Int. J. Nonlinear Anal. Appl. 12 (2021) No. 2, 1769-1784
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.23334.2522

A hybrid semi-supervised boosting to sentiment analysis

Jafar Tanha[a,*], Solmaz Mahmudyan[b], Ahmad Farahi[b]

[a]*Electrical and Computer Engineering Department, University of Tabriz, Tabriz, Iran*
[b]*Computer Engineering Department, Payame-Noor University, Tehran, Iran*

(Communicated by Madjid Eshaghi Gordji)

Abstract

In this article, we propose a hybrid semi-supervised boosting algorithm to sentiment analysis. Semi-supervised learning is a learning task from a limited amount of labeled data and plenty of unlabeled data which is the case in our used dataset. The proposed approach employs the classifier predictions along with the similarity information to assign label to unlabeled examples. We propose a hybrid model based on the agreement among different constructed classification model based on the boosting framework to assign final label to unlabeled data. The proposed approach employs several different similarity measurements in its loss function to show the role of the similarity function. We further address the main preprocessing steps in the used dataset. Our experimental results on real-world microblog data from a commercial website show that the proposed approach can effectively exploit information from the unlabeled data and significantly improves the classification performance.

Keywords: Semi-supervised learning, Sentiment Analysis, Persian Language, Boosting, Similarity Function.

1. Introduction

The growth of the internet usage has led to the creation of massive texts containing the opinions of people. Awareness of people's opinions is crucial for many decision-making aspects. One of the traditional approaches is to employ the text mining approaches, a branch of data mining, which extracts useful information from the text [2][17]. However, more advanced methods are needed to extract the opinions of the users. Sentiment analysis is the solution to automatically analyze these

*Corresponding author
Email address: tanha@tabrizu.ac.ir (Jafar Tanha)

Received: April 2021 *Accepted:* June 2021

comments. The aim of sentiment analysis is to analyze and review the views of the people [1]. Sentiment analysis is also called opinion mining [5].

Basically, the main approaches to sentiment analysis are as follows [35]: machine learning based approach [16], lexical-based approach [15], and hybrid (combined) approach [27]. In this article, the main focus is on the machine learning algorithms to sentiment analysis. In recent years, sentiment analysis have attracted the attention of the researchers, see [44], [1], and [43]. Projects on beliefs are the initial pioneering research studies in this field. Over the past few years, many machine learning techniques have been proposed for sentiment analysis [3][3][33]. These studies mainly use different supervised machine learning algorithms, such as Naive Bayes (NB), Maximum Entropy (ME), Stochastic Gradient Descent (SGD), and Support Vector machine (SVM) to classify human sentiments. In these studies data are often labeled. However, labeling data in many domains is not easy task and need human effort. Using semi-supervised learning is a solution to handle these issues which is the main goal of this article.

Semi-supervised learning employs a large amount of unlabeled data along with a small set of labeled data to build a better classifier, while it needs less human effort and yields a high performance classification model [42]. The reason for this improvement is due to unlabeled data, which enables the system to model the intrinsic structure of the data more accurately. There are different kinds of approaches to semi-supervised learning, such as self-training and co-training [42], generative models, graph-based [21], boosting-based approaches, like MSAB [41] and MSSBoost [38]. These approaches are mainly proposed to general datasets and are not directly applicable to sentiment analysis. Recently, several semi-supervised approaches are extended to the sentiment analysis for building a strong recognition model [6][46]. We consider boosting-based approach in this study.

In this article, we extend our recent approaches to multiclass semi-supervised boosting for sentiment analysis [41][38]. Boosting framework is a general ensemble method for improving the classification performance of any learning algorithm, also producing an accurate classifier, by combining rough and moderately weak classifiers. In our recent approach, we used the margin on labeled data, the similarity among labeled and unlabeled data, and the similarity among unlabeled data in an exponential loss function to multiclass semi-supervised classification problem. Based on the experimental results, the proposed algorithm outperforms the state-of-the-art boosting algorithms. However, finding a suitable similarity measurement is not easy task. The used Radial Basis Function (RBF) needs tuning parameter and may not find properly similarity information in practice. Our main goal in this study is to find a suitable similarity measurement for the sentiment analysis tasks using semi-supervised boosting framework.

We here focus on similarity functions in the proposed boosting approach to multiclass semi-supervised classification problem. As addressed in [38], similarity information from the unlabeled data plays main role in the loss function formulation of the recent semi-supervised boosting algorithms. Therefore, the quality of this measurement has high impact to achieve good classification performance. We consider this issue in sentiment analysis problem and propose an ensemble method to handle the problem. We therefore propose a hybrid boosting based model to sentiment analysis, named MS3A-Ensemble. The proposed algorithm combines different boosting-based semi-supervised algorithms in order to improve the classification performance of the supervised base learner using a large set of unlabeled data. Our main contributions in this article are as follows:

- We first perform several preprocessing steps in Persian language, which has several issues and difficulties.
- We annotate less than one percent of data and use the rest of the data as unlabeled data in our proposed algorithm

- We then adapt a multiclass semi-supervised boosting framework using several similarity functions for text dataset.
- We finally propose a hybrid ensemble-based model to sentiment analysis based on the adapted semi-supervised boosting approach.

The present study regarded this topic by collecting data from a commercial Website called *Digikala (http://www.digikala.com/Search/Category-mobile-phone)*, which is one of the most active e-commerce websites in Iran. The collected data is comments on digital devices, like mobile phones. The data is originally fully unlabeled. We first annotate a small set of data, three different class labels are assigned to the comments: P for positive, N for negative, and O for neutral comments. Our experimental results on the collected dataset show that the proposed approach improves significantly the recognition rate in sentiment analysis. The results also indicate that the proposed hybrid model analyses the comments properly.

The rest of this article is organized as follows: section 2 reviews the related studies on machine learning approaches; section 3 introduces the multiclass semi-supervised algorithm and proposes algorithm for sentiment analysis; section 5 presents the experiment; section 6 addresses the results, and finally section 7 draws the main conclusions.

2. Related Literatures

The sentiment analysis can be applied on various topics, such as movie reviews, product reviews, news, and blogs [19]. In general, there are three different approaches for sentiment analysis that are as follows [35]: Lexical analysis, Machine learning based analysis, and Hybrid analysis.

