Applied Intelligence (2025) 55:313
https://doi.org/10.1007/s10489-024-06071-z

SSSA: low data sentiment analysis using boosting semi-supervised approach and deep feature learning network

Shima Rashidi[1] · Jafar Tanha[2] · Arash Sharifi[1] · Mehdi Hosseinzadeh[3,4]

Accepted: 13 November 2024 / Published online: 14 January 2025

Abstract
Sentiment analysis is the process of determining the expressive direction of the user reviews. Recently, sentiment analysis gets more attention. However, low data sentiment analysis receives less attention. The existing works try to augment the samples to consider this issue. In this study, we have utilized a semi-supervised approach to propose a new approach for low-data sentiment analysis. To do so, we have utilized pre-trained XLNet as a feature extractor network to initialize the feature vector for each tweet. Next, these initial representations are fed into the embedding update module to map features into the new space by optimizing the contrastive loss. Then, we utilized a semi-supervised boosting method to assign pseudo labels to unlabeled data. The iteration between the semi-supervised module and the embedding update module is done until convergence is happened. During these iterations, the embedding update module propagates the error-correcting signals to a semi-supervised module. To evaluate the proposed approach, we have applied it to the SemEval2017dataset (task 4), Sentiment 140, and IMDB Movie Reviews. We have designed many different experiment settings to validate the proposed approach's different modules. On SemEval2017dataset (task 4), we have got 75.9% and 77.1% in AvgRec and F_1^{PN} respectively. Also, when only 10% of the training samples as labeled samples are used, we get the 71.8% and 73.6% in AvgRec and F_1^{PN} respectively. The results show that our approach significantly improves with respect to the comparable methods. Also, on IMDB Movie Reviews and Sentiment 140, the proposed approach demonstrates improved performance compared to comparable methods.

Keywords Sentiment analysis · Low data sentiment analysis · Semi-supervised · Embedding update mechanism · Pseudo labeling approach

1 Introduction

Twitter sentiment detectors (TSDs) are systems that evaluate the quality of services and products. It is shown that it entirely depends on the quality of the provided input features. Sentiment analysis is a promising technique in several real-world applications like recommender systems. One of the most important categories of the traditional approaches in sentiment analysis is dictionary-based methods. These methods utilize lexicons and traditional machine-learning algorithms. Until now, several sentiment lexical resources for the English language are developed [1].

In recent years, deep learning has attracted great attention in sentiment analysis. Deep learning-based approaches commonly include two min sub-networks: feature representation and task prediction subnetworks. A feature representation network provides a representation of raw data such as texts or tweets. These representations are fed into the task predictor to provide a sentiment for the input text. The learned model is used to infer knowledge from unseen data. In some cases, sentiment analysis

✉ Shima Rashidi
Shimarashidi6@gmail.com

✉ Jafar Tanha
tanha@tabrizu.ac.ir

Arash Sharifi
arash.sharifi@gmail.com

Mehdi Hosseinzadeh
mehdihosseinzadeh@duytan.edu.vn

1 Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

2 Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

3 School of Computer Science, Duy Tan University, Da Nang, Vietnam

4 Jadara University Research Center, Jadara University, Irbid, Jordan

suffers from a lack of data. The reason is that humans should annotate data, and it is costly and time-consuming. In recent years, sentiment analysis in the presence of low data gets low attention. Gupta [2] utilizes a generative adversarial network (GAN) [3] as a method of data augmentation to learn a more powerful model. Meetei et al. [4] introduced a sentiment analysis approach for a language with low resources. To do so, they used some preprocessing techniques to generate additional linguistic features and then utilized those features to learn a more discriminative model. Yu and Zhang [5] introduced an approach for multi-modal sentiment analysis (MSA) in few-shot scenarios (i.e., low data) [6]. This paper proposes a new low-data sentiment analysis approach that utilizes semi-supervised methods to learn a model with better generalization ability.

The proposed approach uses the deep learning-based model to learn more discriminative feature space and utilize boosting for multi-class semi-supervised learning [5]. In recent years, semi-supervised learning (SSL) has attracted the most interest because it could utilize the knowledge of a large amount of unlabeled data [7, 8]. Pseudo-labeling is a powerful technique in SSL, which assigns a pseudo-label to some unlabeled data and then utilizes those samples to learn a more powerful model [9–13]. The proposed method relies on pseudo-labeling. One of the reasons that we have selected boosting for multi-class semi-supervised learning [5] is that it not only defines a loss function on labeled data but also defines two regularization terms on labeled and unlabeled data. These two regularization terms rely on two matrices: 1) similarity matrix of the labeled data to the unlabeled data, and 2) similarity matrix of the unlabeled data. Hence, the feature vectors of tweets and similarity measures play an important role. In this work, we emphasize feature representation learning. The proposed approach feeds the raw tweets into one of the pre-trained language models (like XLNet). To fine-tune these representations with our custom data, we design an embedding update network that embeds the generated feature vector by language model into a new feature space in which similar samples are near together and the dissimilar samples have a larger distance. To satisfy this property, we design a deep neural network with a triplet loss function (by utilizing labeled data). The triplet loss function helps the model learn more discriminative feature representation, which leads to a more accurate similarity matrix creation. Next, by using a boosting semi-supervised algorithm, we assign a pseudo-label to some of the unlabeled data with high confidence, and then after some epochs, this process is repeated until convergence is happened. It is applied to SemEval 2017 dataset task 4 to evaluate the proposed approach. Different experimental settings are designed to show the effectiveness of the proposed approach.

