prado, A. & Plastino, A. (2014). A Statistical and Evolutionary Approach to Sentiment Analysis, Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on. vol. 2, IEEE, pp. 110-117.

[37] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, vol. 220, no. 4598, pp. 671-680.

[38] Farasat, A., Menhaj, M. B., Mansouri, T. & Moghadam, M. R. S. (2010). ARO: a new model-free optimization algorithm inspired from asexual reproduction, Applied Soft Computing 10, no. 4, pp. 1284-1292.

[39] Go, A., Bhayani, R., Huang, L. (2010). Twitter sentiment classification using distant supervision, Technical report, Stanford University.

[40] Nakov, P., Kozareva, Z., Ritter, A., Rosenthal, S., Stoyanov, V., & Wilson, T. (2013) Semeval-2013 task 2: Sentiment analysis in twitter, In Proceedings of the 7th International Workshop on Semantic Evaluation, pp. 312-320.

[41] Da Silva, N. F. F, Hruschka, E. R. & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles, Decision Support Systems, vol. 66, pp. 170-179.

[42] Hassan, A., Abbasi, A. & Zeng, D. (2013). Twitter sentiment analysis: A bootstrap ensemble framework, Social Computing (SocialCom), 2013 International Conference on. IEEE, pp. 357-364.

[43] Pang, B & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in Proceedings of the 42nd annual meeting on Association for Computational Linguistics, pp. 271-278.

[44] Saif, H., Fernandez, M., He, Y. & Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of Twitter, In Proceedings of the 9th language resources and evaluation conference (LREC), pp. 810-817.

[45] De Paula, L. C. M, Soares, A. S., de Lima, T. W., Delbem, A. C. B., Coelho, C. J. & Arlindo Filho, R. G. (2014). A gpu-based implementation of the firefly algorithm for variable selection in multivariate calibration problems, PloS one, vol. 9, no. 12: e114145.

[46] Golpar-Rabooki, E., Zarghamifar, S., & Rezaeenour, J. (2015). Feature extraction in opinion mining through Persian reviews, Journal of AI & Data Mining, vol. 3, no. 2, pp. 169-179.