Lexical analysis uses a lexicon sentiment that includes sentimental words [35]. This method can be divided into two categories: dictionary based approaches that use dictionaries and corpus-based approaches, that use statistical methods to determine the polarity of sentiments [11]. However, in many languages, such as Persian, there is not enough lexicon resources to analysis data.

There are methods known as hybrid approaches which use the combination of lexical-based techniques and machine learning for sentiment analysis [18][28]. There are also several studies which use linguistic resources, such as Part-Of-Speech (POS), Lexicons, and writing style, see [18, 27].

Machine learning is a process of data usage that automatically builds a model, which uses features as input and provides a prediction as output. Most studies of sentiment analysis have used the machine learning algorithms to generate sentiment classification models. The main machine learning methods are: supervised, unsupervised, and semi-supervised learning. Supervised learning algorithms require a labeled training dataset. Several studies use feeling and hashtags to build a training set [35]. In [10], emotions are used as class labels to identify the polarity. This type of strategy is known as distant supervision. Other algorithms employ social network features, like followers/followee and links [16]. Unsupervised learning works with unlabeled data. It finds hidden structure in data. In this case, according to the semantic orientation of phrases that include adjectives or adverbs, a document is classified as positive or negative. In [31], three different methods are proposed for measuring the similarity between words using lexical-based, semantic, and distributional similarity methods. Since, labeled data is not used by unsupervised learning methods, it is expected that they will be less accurate than supervised learning methods. From this perspective, the systems of sentiment analysis mainly refer to the lack of labeled reviews [35].

Compared to supervised learning and unsupervised learning, semi-supervised learning takes note of labeled and unlabeled data during the training procedure. It can be said that the semi-supervised

algorithm is between supervised and unsupervised learning [35]. There are three approaches for analyzing sentiments; graph-based [32][41], wrapper-based (self-training [14], co-training [48]), and topic-based methods [46].

Generally, the graph-based methods distribute labels to unlabeled data using the structure of a graph [35][34][21]. The distribution process requires the calculation of the similarity between the data samples. But finding suitable similarity measures, especially for analyzing sentiments, is an unconditional task [35][34]. Cosine similarity works based on the bag of words representation. Another approach works with the similarity between topic instead of the similarity of sentiments [35][34]. Using graph-based methods and accessible social information has been motivated to find out the sentiments of users [3].

Another approach to semi-supervised learning is self-training [42]. The self-training algorithm has been used in various fields. For example, the AROW algorithm [9] uses self-training algorithm to predict the polarity of reviews. Zagibalov and Carroll [50] proposed a variation of self-training which adds lexical to words for the Chinese text. The self-training method for the Chinese microblog is also used by Liu et al. [25]. Qiu et.al [30] used a lexical iterative process to enhance the sentiment dictionary. There are methods that use the self-training algorithm to increase the size of the feature space [52]. The main motivation of these methods is to expand unlabeled comments, by finding the best adjustment from the lexical polarity.

Another paradigm of semi-supervised learning approach is co-training [48, 42]. The co-training algorithm uses two different feature subsets to sentiment analysis. Ning [48] discussed several co-training strategies for sentiment analysis. [49] used semi-supervised learning for opinion detection. They used self-training and co-training in their work. Liu et al. [23] used the co-training framework in their approach for tweet classification where the features were divided into two groups: textual and non-textual texts. These features were extracted and divided into two views for co-training in [24] and select more reliable tweets to increase performance.

A new concept has recently been reviewed by [46], who implemented a semi-supervised method for tweet sentiment analysis, which uses topic-based modeling. The authors performed clustering analysis and several classifiers in a training dataset. As a result, a combination of sentiments is obtained and then used to predict unlabeled tweet class. In order for the results to be satisfactory, a large set of labeled tweet is required. The authors used 9684 labeled tweets and 2 million unlabeled data. As the topic structure is formed by clustering the training dataset, the key disadvantage of this method is that no topic analysis occurs in tweet without labels, while this supplementary information can be useful.

Recently, in [18], a semi-supervised sentiment analysis approach is proposed to incorporate lexicon-based methodology with machine learning in order to improve sentiment analysis classification performance. It employs information gain and cosine similarity to revise the sentiment scores defined in SentiWordNet. A semi-supervised sentiment-discriminative objective is proposed using partial sentiment information of documents in [29]. This method not only reflects the partial sentiment information, but also preserves local structures induced from original distributed representation learning objectives by considering only sentiment relationships between neighboring documents. In [47], a novel semi-supervised method is addressed to derive and utilize the underlying sentiment of unlabeled examples using a deep generative model. This algorithm assumes that when given the aspect, the sentence is generated by two stochastic variables, i.e., the context variable and the sentiment variable.

Most recently, in [26], the problem of how to significantly reduce the amount of labeled training data required in fine-tuning language models is proposed for opinion mining. This algorithm addresses an opinion mining system developed over a language model that is fine-tuned through semi-supervised

learning with augmented data. A novel semi-supervised model based on dynamic threshold and multi-classifiers is proposed in [13]. This algorithm assigns auto-labeled to training data in an iterative way based on the used dynamic threshold algorithm, where a dynamic threshold function is addressed to set thresholds for selecting the auto-labeled examples.

In this article, we adapt one of the latest approach for semi-supervised learning to sentiment analysis [38][41].

3. Semi-supervised Learning

There are many different approaches to semi-supervised learning, see self-training [42], co-training [7], generative models, graph-based [4], margin-based approaches, like MSAB [41] and MSSBoost [38]. In this study, we focus on boosting framework which is one of the promising approach for supervised learning. We therefore employ one of the latest developed boosting approach for multiclass semi-supervised learning to handle sentiment analysis.

Recently, in [45] a new boosting algorithm has been introduced to learn the semi-supervised multiclass, which uses a similarity between predictions and data. In this study, the labels are mapped to n-dimensional space, but this mapping never leads to the formation of a classifier that minimizes margin cost properly. Most recently, we have proposed several approaches to handle the multiclass semi-supervised classification problem regarding the aforementioned issues, see [41][38]. In this study, we extend MSAB [41] and MSSBoost [38] to handle the sentiment analysis problem. Unlike many semi-supervised learning algorithms that are the expansion of a specific base classifier, MSAB and MSSBoost can boost any base classifiers. The MSAB and MSSBoost algorithms minimize the experimental error in labeled data and heterogeneous data in labeled data and unlabeled data based on cluster-based and manifold assumptions.