To recap, the differences between the proposed model and others are as follows:

1) It is designed for low-labeled data scenarios with a large amount of unlabeled data.
2) It utilizes a semi-supervised algorithm to assign pseudo-label to some of the unlabeled data with high confidence.
3) It utilizes the triplet loss function to learn the update embedding module, which led to a more accurate similarity matrix between labeled and unlabeled data.
4) It utilizes an iterative process that iterates between the update embedding and semi-supervised modules.

This paper is organized as follows: in Sect. 2, the details of the proposed approach are given. Section 3 applies the proposed approach to the well-known datasets and analyzes the results. Finally, in Sect. 4, the obtained results are discussed.

2 Related work

Sentiment analysis gets more attention in the last few years, and many approaches are given. Based on the task definitions, the sentiment analysis approaches are divided into three categories: document-level sentiment classification, sentence-level sentiment classification, and aspect-level sentiment classification [14]. Our approach is sentence-level sentiment analysis. In recent years, multi-modal and multi-domain sentiment analysis has received more attention [14]. In multi-modal SA, the input is the text tweets along with information in another medium like video or image. In multi-domain SA, the goal is to transfer knowledge between two different domains. In recent years, sentiment analysis using deep learning has received growing interest. For sentence-level SA, RNN and LSTM are utilized [14]. After introducing transformers, sentiment analysis is significantly influenced by the growth of transformers' capabilities in NLP applications [15]. In the following, we focus on the approaches that have utilized transformers and unlabeled data.

Wu et al. [16] introduced an approach that utilizes the information from unlabeled data. Their approach has three steps: 1) the first step is to learn a hidden representation for input, 2) the second step is to learn a sentiment prediction model, and 3) the third step is a decoding module that takes the hidden representation and the predicted sentiment as input and reconstruct the input text as output. Tesfagergish et al. [17] introduced an approach that includes two steps. The first step is an unsupervised step based on zero-shot learning, which detects the probabilities for subsets of 34 emotions. The second step is a supervised step that utilizes ensemble learning, which takes the output of the first step as input. Tiwari and Nagpal [18] introduced a Knowledge-Enriched Attention-based Hybrid Transformer (KEAHT) model by utilizing a Latent Dirichlet Allocation (LDA) [19] topic modeling approach. They have utilized a pre-trained Bidirectional Encoder Representation from Transformer (BERT) to learn representation for input texts. Almalis et al. [20] introduced a deep learning-based approach for sentiment analysis, trained on financial news, that is further enhanced via a semi-supervised learning method. It relies on detecting and relabeling news that has most likely been mislabeled as neutral. Also, some comparative studies review and compare deep learning approaches in sentiment analysis [14, 21]. Chen & Qian [22] introduced a new domain adaptation technique for aspect-level sentiment classification to improve the transferability of domain-specific words. Singh et al. [23] introduced an approach for sentiment analysis of Twitter data related to COVID-19 reviews. Their approach is based on LSTM-RNN, which utilizes an attention mechanism to learn more discriminative features. Wu and Shi [24] introduced an Adversarial Soft Prompt Tuning method (AdSPT) in which cross-domain sentiment analysis is modeled. They have utilized the adversarial domain adaptation technique to transfer knowledge between the source domain and the target domain to learn domain-invariant representations. Tan et al. [25] proposed a hybrid model called RoBERTa-LSTM which utilize transformes and LSTM simultenaously. Mathew and Bindu [26] introduced an efficient transformer-based sentiment classification (ETSC) model which utilize simple learning model with low training time. Zhang et al. [27] introduced a broad multitask transformer network called BMT-Net.

Chen et al. [28] introduced a paradigm for semi-supervised learning which includes three steps: first, unsupervised pretraining of a big (deep and wide) neural network is done using SimCLRv2, then, fine-tuning of the model is done using a few labeled examples, and finally refining and transferring the task-specific knowledge go through the distillation with unlabeled examples. Zheng et al. [29] introduced a semi-supervised learning algorithm called SimMatch. To do so, they have defined two similarity levels: the semantic level and the instance level. Their method encourages different views of the same sample to belong to the same class label. To bridge the gap between these two different similarity levels, they have introduced a new aggregation method. Sohn et al. [30] introduced FixMatch, which first generates weakly augmented unlabeled samples and trains the model to assign pseudo-labels to the unlabeled samples. Then, for highly confidential pseudo-labeled samples, they fine-tune the model with strongly augmented samples. In this case, it is expected that the model assigns the same labels to those samples.