In this article, we use the mapping idea from MSAB and the similarity learning from MSSBoost to develop an approach to handle sentiment analysis problem.

3.1. Multiclass Semi-Supervised Setting

In multiclass semi-supervised learning for the labeled points X_l= $(x_1, x_2, ..., x_l)$ labels $\{1, ..., K\}$ are provided, and for the unlabeled points $X_u = (x_{l+1}, x_{l+2}, ..., x_{l+u})$, the labels are not known. We use the coding method in (3.2) to formulate the multiclass semi-supervised learning task.

Our algorithm needs a (symmetric) similarity matrix $S = [S_{i,j}]_{n\times n}$, where $S_{i,j} = S_{j,i}$ is the similarity between the points x_i and x_j. $S^{lu} = [S_{i,j}]_{n_l\times n_u}$ denotes the similarity matrix of the labeled and unlabeled data and $S^{uu} = [S_{i,j}]_{n_u\times n_u}$ of the unlabeled data. Our algorithm is a "meta-learner" that uses a supervised learning algorithm as its base learner.

$$y_{i,j} = \begin{cases} 1 & \text{if } i = j \\ \frac{-1}{K-1} & \text{if } i \neq j \end{cases} \tag{3.1}$$

where K is the number of classes. Then Y, the set of K-dimensional vectors, will be as follows:

$$Y = \begin{pmatrix} (1, -\frac{1}{K-1}, ..., -\frac{1}{K-1})^T \\ (-\frac{1}{K-1}, 1, ..., -\frac{1}{K-1})^T \\ \vdots \\ (-\frac{1}{K-1}, , ..., -\frac{1}{K-1}, 1)^T \end{pmatrix} \tag{3.2}$$

where $Y_i \in Y$ and $\sum_{j=1}^{K} y_{i,j} = 0$.

We assume that the labeled and unlabeled data are drawn independently from the same data distribution. In applications of semi-supervised learning normally $l \ll u$, where l is the number of labeled data and u is the number of unlabeled data.

3.2. The MSAB Algorithm

We first present the MSAB algorithm and then extend it to the sentiment analysis problem. The MSAB algorithm uses a number of weak classifiers through the training procedure. Each classifier has a special weight. When the conditions for the end of the algorithm are exceeded, the weighted combination of the classifiers is used to build the final classification model. In more details, the algorithm first begins with labeled data, at each iteration a set of newly labeled data (pseudo-labeled assign to unlabeled data) is selected to build the current classification model. For this algorithm, we need to find four main components which are: 1) weights for labeled and unlabeled data, 2) weights for the built classifiers, 3) step size, and 4) final classification model [41][38]. We here address how we derive these factors.

Assume that $H^t(x) : X \to R^k$ defines the linear combination of the classification models after $t-1$ iterations, then in t iteration, $H^t(x)$ is calculated as follows:

$$H^t(x) = H^{t-1}(x) + \beta^t h^t \tag{3.3}$$

where $\beta^t \in R$ is the weight of the base classifier $h^t(x)$ and $h^t(x)$ is a multiclass learner. The weight for each labeled example is computed as follows:

$$W_i = \exp(\frac{-1}{k}(H_i^{t-1}, Y_i)) \tag{3.4}$$

The weights for unlabeled data are also obtained as follows:

$$\begin{aligned} P_{i,k} = &\sum_{j=1}^{n_l} S^{lu}(x_i, x_j) e^{(\frac{-1}{K-1} H_i^{t-1}.e_k)} \delta(Y_i, e_k) \\ &+ \sum_{j=1}^{n_u} S^{uu}(x_i, x_j) e^{(\frac{1}{K-1}(H_j^{t-i} - H_i^{t-1}).e_k)} e^{\frac{1}{K-1}} \end{aligned} \tag{3.5}$$

Hence,

$$\hat{Y} = \arg\max_k (P_{i,k}) \tag{3.6}$$

and the value of β will be as follows:

$$\begin{aligned} \beta = \frac{(K-1)^2}{K} \Big(&\log(K-1) \\ &+ \log\big(\frac{\sum_{i \in n_u} \sum_{k \in l} P_{i,k} \delta'(h_i^t.e_k, P_i = k) + \sum_{\substack{i \in n_l \\ h_i^t = Y_i}} W_i}{\sum_{i \in n_u} \sum_{k \in l} P_{i,k} \delta'(h_i^t.e_k, P_i \neq k) + \sum_{\substack{i \in n_l \\ h_i^t \neq Y_i}} W_i}\big)\Big) \end{aligned} \tag{3.7}$$

ϵ^t represents the weighted error rate of the classifier as below:

$$\epsilon^t = \frac{\sum_{i \in n_u} \sum_{k \in l} P_{i,k} \delta'(h_i^t.e_k, P_i \neq k) + \sum_{\substack{i \in n_l \\ h_i^t \neq Y_i}} W_i}{\sum_{i \in n_u} \sum_{k \in l} P_{i,k} + \sum_{i \in n_l} W_i} \tag{3.8}$$

Now by placing the relation (3.8) in (3.7) β^t is obtained as:

$$\beta^t = \frac{(K-1)^2}{K}\Big(\log(K-1) + \log(\frac{1-\epsilon^t}{\epsilon^t})\Big) \tag{3.9}$$

The resulting Algorithm is presented in Algorithm 2. AS shown, the weights are first calculated for each unlabeled and labeled sample. This calculation is perform based on the classifier prediction and the similarity information which we will discuss in Section 4.3. The algorithm then uses (3.5) to assign pseudo-label to the unlabeled data, and the value of (3.5) is considered as the weight to unlabeled data. In the next step, a newly-labeled set is used as training data for the new classifier. The algorithm then uses the value of $P_{i,k}$ as weight to sample data which will lead to a decrease of the value of the objective function. As shown in Algorithm 2, a new classifier is built at each iteration of the boosting process. The boosting process is repeated until it reaches a stopping condition.

Algorithm 1 MSAB

Initialize: L, U, S, $H^0(x) = 0$; L: Labeled data; U: Unlabeled data;
S:Similarity Matrix; $H^0(x)$:Ensemble of Classifiers; $t \leftarrow 1$;
while $(\beta^t > 0)$ and $(t < M)$ **do** // M is the number of iteration
  for each $x_i \in L$ **do**
    Compute W_i for labeled example x_i based on(3.4)
  for each $x_i \in U$ **do**
    Compute $P_{i,k}$ for unlabeled example x_i based on the pairwise similarity and classifier prediction using (3.5)
    - Assign pseudo-labels to unlabeled examples based on (3.6)
  - Normalize the weights of labeled and unlabeled examples
  - Sample a set of high-confidence examples from labeled and unlabeled examples
  - Build a new weighted classifier $h^t(x)$ based on the newly-labeled and original labeled examples
  - Compute the weights and β^t for each new classifier $h^t(x)$ using (3.7)
  - Update $H^t \leftarrow H^{t-1} + \beta^t h^t$
  - $t \leftarrow t + 1$
end while
Output: Generate final hypothesis based on the weights and classifiers

4. The Proposed Semi-Supervised Sentiment Analysis

In this section, we first present our proposed multiclass semi-supervised approach to sentiment analysis, called MS3A. We further address the main challenges in Persian text.

4.1. Problems with the processing of Persian texts

To construct a processing system and understanding Persian texts, we face several issues and problems, some of which appear in most languages and some are dedicated to Persian language. Some of these complexities are also related to the nature of language and the shortcomings of grammar and others arising from the problems of creating artificial intelligence systems. This section will address some of these issues. Persian language has many differences to English in terms of the structure of

the sentences. In English, the structure of each sentence ia as Subject, Verb, and Object where as in Persian the sentence is formed as Subject, Object, and Verb respectively. In Persian, there are some pronouns connected to nouns and verbs (connected pronouns), which cause different forms to words, which are not in English, and all pronouns are disjunctive. According to the mentioned cases and since Persian language is a form of non-structured languages, there are much more problems than in English. The main problems in the processing of Persian texts can be summarized as follows:

- Lack of adequate linguistic resources for Persian language
- Diagnostic word border problem (issue of different writing methods)
- Diagnostic nominal groups of problem (an invisible additional vowel point issue)
- Ambiguity issue
- Compound verbs and expressions

One of the main steps in the sentiment analysis problems is the preprocessing step which is different from language to language. In the proposed method in this article, we employ several preprocessing steps. One of these steps is the removal of stop words. Stop words are basically a set of commonly used words in any language but they do not have important information [53]. The reason why removing of stop words are critical to many applications is that we may need to focus on the significant words. In this study, we use a list of stop words which has been proposed in [37] and [36]. We further discuss the preprocessing steps in experiment section.

4.2. The proposed MS3A Algorithm

Since, the used dataset in this study is originally unlabeled, we first assign label to one hundred data points out of 13465 examples manually, less than 1% of the data. We use three different classes which are: P for positive, N for negative, and O for neutral comments. Now, the data is ready to learn by the semi-supervised learning algorithm.

As mentioned earlier, we employ one of the recently developed approaches to semi-supervised learning, MSAB, to solve the sentiment analysis problem. However, MSAB employs a radial basis function (RBF) as its similarity measurement which is not suitable for measuring similarity in text data. RBF also requires a tuning parameter. Since, the performance of all similarity-based approaches strongly depend on the used similarity functions. Therefore, finding a suitable similarity function is a challenging task in many similarity-based approaches. In [38], a boosting-based similarity approach is proposed to handle the aforementioned issue. However, this approach may results in a very slow training process and decreases the classification performance of text datasets, because it uses a different versions of a specific similarity function and tunes its weight through the boosting procedure. This approach may not be an effective approach for text data which consists of a high-dimensional feature space, we experimentally show this issue. We hence propose an ensemble approach to handle this issue. Figure 1 gives an overview of the proposed approach.

In the proposed approach, we first give the most effective similarity functions for text data to the algorithm. It then uses MSAB to select and assign label to unlabeled data. Next, the constructed models are combined to build final classification. The final model is then used to assign label to unlabeled data. Algorithm 2 represents the proposed algorithm, named MS3A-Ensemble.

As shown in the algorithm, after several preprocessing steps, MS3A employs N different base classifiers to assign labels to unlabeled examples. The MS3A algorithm first uses N different base learners and similarity functions. Each base learner in MS3A then finds a set of high-confidence

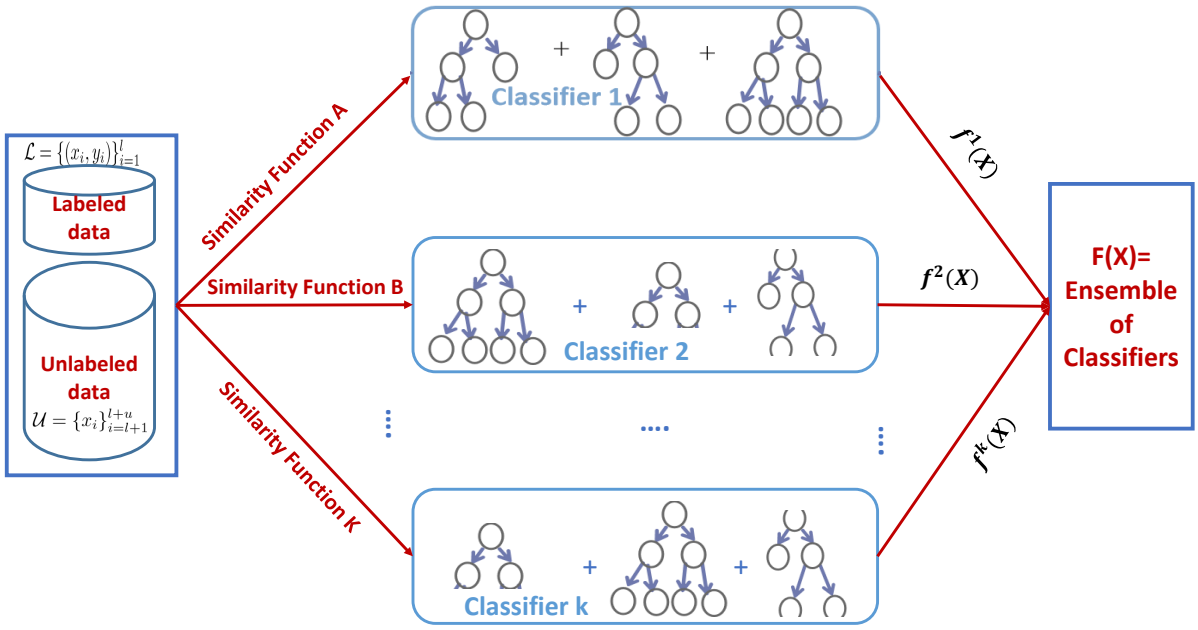

Figure 1: An overview of the proposed algorithm.