3 The proposed approach

In this section, the proposed approach is given. The overall schematic of the proposed approach is given in Fig. 1. As it is shown, our approach includes three main steps:

1) Document feature extractor: Extract the feature representation of the input document.
2) Pseudo labeling of unlabeled data: Utilize a semi-supervised learning approach.
3) Update embedding module: Learn a projection matrix to map the sample to the new representation space.

The problem formulation is first given in the following, and then each step of the proposed approach is explained.

3.1 Problem formulation

Given $\{\left(x_i^l, c_i^l\right)_{i=1}^{n^l}, (x_i^u)_{i=1}^{n^u}\}$ denote the whole data where $\left(x_i^l, c_i^l\right)$ shows the i^{th} labeled sample include texts or tweets (x_i^l) and its corresponding label (c_i^l). The class label shows the emotional tone of the text. It includes three states: positive, negative, and neutral. Also, (x_i^u) shows the i^{th} unlabeled data. As it is shown, there is no corresponding class label with the unlabeled data. The goal is to learn a function that gets the text (a tweet) as input and then maps it into the label space, including positive, negative, and neutral. The contribution of this paper is to propose a new approach that utilizes the unlabeled data along with the labeled ones. As shown in Fig. 1, the proposed approach is an iterative process between the update embedding module (Fig. 1-B) and the semi-supervised module (Fig. 1-C).

3.2 Feature extractor

The goal of this subnetwork is to extract features from the input text. In text analysis, word embedding is one of the most important steps, and its goal is to learn a vector representation for a particular word. In recent years, many successful networks for this purpose are introduced, including word2vec [31], Glove [32], BERT [33], and XLNet [34]. In this study, we have utilized all of these networks, and a comparison is done between them. The results show that XLNet performed better than the others. In recent years, ChatGPT has received more attention. ChatGPT and XLNet have been trained on large amounts of unlabeled data using deep learning techniques, but they differ significantly in architecture and training objectives. One of the main differences that led us to utilize XLNet is the task they are learning for. ChatGPT is designed for conversational AI, while XLNet is designed for tasks like sentiment analysis.

Word2vec is one of the most common approaches that was introduced by Mikolov et al. [31]. It has utilized a shallow neural network. GloVe (Global Vectors for Word Representation) is another popular unsupervised word embedding algorithm. GloVe, in comparison with Word2Vec, performs competitively with minimal computational overheads. GloVe minimizes the difference between observed co-occurrences and predicted values, leading to meaningful vector representations encoding linguistic properties. It should be noted that Glove utilizes global word-to-word co-occurrence counts through the entire corpus, while Word2vec utilizes co-occurrence within the local context. BERT is inspired by human cognitive behavior when reading sentences. It utilizes context information by adding special tokens around text inputs, whether the sentence is positive or negative. XLNet is a generalized autoregressive method that utilizes a bidirectional transformer.

In the proposed approach, regardless of which word embedding algorithm is used, the designed feature extractor network is shown by N_E. This network takes the raw tweet as input and then produces a feature extractor as output.