predictions from the newly-labeled data at each boosting procedure based on (4.5), see Section 4.4. The built models are next employed to assign label to whole unlabeled data. The MS3A algorithm then finds the agreement among the constructed models to assign final label to unlabeled examples. Now, the whole dataset is labeled and ready to learn by any base learners.

We employ several different learning algorithms to construct the best classification model for the selected sentiment analysis dataset. The experiment section describes the results of the experiments in more details.

Algorithm 2 The Proposed MS3A Algorithm

Initialize: L, U, S^i, F^i L: Labeled data; U: Unlabeled data; F^i: Fully labeled data;
S^i: Similarity Function; H^i:Supervised Base Classifier;
Preprocess the data;
Assign label to a small set of data by experience, human experts;
$i \leftarrow 1$; N: Number of Iterations;
while (each $i \leq N$ **do**)
  Compute $F^i \leftarrow MSAB(L, U, H^i, S^i)$;// Assign pseudo-label to unlabeled data
  $i \leftarrow i + 1$;
end while
$F \leftarrow MajarityVoting_{1 \leq i \leq n} F^i$;// MS3A-Ensemble which is a final Hybrid model
Output: Generate final hypothesis based on F and the selected supervised learner;

As can be seen, the MSAB algorithm is the main part of the proposed algorithm. One of the key component of MSAB is S parameter which is a similarity function. As addressed in [38][40][39], the similarity function plays the main role in the used loss function. Therefore, using a proper similarity measure is vital to build a strong classification model. In the next section, we discuss this parameter and give several new approaches to compute the similarity information in text datasets.

4.3. Pairwise Similarity Measurement Approaches in Text Datasets

One of the key advantages of the MSAB algorithm is a new approach for combining the classifier predictions and the similarity information among labeled and unlabeled data in its loss function. However, finding a suitable similarity measurement is not easy task in many application domains, see [38][40][39]. There are lots of different distance-based approaches to compute the similarity between the data. Most of the current similarity-based semi-supervised learning approaches employ the Radial Basis Function (RBF) as similarity function, see MSAB [41], Mssboost [38], and RegBoost [8]. However, RBF similarity measure consists of some tuning parameter which consumes too much time to find a suitable value. Meanwhile, this tuning process is data-specific. Besides, there is no guarantee that what we find be the optimal parameter value, which makes it hardly comparable to other distance metrics.

In this study, instead of using RBF, we employ several distance metrics to text data which are not widely used in semi-supervised classification problems and these distance-based approaches are the most well-known methods to distance learning. There are also other criteria for similarity measurements, such as TOPSIS and fuzzy-based TOPSIS [51]. However, using these approaches need an adaptation to text dataset.

Normalized Euclidean distance

The most widely used distance metric is the Euclidean distance, $d(p,q) = \sqrt{\Sigma_{i=1}^{m}(p_i - q_i)^2}$ where p and q are m-dimensional vectors. However, features can have significantly different value ranges. Without normalization, it is equivalent to give higher weights to features with higher value. This is not wise especially when the attribute with higher value is not informative. The normalization to the Euclidean distance (NED) can be computed as follows:

$$d(p,q) = \sqrt{\Sigma_{i=1}^{m}(\frac{(p_i - q_i)^2}{s_i^2})} \tag{4.1}$$

where s_i is the standard deviation of p_i and q_i over the sample set.

Cosine Similarity

The cosine similarity criterion is used to find the similarity between two texts based on the similarity of two vectors. If the vectors p and q are normalized, that is, their length will be unit, $cosine(p,q)$ is equal to the internal multiplication of two vectors p and q as follows:

$$p.q = \sum_{i=1}^{N} p_i q_i \tag{4.2}$$

In this case, the cosine similarity between the two vectors is defined as follows:

$$\cos(p,q) = \frac{p.q}{\|p\| \times \|q\|} \tag{4.3}$$

Pairwise-adaptive Similarity

The Pairwise-adaptive similarity measure typically gives the similarity between two documents [12]. This approach dynamically selects a number of features out of p and q and is defined as follows:

$$Pair(p,q) = \frac{p_k.q_k}{(p_k.q_k)^{1/2}(p_k.q_k)^{1/2}}, \tag{4.4}$$

where p_k and q_k are subsets of p and q, containing the values of the features which are the union of the k largest features appearing in p and q respectively.

SMTP Similarity

Recently, a new approach has been presented to compute the similarity between two documents which focuses on the similar features of two documents, called SMTP similarity [22]. In this approach in order to compute the similarity between two documents with respect to a feature, three cases are considered which are: a) the feature appears in both documents, b) the feature appears in only one of the document, and c) the feature appears in none of the documents. In the first case, the similarity increases as the difference between the two involved feature values decreases. Furthermore, the contribution of the difference is normally scaled. For the second case, a fixed value is contributed to the similarity. For the last case, the feature has no contribution to the similarity measurement.

In fact in this approach, the difference between presence and absence of a feature is considered more vital than the difference between the values associated with a present feature. The similarity will be high when the difference between the two values associated with a present feature decreases.

4.4. Metric for Sampling Data

In the sampling process only the high-confidence data points must be selected to use through the training procedure. However, finding the best selection subset is a difficult and challenging task [42]. On one hand, selecting a small set of newly-labeled examples might lead to slow convergence, and on the other hand, selecting a large set of newly-labeled examples may include some poor newly-labeled examples. One possible solution is to use a threshold or even a fixed number which is optimized through the training process.

We employ the following sampling data approach from the newly-labeled data:

$$P_d(x_i) = \frac{\hat{Y}_{i,k} - \max\{Y_{i,k} | Y_{i,k} \neq \hat{Y}_{i,k}, \quad k = 1, ..., K\}}{\sum_{i=1}^{n_u} (\hat{Y}_{i,k} - \max\{Y_{i,k} | Y_{i,k} \neq \hat{Y}_{i,k}, \quad k = 1, ..., K\})} \tag{4.5}$$

where $\hat{Y}_{i,k}$ is the maximum value of $P_{i,k}$. $P_d(x_i)$ is viewed as the probability distribution of classes of the example x_i, which amounts to a measure of confidence. We then select the top 15% of the unlabeled data based on the weights and add them to the training set as a set of high-confidence predictions.

5. Experiments

In the experiment section, we perform several experiments using the proposed algorithm on the sentiment analysis dataset. Since, the used data includes three classes which are positive, negative, and neutral classes, therefore it is a multiclass classification problem. In the experiment, we first assign labels to one hundred data points of dataset manually and the rest of the dataset is kept as unlabeled data. The proposed MS3A algorithm uses several different base learners and similarity functions to assign the most reliable labels to unlabeled data.

In the experiments, we first use only the labeled data and performs several experiments employing several different base learners. We then employ the MS3A algorithm to assign label to unlabeled data and build the final classification model. In this experiment, we also use several different base learners and similarity functions. The used similarity functions include: 1) Normalized Euclidean distance (NED), 2) Cosine Similarity (CS), 3) Pairwise-adaptive Similarity (PDS), and 4) SMTP Similarity (SMTP).

In the experiments, we use the WEKA implementation of the base classifiers with default parameter settings in Java.

5.1. Supervised Base Learner

As mentioned above the MS3A algorithm with several supervised learning algorithms are selected including Naive Bayes, Decision tree (J48, the Java implementation of C4.5 decision tree classifier in default setting), and Multilayer Perceptron (MLP). The performance of the MSAB algorithm depends on the supervised based learner algorithm which is main reason why we propose a hybrid model.

5.2. The used Sentiment Analysis Dataset

In order to examine the algorithm and compare them, it is necessary to use a dataset that is reliable and can test various aspects of the proposed algorithm. The used dataset in this study has been collected from Digikala, a Persian e-commerce website (http://www.digikala.com), one of the most active e-commerce websites in Iran by a crawler. The dataset included 13465 records and it is related to user opinions about the types of phones, including Samsung, Apple, LG, HTC, and Huawei. The data is originally fully unlabeled. We then annotate a small set, one hundred examples, and three different labels are assigned to this small set which are: P for positive, N for negative, and O for neutral comments.

5.3. Experimental setup

For each dataset, we randomly select 30% of the data for testing and the rest as training set in which the labels of only 100 examples are kept. We run each experiment 10 times with different subsets of training and testing data. We report the mean classification accuracy and the standard deviation (std) of 10 times repeating the experiments. The results reported refer to the separate test set.

5.4. Preprocessing Steps

Data Preprocessing is the first step and one of the important steps in text mining. The real data mainly is noisy, incomplete, and inconsistence. Preprocessing may improve the classification performance and the speed of the classification process. We use text cleaning, white space removal, stop words removal, and feature selection to the used Persian sentiment analysis dataset. Stop words are commonly used words involves useless information in any languages [20].

The feature selection has several methods [20]. We employ the term frequency and inverse document frequency (also called tf-idf) in this research. It is a well know method to evaluate how important is a feature in a document. TF-IDF is defined as follows:

$$TF - IDF = FF * \log(\frac{N}{DF}) \tag{5.1}$$

FF is the number of occurrences in the document. N is the total number of documents. DF is the number of documents containing a specific features [20].

6. Results

Tables I, II, and III give the results of all experiments using MS3A and three different base learners on Digikala dataset. In tables I, II, and III, the first column shows the name of dataset. The second column in these tables give the classification performance of supervised multiclass base classifier (SL). We employ Naïve Bayes (NB), Decision Trees (J48), Multilayer Perceptron (MLP) base learners in our experiments. The third column shows the classification performance of MS3A using different similarity functions and the proposed ensemble approach.

Table I:

The classification Performance of MS3A for Digikala dataset when the base learner is J48

	SL	Semi-Supervised Learning				
Datasets	J48	MS3A-NED	MS3A-CS	MS3A-PAS	MS3A-SMTP	MS3A-Ensemble
DigiKala	74.19	80.5	80.65	82.51	82.01	**84.22**

Table I shows the classification performance of all used methods when the base learner is J48. As shown, the proposed method significantly improves the classification performance of the J48 classifier, the improvement is 10%. It is also observed that MS3A-Ensemble outperforms the other methods.

Table II:

The classification Performance of MS3A for Digikala dataset when the base learner is Naive Bayes

	SL	Semi-Supervised Learning				
Datasets	NB	MS3A-NED	MS3A-CS	MS3A-PAS	MS3A-SMTP	MS3A-Ensemble
DigiKala	61.29	64.09	64.52	65.09	**68.45**	68.21

In table II, the results of experiments are shown when the base learner is Naive Bayes. As can be seen, MS3A significantly improves the classification performance of Navie Bayes. We also observe that MS3A and MS3A-Ensemble give the best classification performance.

Table III:

The classification Performance of MS3A for Digikala dataset when the base learner is MLP

	SL	Semi-Supervised Learning				
Datasets	MLP	MS3A-NED	MS3A-CS	MS3A-PAS	MS3A-SMTP	MS3A-Ensemble
DigiKala	74.19	77.09	77.42	79.01	80.56	**82.01**

Table III gives the classification performance of all used methods when the base learner is Multilayer Perceptron (MLP). As shown, MS3A significantly improves the classification performance of the MLP classifier, the improvement is 8%. It is also observed that the proposed method outperforms the other methods.

As these tables show the MS3A-Ensemble algorithm improves the classification performance of supervised learning using several base classifiers. Comparing the results, it is observed that the MS3A-Ensemble algorithm improves the classification performance of the J48 algorithm more than the other base learners. We further observe that the MS3A-Ensemble algorithm gives the best result.