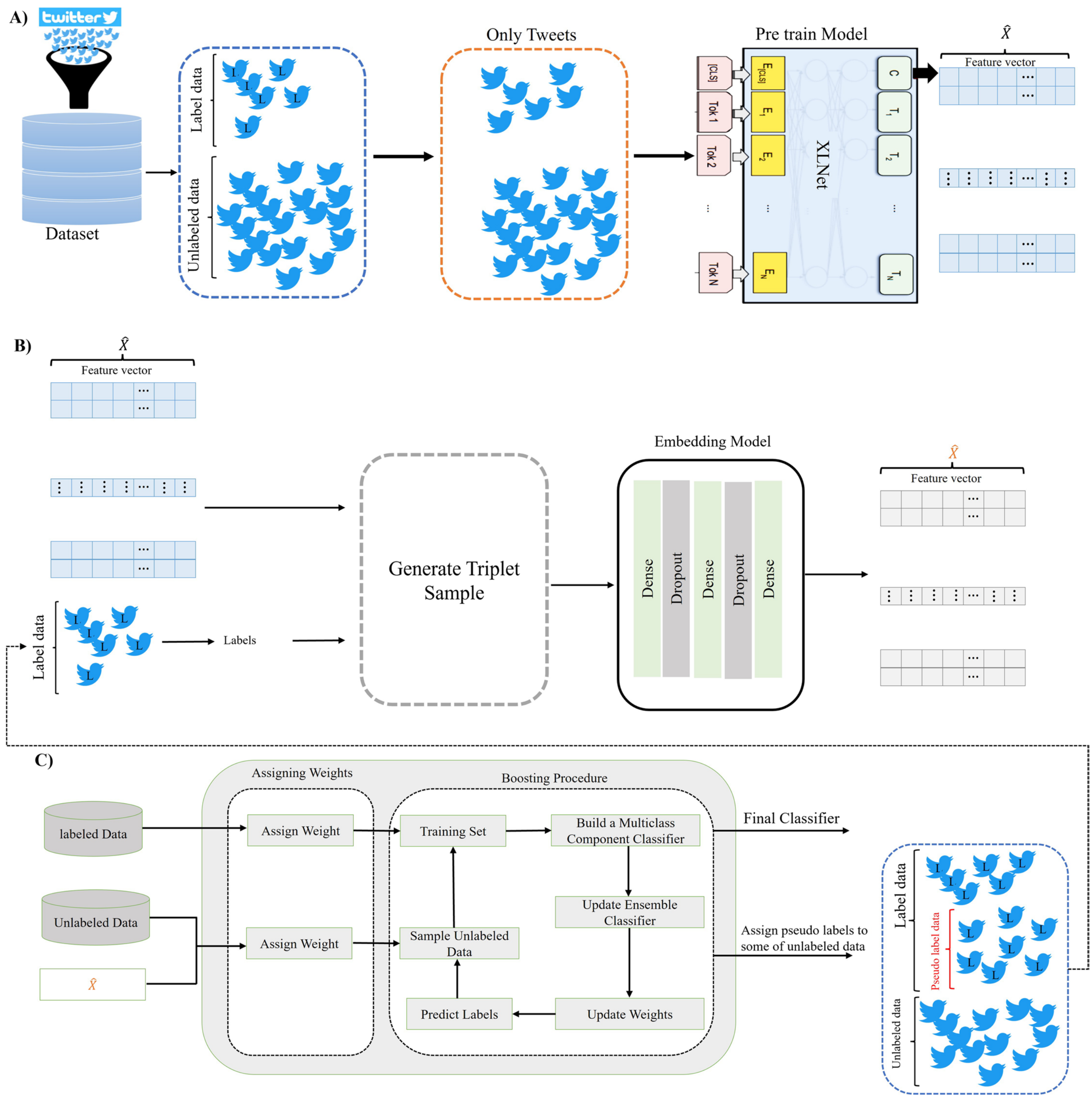

Fig. 1 The overall schematic of the proposed approach. **A** learn an initial feature vector for raw tweets using the pre-trained language model like XLNet. **B** Fine-tune the representation of the labeled data using the triplet loss function. **C** apply the semi-supervised approach to assign a pseudo label to some unlabeled data. The proposed approach is an iterative approach in which the pseudo-labeled data are fed to step (**B**), and then this procedure is repeated until convergence has happened

3.3 Pseudo labeling

This step explains the semi-supervised algorithm we have utilized to pseudo-label the samples and select the most confident ones. There are many semi-supervised learning algorithms. We have utilized a multi-class semi-supervised boosting algorithm (MSAB) [11]. MSAB takes labeled and unlabeled data, a similarity matrix (shown by S) and an ensemble multiclass classifier (shown by H) as input and learn a classifier model as output. To this end, the following function is considered as a loss function:

$$F(Y,S,H) = C_1F_1(Y,H) + C_2F_{lu}\left(Y,S^{lu},H\right) + C_3F_{uu}(S^{uu},H) \quad (1)$$

where C_1, C_2 and C_3 show the importance and weight of each term. The first term is the loss function for labeled data. Similar to [11], this loss function is defined as follows:

$$F_1(Y,H)=\sum_{i=1}^{n_l}\exp(-\frac{1}{K}<H(\widetilde{x}_i),y_i>) \tag{2}$$

where $H(x_i)$ is the predicted output for x_i and K is the number of classes. The second term of Eq. (1) considers the consistency between the labeled and unlabeled data. It states that similar unlabeled and labeled samples should have the same label. However, the unlabeled sample does not have any corresponding label. As a result, for similar pairs of unlabeled and labeled ones, the predicted label for the unlabeled sample should be the same as the class of labeled sample. This term is defined as follows:

$$F_2(Y,H)=\sum_{i=1}^{n_l}\sum_{i=1}^{n_u}S^{lu}(\widetilde{x}_i,\widetilde{x}_j)\exp(-\frac{1}{K-1}<H(\widetilde{x}_j),y_i>) \tag{3}$$

where S^{lu} is a similarity matrix that computes the similarity between the i^{th} labeled tweet and the j^{th} unlabeled tweet.

The third term of Eq. (1) considers the consistency assumption of all unlabeled samples. It states that the (predicted) class labels of the nearby unlabeled samples on the same manifold are likely to be the same. Hence, it is defined as follows:

$$F_3(Y,H)=\sum_{i=1}^{n_u}\sum_{i=1}^{n_u}S^{uu}(\widetilde{x}_i,\widetilde{x}_j)\exp(-\frac{1}{K-1}<H(\widetilde{x}_i)-H(\widetilde{x}_j),\vec{1}>) \tag{4}$$

where S^{uu} is a similarity matrix that computes the similarity between the i^{th} unlabeled tweet and j^{th} unlabeled tweet. It should be noted that $<H(\widetilde{x}_i)-H(\widetilde{x}_j),\vec{1}>$ tends to be zero if the predicted class for the i^{th} unlabeled tweet ($H(\widetilde{x}_i)$) and predicted class for the j^{th} unlabeled tweet $H(\widetilde{x}_j)$ to be different. Also, if the predicted classes for both samples are the same, then $<H(\widetilde{x}_i)-H(\widetilde{x}_j),\vec{1}>$ tends to be one.