6.1. Using hybrid model in MS3A to assign label

We here assign label to unlabeled data based on the hybrid model in MS3A using a majority voting approach, named MS3A-Ensemble. We now have a fully labeled dataset. We use several different base learners to achieve the best classification performance on the used dataset in this article. We employ NB, Support Vector Machine (SVM), KNN, J48, Logistic Regression Tree, MLP, AdaBoost.M1 (Decision Stump), Bagging (REPtree), LogiBoost (Decision Stump), Bagging (SVM), SVM , Bagging (SVM), and Random Forest as the base learners.

Table IV shows the results. The first column in table IV shows the different classification algorithms applied to the data. Next column gives the classification performance of these algorithms for the used dataset.

Table IV:

The classification Performance of MSAB_OP for Digikala dataset

Base Learner	Classification Accuracy
NB	86.98
SVM	95.22
KNN	94.95
J48	**99.38**
Logistic Regression Tree	**99.38**
MLP	99.28
AdaBoost(DS)	85.05
Bagging(REPtree)	**99.46**
LogiBoost(DS)	99.31
Bagging(SVM)	95.39
SVM	95.22
Random Forest	99.19

As shown in Table IV, J48, Logistic Regression Tree, and Bagging with REPTree as the base learner give the best classification performance. The achieved classification accuracy is 99.46 with Bagging ensemble learner to the used sentiment analysis dataset.

Comparing the results of Tables I, II, and III to IV, we observe that the MS3A algorithm improves the classification performance 25.47% when the base learner is J48. It is also observed that the MS3A algorithm improves the classification performance of NB and MLP base learners significantly.

7. Conclusion and Discussion

In this article, an algorithm for sentiment analysis is presented to sentiment analysis of the Persian commercial website. We propose a hybrid model based on MSAB and MSSBoost using different similarity functions. The proposed approach uses the MSAB algorithm to label the unlabeled data. The proposed approach employs the classifier predictions along with the similarity information to assign label to unlabeled data. We employ a hybrid model to combine different models in order to improve the classification performance. The resulting algorithm improves the performance of supervised learning with different base classifiers for the used dataset.

It is observed that MS3A-Ensemble improves the performance of J48 more than Naive Bayes and Artificial Neural Network. The algorithm then uses the several classification algorithms to assign final label to unlabeled data. We further observe that the similarity information plays the main role. We finally show that the proposed algorithm significantly improves the classification performance and analyzes the comments properly.

References

[1] U. Aggarwal and G. Aggarwal. *Sentiment analysis: A survey.* Int. J. Comput. Sci. Engin. 5(5), (2017) 222–225.
[2] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. *A brief survey of text mining: Classification, clustering and extraction techniques.* arXiv preprint arXiv:1707.02919, 2017.
[3] J. A. Balazs and J. D. Velásquez. *Opinion mining and information fusion: a survey.* Inf. Fus. 27, (2016) 95–110.
[4] M. Belkin, P. Niyogi, and V. Sindhwani. *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.* J. Machine Learn. Res. 7, (2006) 2399–2434.
[5] S. Bhatia, M. Sharma, and K. K. Bhatia. Sentiment analysis and mining of opinions. 503–523. Springer, 2018.

[6] P. Biyani, C. Caragea, P. Mitra, C. Zhou, J. Yen, G. E. Greer, and K. Portier. *Co-training over domain-independent and domain-dependent features for sentiment analysis of an online cancer support community.* Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, 2013, 413–417.

[7] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. Eleventh Annual Conf. Comput. Learn. Theory*, (1998) 92–100.

[8] K. Chen and S. Wang. *Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions.* IEEE Trans. Pattern Anal. Machine Intel. 33(1), (2011) 129–143.

[9] K. Crammer, A. Kulesza, and M. Dredze. *Adaptive regularization of weight vectors.* Adv. Neural Inf. Proc. Syst. 2009, (2009) 414–422.

[10] D. Davidov, O. Tsur, and A. Rappoport. *Enhanced sentiment learning using twitter hashtags and smileys.* Proc. 23rd Int. Conf. Comput. Ling. Posters, 2010, 241–249.

[11] W. C. Dhaoui, C. and L. Tan. *Social media sentiment analysis: lexicon versus machine learning.* J. Consumer Market. 39(6), (2017) 480–488.

[12] J. D'hondt, J. Vertommen, P. Verhaegen, D. Cattrysse, and J. R. Duflou. *Pairwise-adaptive dissimilarity measure for document clustering.* Inf. Sci. 180(12), (2010) 2341–2358.

[13] Y. Han, Y. Liu, and Z. Jin. *Sentiment analysis via semi-supervised learning: a model based on dynamic threshold and multi-classifiers.* Neural Computing and Applications, 32, 2020, 5117–5129.

[14] S. Hong, J. Lee, and J.-H. Lee. *Competitive self-training technique for sentiment analysis in mass social media.* Soft Comput. Intel. Syst. 2014 Joint 7th Int. Conf. Adv. Intel. Syst. 15th Int. Symp. 2014, (2014) 9–12.

[15] M. Hu and B. Liu. *Mining and summarizing customer reviews.* Proc. Tenth ACM SIGKDD Int. Conf. Knowledge Disc. Data Min., (2004) 168–177.

[16] X. Hu, L. Tang, J. Tang, and H. Liu. *Exploiting social relations for sentiment analysis in microblogging.* Proc. Sixth ACM Int. Conf. Web Search Data Min., 2013, 537–546.

[17] S. Inzalkar and J. Sharma. *A survey on text mining-techniques and application.* Int. J. Res. Sci. Engin. 24, (2015) 1–14.

[18] F. H. Khan, U. Qamar, and S. Bashir. *A semi-supervised approach to sentiment analysis using revised sentiment strength based on sentiwordnet.* Knowl. Inf. Syst. 51(3), (2017) 851–872.

[19] S. Kumar, K. De, and P. P. Roy. *Movie recommendation system using sentiment analysis from microblogging data.* IEEE Trans. Comput. Soc. Syst. 2020, (2020) 1–9.

[20] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili. *A novel multivariate filter method for feature selection in text classification problems.* Engin. Appl. Artif. Intel. 70, (2018) 25–37.

[21] Z. Li, C. Li, L. Yang, P. S. Yu, and Z. Li. *Mixture distribution modeling for scalable graph-based semi-supervised learning.* Knowledge-Based Syst. 200, (2020) 105974.

[22] Y. Lin, J. Jiang, and S. Lee. *A similarity measure for text classification and clustering.* IEEE Trans. Knowledge Data Engin. 26(7), (2014) 1575–1590.

[23] S. Liu, W. Zhu, N. Xu, F. Li, X.-q. Cheng, Y. Liu, and Y. Wang. *Co-training and visualizing sentiment evolvement for tweet events.* Proc. 22nd Int. Conf. World Wide Web, 2013, 105–106.

[24] W. Liu, X. Jing, Y. Chen, and J. Li. *Co-training based on multi-type text features.* Int. Conf. Signal Inf. Proc. Network. Comput., 2017, 213–220.

[25] Z. Liu, X. Dong, Y. Guan, and J. Yang. *Reserved self-training: A semi-supervised sentiment classification method for chinese microblogs.* Proc. Sixth Int. Joint Conf. Natural Lang. Proc., 2013, 455–462.

[26] Z. Miao, Y. Li, X. Wang, and W. Tan. *Snippext: Semi-supervised opinion mining with augmented data.* CoRR, abs/2002.03049, 2020.

[27] S. M. Mohammad, S. Kiritchenko, and X. Zhu. *Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets.* arXiv preprint arXiv:1308.6242, 2013.

[28] A. Pak and P. Paroubek. *Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives.* Proc. 5th Int. Workshop Semantic Ev., 2010, 436–439.

[29] S. Park, J. Lee, and K. Kim. *Semi-supervised distributed representations of documents for sentiment analysis.* Neural Networks, 119, (2019) 139–150.

[30] L. Qiu, W. Zhang, C. Hu, and K. Zhao. *Selc: a self-supervised model for sentiment classification.* Proc. 18th ACM conf. Inf. Knowledge Manag., 2009, 929–936.

[31] J. Read and J. Carroll. *Weakly supervised techniques for domain-independent sentiment classification.* Proc. 1st Int. CIKM Workshop Topic-sentiment Anal. Mass Opin., 2009, 45–52.

[32] H. Saif, T. Dickinson, L. Kastler, M. Fernandez, and H. Alani. *A semantic graph-based approach for radicalisation detection on social media.* Euro. Semantic web Conf., 2017, 571–587.

[33] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma. *Sentiment analysis: A review and comparative analysis of web services.* Inf. Sci. 311, (2015) 18–38.
[34] N. F. F. Silva, L. F. Coletta, E. R. Hruschka, and E. R. Hruschka Jr. *Using unsupervised information to improve semi-supervised tweet sentiment classification.* Inf. Sci. 355, (2016) 348–365.
[35] N. F. F. D. Silva, L. F. Coletta, and E. R. Hruschka. *A survey and comparative study of tweet sentiment analysis via semi-supervised learning.* ACM Computing Surveys, 49(1), (2016) 1–15.
[36] K. Taghva, R. Beckley, and M. Sadeh. *A list of farsi stopwords.* Ret. Sept. 2003(7), (2003).
[37] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. *User-level sentiment analysis incorporating social networks.* Proc. 17th ACM SIGKDD Int. Conf. Knowledge Disc. Data Min., 2011, 1397–1405.
[38] J. Tanha. *Mssboost: A new multiclass boosting to semi-supervised learning.* Neurocomput. 2018, (2018).
[39] J. Tanha. *A multiclass boosting algorithm to labeled and unlabeled data.* Int. J. Machine Learn. Cyber. 2019, (2019).
[40] J. Tanha, M. J. Saberian, and M. Van Someren. *Multiclass semi-supervised boosting using similarity learning.* Data Mining (ICDM), 2013 IEEE 13th Int. Conf., 2013, 1205–1210.
[41] J. Tanha, M. Van Someren, and H. Afsarmanesh. *Boosting for multiclass semi-supervised learning.* Pattern Recog. Let. 37, (2014) 63–77.
[42] J. Tanha, M. van Someren, and H. Afsarmanesh. *Semi-supervised self-training for decision tree classifiers.* Int. J. Machine Learn. Cyber. 8(1), (2017) 355–370.
[43] H. Thakkar and D. Patel. *Approaches for sentiment analysis on twitter: A state-of-art study.* arXiv preprint arXiv:1512.01043, 2015.
[44] A. Tripathy, A. Agrawal, and S. K. Rath. *Classification of sentiment reviews using n-gram machine learning approach.* Expert Syst. Appl. 57, (2016) 117–126.
[45] H. Valizadegan, R. Jin, and A. K. Jain. *Semi-supervised boosting for multi-class classification.* Joint Euro. Conf. Machine Learn. Knowledge Disc. Datab. 2008, (2008) 522–537.
[46] B. Xiang and L. Zhou. *Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training.* Proc. 52nd Annual Meet. Assoc. Comput. Ling. 2, (2014) 434–439.
[47] W. Xu and Y. Tan. *Semi-supervised target-oriented sentiment classification.* Neurocomput. 337, (2019) 120–128.
[48] N. Yu. *Exploring c o-training strategies for opinion detection.* J. Assoc. Inf. Sci. Tech. 56(10), (2014) 2098–2110.
[49] N. Yu and S. Kubler. *Semi-supervised learning for opinion detection.* Web Intel. Intel. Agent Tech. (WI-IAT), 2010 IEEE/WIC/ACM International Conf. 3, (2010) 249–252.
[50] T. Zagibalov and J. Carroll. *Unsupervised classification of sentiment and objectivity in chinese text.* Proc. Third Int. Joint Conf. Natural Lang. Proc. Volume-I, 2008.
[51] S. Zeng, D. Luo, C. Zhang, and X. Li. *A Correlation-Based TOPSIS Method for Multiple Attribute Decision Making with Single-Valued Neutrosophic Information.* Int. J. Inf. Tech. Dec. Mak. 19(1), ().
[52] J. Zhao, M. Lan, and T. Zhu. *Ecnu: Expression-and message-level sentiment orientation classification in twitter using multiple effective features.* Proc. 8th Intm Workshop Semantic Ev. 2014, (2014) 259–264.
[53] F. Zou, F. L. Wang, X. Deng, and S. Han. *Automatic identification of chinese stop words.* Res. Comput. Sci. 18, (2006) 151–162.