By putting Eqs. 2, 3, and 4 in Eq. (1), the following formula is obtained:

$$\begin{aligned}F(Y,S,H)=C_1\sum_{i=1}^{n_l}\exp(-\frac{1}{K}<H(\widetilde{x}_i),y_i)+C_2\sum_{i=1}^{n_l}\sum_{i=1}^{n_u}S^{lu}(\widetilde{x}_i,\widetilde{x}_j)\\ \exp(-\frac{1}{K-1}<H(\widetilde{x}_j),y_i)+C_3\sum_{i=1}^{n_u}\sum_{i=1}^{n_u}S^{uu}(\widetilde{x}_i,\widetilde{x}_j)\\ \exp(-\frac{1}{K-1}<H(\widetilde{x}_i)-H(\widetilde{x}_j),\vec{1})\end{aligned} \tag{5}$$

As is shown in this formula, S^{lu} and S^{uu} play an important role because they are responsible for effectively incorporating unlabeled data in sentiment analysis.

We have utilized the following procedure in the proposed approach to define these two similarity matrices. In this procedure, we should utilize the word embedding network (N_E). To do so, these matrices (S^{lu} and S^{uu}) are defined as follows:

$$S^{lu}=\exp(-\frac{\|N_E(\widetilde{x}_i^l)-N_E(\widetilde{x}_j^u)\|_2^2}{\delta^2}) \tag{6}$$

$$S^{uu}=\exp(-\frac{\|N_E(\widetilde{x}_i^u)-N_E(\widetilde{x}_j^u)\|_2^2}{\delta^2}) \tag{7}$$

where δ^2 plays an important role in similarity matrix computation. In this approach, like [35] and [11], we have defined it as 15 percent of the range of the distance between examples.

3.4 Embedding update mechanism

In this step, tweet embedding is learned again with the new confident pseudo-labeled samples and all labeled samples. To this end, we should learn a new projection matrix that could help the entire model to learn a stronger sentiment analysis model. As it is mentioned, the similarity matrices, including S^{lu} and S^{uu}, play an important role in optimizing Eq. 5. As it is clear, the feature vector is pivotal in creating these matrices. Therefore, this step aims to learn more discriminative feature vectors for each sample. We have utilized the triplet loss function to learn a better representation embedding for whole tweets (including labeled and unlabeled samples). Triplet loss is introduced by Schroff et al. [36], and its goal is to minimize the distance between the anchor and positive samples representation and maximize the distance between the anchor and negative samples representation. As it is clear, in this case, we need three samples: anchor, positive and negative. We have utilized the following algorithm in the proposed approach to select these pairs. Given $n^{(i)}$ shows the number of samples in i^{th} class label (number of the new confident pseudo-labeled samples and all labeled samples). First, for class label i, we extract all labeled samples and then extract $2\,n_i^{half}$ different pairs randomly without replacement, where $n_i^{half}=floor(\frac{n^{(i)}}{2})$. These extracted samples are considered anchor and positive samples. Next, for extracting the negative samples, we randomly select n_i^{half} samples from the samples that do not belong to the i^{th} class label. It should be noted that negative samples are drawn with replacement. This procedure is repeated for all class labels. In this case, each extracted data contains anchor, positive, and negative samples. All extracted triplet data is shown by $\{(\widetilde{x}_i^{anchor},\widetilde{x}_i^{positive},\widetilde{x}_i^{negative})\}_{i=1}^{N^{triplet}}$ where $N^{triplet}$ is the number of extracted triplet samples. Then, the triplet loss function is defined as follows

$$l_{triplet}\left(\{(\widetilde{x}_i^{anchor},\widetilde{x}_i^{positive},\widetilde{x}_i^{negative})\}_{i=1}^{N^{triplet}}\right)=\sum_{i=1}^{N^{triplet}}\max(\|N_u(\widetilde{x}_i^{anchor})-N_u(\widetilde{x}_i^{positive})\|_2-\|N_u(\widetilde{x}_i^{anchor})-N_u(\widetilde{x}_i^{negative})\|_2+\alpha,0) \tag{8}$$

The triplet loss is minimized by adjusting the parameters of the neural network (N_u). This network (N_u) is an MLP network with the following architecture: dense layer, batch normalization layer, dense layer, batch normalization layer, and dense layer. After optimizing the triplet loss function, the feature vector of the labeled samples and unlabeled samples are updated as follows:

$$\tilde{x}_i^{anchor} = N_u\left(\tilde{x}_i^{anchor}\right) \tag{9}$$

$$\tilde{x}_i^{positive} = N_u\left(\tilde{x}_i^{positive}\right) \tag{10}$$

$$\tilde{x}_i^{negative} = N_u\left(\tilde{x}_i^{negative}\right) \tag{11}$$

Next, these updated feature representations are fed into the semi-supervised module. This process is repeated until the convergence has happened.

The pseudo-code of the proposed approach (SSSA) is given in Algorithm 1. As it is shown, the proposed approach has two loops. In each iteration of the semi-supervised algorithm (the inner loop), the embedding update mechanism should be done to learn more discriminative feature vectors for samples (the outer loop).

Algorithm 1 The pseudo-code of the SSSA algorithm

Input: $\{(x_i^l,\ c_i^l)_{i=1}^{n^l}, (x_i^u)_{i=1}^{n^u}\}$
$\boldsymbol{M}$**:** the number of maximum iterations to learn the feature extraction network
$\boldsymbol{M_1}$**:** the number of maximum iterations to learn the semisupervised boosting algorithm
Output: A classifier model

- Fine-tune N_E using $\{(x_i^l)_{i=1}^{n^l}, (x_i^u)_{i=1}^{n^u}\}$
- Extract feature vector for all samples using the learned model (N_E) which are shown by $\{(\tilde{x}_i^l)_{i=1}^{n^l}, (\tilde{x}_i^u)_{i=1}^{n^u}\}$
- $t \leftarrow 0$

While $\beta^t > 0$ and $t < M$ **do**
  ○ Extract the triplet samples using the labeled samples, which are denoted by $\{(\tilde{x}_i^{anchor}, \tilde{x}_i^{positive}, \tilde{x}_i^{negative})\}_{i=1}^{N^{triplet}}$
  ○ Train N_u using the extracted triplet samples
  ○ Update the feature vector of the sample using Eq. 9-11
  ○ $t_1 \leftarrow 0$
  While $\beta^t > 0$ and $t_1 < M_1$ **do**
    ▪ Compute the similarity matrix between labeled and unlabeled samples S^{lu} using Eq. 6
    ▪ Compute the similarity matrix between unlabeled samples S^{uu} using Eq. 7
    ▪ Assign pseudo-labels to unlabeled examples by optimizing Eq. 5
    ▪ Sample a set of high-confidence examples from labeled and unlabeled examples
    ▪ Build a new weighted classifier *H* based on the newly-labeled and original labeled examples
    ▪ $t_1 \leftarrow t_1 + 1$
  end
  ○ $t \leftarrow t + 1$
end
Output: Generate final model

4 Experiments

In this section, the proposed approach is evaluated. To do so, we applied it to the SemEval2017 dataset. SEMEVAL is an evaluation campaign organized by the Special Interest Group (SIG) on Semantic Evaluation and Applications (SEA) within the Association for Computational Linguistics (ACL). Its main purpose is to provide public evaluations on various NLP subtasks, including sentiment analysis.

4.1 Experimental settings

The semEval-2017 dataset is retrieved from the Twitter social network, and it contains symbols of emojis and weblinks that are filtered out in the preprocessing step. SemEval-2017 consists of five subtasks; however, the first task (subtask A) is used in this article. SemEval-2017 Task 4 is a text sentiment classification task in which its input is a tweet, and the corresponding output is the overall sentiment of the tweet as positive, negative, or neutral. The statistics of the dataset for subtask A are given in Table 1. As it is shown, the training set of SemEval 2017 includes 50,333 samples, and the test set contains 12,284 samples.

Sentiment140 [37] is an English language dataset consisting of 1.6 million tweets. These tweets were extracted using the Twitter API and are annotated with two classes: positive and negative sentiment. We randomly selected a subset of 5000 texts from each class for our experiments, resulting in a balanced dataset of 10,000 tweets. This subset was chosen to ensure that our models were trained and evaluated on a manageable and representative sample of the larger dataset.

IMDB Movie Reviews contains 50,000 samples obtained from Stanford University (Maas et al., 2011). Each sample is a comment from audiences about the story of films. The sentiment labels could be categorized into positive and negative classes.

In all experiments, we have used the following measure as evaluation metrics: AvgRec, F_1^{PN} and accuracy. As mentioned in SemEval-2017 Task 4, the dataset is imbalanced, and among the measures, AvgRec is the robust one, and it is considered the primary measure for ranking. AvgRec is the average recall measure on three classes: positive, negative, and neutral. Hence, it is defined as follows:

$$AvgRec = \frac{1}{3}(R_p + R_N + R_U) \tag{12}$$

where R_p, R_N, and R_U respectively denote to recall measure with respect to positive, negative, and neutral classes. It is shown that this measure has more desirable theatrical properties, such as being invariant to class imbalance [38, 39]. F_1^{PN} is the another measure which is defined as the average of F1-measure on positive and negatives classes. Also, accuracy is the standard measures that compute the ratio of the correctly classified samples.

In the proposed approach, the hyper-parameters optimization is done using the grid search for the learning rate of embedding update network search over [0.1, 0.01, 0.001, 0.001, 0.0005, 0.0001, 0.00005, 0.00001]. In each iteration, we have picked up 5% of all available data in the training step (i.e., the total number of labeled and unlabeled samples). The reason is that adding too many pseudo-labeled samples to the labeled dataset may negatively impact the model's performance. This is because the model's predictions may introduce incorrect or noisy labels, especially if it is not highly confident. If a large number of incorrect labels are added, it could lead to a degradation in the model's ability to learn and make accurate predictions on new data.. Also, the value of β^t is set similar to [11].

This paper runs the program on a computer with Intel(R) Core (TM) i9 CPU, NVIDIA GeForce RTX 3090 24GB GDDR5, and 128 GB DDR4 RAM. We implemented our method using Python 3.8, TensorFlow, and Keras.

4.2 Ablation study

In this section, an ablation study is done to evaluate the effectiveness of the different modules of the proposed approach. We have created two versions of the proposed approach to investigate the effectiveness of the different modules of the proposed method. The first version is "our approach w.o. SS"; in this version, the semi-supervised (SS) module is removed, and we have added a fully connected layer with three neurons (i.e., the number of classes) as the task predictor layer. This version explores how the pseudo-labeling step impacts learning a better model. The second version is "our approach w.o. EU": In this case, the second module, the embedding update mechanism, which updates the representation of the tweets using the triplet loss function, is removed. The goal of this version is to consider how updating the representation of the tweets by using the extra pseudo-labeled data could impact the performance of the proposed approach. It should be noted that the iterative process of the proposed method is important because it creates a link between the second and the third steps.

Also, To compare the proposed approach with the current state-of-the-art approaches, we choose XGBoost, AdaBoost [17], BERTweet [40], DataStories [41], and BB_twtr [42] as baseline approaches.

As mentioned in this paper, the aim is to investigate the low data sentiment analysis and how we can utilize the unlabeled data to improve performance. Hence, we design a different set of experiments. We divided the training set into labeled and unlabeled sets in the first setting. In this case, the ratio of division is set to 0.1. In other words, it is assumed that the number of labeled tweets is 5033 and the remaining ones (45,300) are unlabeled. In this case, we have run the proposed approach, and after learning the model, the obtained performance is reported on the unlabeled training data. The obtained results are given in Table 2. As is shown, the obtained results are compared with some basic approaches, and as it is clear, the proposed approach significantly improves the results.

The second setting is similar to the first set, with the difference being that we utilized the test data and reported the result on the test data. For this setting, the obtained results are given in Table 3. As is shown, the proposed approach significantly improves the results in this case, too.

4.3 semEval-2017 dataset

In the third setting, to have a fair comparison with the state-of-the-art results, we consider all training data labeled and the test data unlabeled data. Then, we applied the learned model to the test set. The obtained results are given in Table 4. As is shown, our proposed approach, which uses only 10% of training data as labeled data and utilizes the remaining ones as unlabeled data, could achieve competitive performance with respect to state-of-the-art methods. Also, if we have utilized the full training set as the labeled data, we get the best performance with respect to the other methods. In this case, our approach improves 2.4% of the AvgRec measure over BERTweet [40].

It should be noted that in "Our approach w.o. SS" and "Our approach w.o. EU", only 10% of the training data is used as labeled data. Also, our approach is compared with the semi-supervised approaches. Also, our approach with only 10% of labeled training data outperforms [17] in different settings. It should be noted that [17] has utilized all training data to learn the model.

In the following, we study the language model's impact on the proposed approach's total performance. We have selected four language models, including word2vec, Glove, BERT, and XLNet. In Table 5, the obtained results are shown. As shown, XLNet could perform better with respect to the other approaches. In the proposed approach, we initialize an embedding feature space with the pre-trained language model and then, using the designed network, find a better representation in that space. Since XLNet provides better results, it provides better representation space for the approach.

In the following, we investigate how different ratios of the labeled samples impact the performance. For this case, we set the different numbers for the labeled ratio and report the results in Table 6. As shown, the results obtained confirm that by increasing the labeled ratio, we should achieve better performance. For each ratio, by comparing the performance of our approach with XGBoost, we found out that our approach demonstrates better performance in all measures.

Table 1 Statistics about the English training set of the SemEval 2017 dataset

Dataset	Task	Positive	Neutral	Negative	Total
Train	A	19,902	22,591	7,840	50,333
Test	A	2,375	5,937	3,972	12,284

4.3.1 Handling the imbalanced data

As it is mentioned, in training N_u, the training samples are imbalanced. We have done the following experiment to investigate how it impacts the final performance of the proposed method. In this case, we have utilized three techniques: oversampling the underrepresented class, class weighted loss function, and focal loss function instead of classic categorical entropy loss function. It should be noted that in the second and the third loss functions, we should change the loss function. In the current implementation, the utilized loss function is the triplet loss function, which focuses on the feature representation of the samples. In the class weighted loss function and focal loss function, we utilized both the supervised loss function and the triplet loss function. It should be noted that in this experiment and all other experiments in this subsection, only 10% of the labeled samples in the dataset are used as labeled data. The obtained results of this experiment is shown in Table 7. As it is shown in the first technique, oversampling the underrepresented class, when we use a low number of labeled samples, the performance decreases. The reason is that, for the pseudo-labeled samples, the oversampling technique can escalate the error if a sample is wrongly labeled. Also, we have used the class-weighted technique. The obtained results show that there is a slight improvement in the obtained results. The model showed little to no significant enhancement in its predictions. The focal loss function mitigates the class imbalance by adjusting the loss contribution of each class. Also, it dynamically focuses the model on misclassified or hard samples, improving overall performance and robustness. In our experiment, the focal loss function is defined as follows:

$$l_{focal} = -\sum \left(1 - p_i\right)^{\gamma} \log(p_i)$$

The obtained results in Table 7 show that there is a slight improvement when we have utilized the focal loss function along with the triplet loss function.

Table 2 The obtained results of the proposed approach for the first setting and its comparison with baselines

Type	Approach	AvgRec	F_1^{PN}	Accuracy
Supervised	XGBoost	54.7	54.1	55.2
	AdaBoost	49.1	48.6	48.9
	Our approach w.o. SS	60.1	59.6	59.2
Semi-supervised	Our approach w.o. EU	65.7	65.1	64.9
	Our approach	72.0	71.5	71.4

Table 3 The obtained results of the proposed approach for the second setting and its comparison with baselines

Type	Approach	AvgRec	F_1^{PN}	Accuracy
Supervised	XGBoost	54.5	53.6	54.1
	AdaBoost	49.3	48.4	47.8
	Our approach w.o. SS	59.8	59.0	59.8
Semi-supervised	Our approach w.o. EU	63.9	63.5	62.8
	Our approach	71.2	70.3	70.5

Table 5 Investigate the impact of the language model on the total performance of the proposed approach

Language Model	AvgRec	F_1^{PN}	Accuracy
Word2vec	62.5	62.0	61.7
Glove	64.8	63.7	64.1
BERT	70.4	69.7	69.9
XLNet	71.2	70.3	70.5

4.3.2 Consistency regularization

Consistency regularization encourages the model to produce consistent outputs when the input is perturbed. This can enhance the robustness and generalization. In this case, the way of input perturbations plays an important role. In this study, we have used a simple approach for random augmentations by removing a random word from the tweet. In future work, other approaches to perturb the input could be utilized. Also, we have modified the total loss function of the approach. In this case, it is the sum of the triplet loss function and consistency loss function, where the consistency loss measures the difference between the features extracted from original and perturbed inputs. In Table 8, the obtained results are given. As shown, in two measures, including AvgRec and Accuracy, adding consistency regularization could improve the results. However, we can see a decrease meant in F_1^{PN}. Future work will involve exploring different data augmentation approaches to understand their effects and identify the most effective strategies for our specific use case.

Adversarial training could be utilized to add regularization. In adversarial training for robustness, the adversary is an algorithm or process designed to generate adversarial examples and integrate these samples into the training loop of the primary model. Various methods exist to generate adversarial examples, including but not limited to: 1)Fast Gradient Sign Method (FGSM): A simple, gradient-based method to generate adversarial examples by perturbing the input in the direction of the gradient of the loss function. 2) Projected Gradient Descent (PGD): An iterative method that is more powerful than FGSM, involving multiple steps of gradient descent in the direction of the loss function, with a projection step to ensure the perturbation stays within a predefined limit. 3) Carlini & Wagner (C&W) Method: Focuses on generating adversarial examples with minimal perturbation, using a different optimization objective. Moreover, the loss function should be modified by correctly classifying both legitimate and adversarial inputs. Modifying a feature extraction network trained with triplet loss to incorporate adversarial training involves significant changes to your

Table 4 The obtained results of the proposed approach and its comparison with baselines

	Approach	AvgRec	F_1^{PN}	Accuracy
Supervised	XGBoost	58.6	57.5	58.6
	BB_twtr [42]	68.1	68.5	65.8
	DataStories [41]	68.1	67.7	65.1
	BERTweet [40]	73.2	72.8	71.7
	Our approach w.o. SS	59.8	59.0	59.8
Semi-supervised	[17]- CART	-	-	61.1
	[17]- ExtraTrees classifier	-	-	61.1
	[17]- HistGradientBoost classifier	-	-	61.1
	[17]- Stacking classifier	62.7	55.4	62.7
	Our approach w.o. EU	63.9	63.5	62.8
	Our approach (10% of training data)	71.2	70.3	70.5
	Our approach (All training data)	75.6	74.8	73.7

Table 6 Investigate the impact of the different labeled ratios on the total performance of the proposed approach

Labeled Ratio	Type	Approach	AvgRec	F_1^{PN}	Accuracy
5%	Supervised	XGBoost	54.1	52.6	53.3
	Semi-supervised	Our approach	69.4	68.9	69.0
10%	Supervised	XGBoost	54.5	53.6	54.1
	Semi-supervised	Our approach	71.2	70.3	70.5
15%	Supervised	XGBoost	55.1	54.8	55.7
	Semi-supervised	Our approach	72.1	71.6	70.9
20%	Supervised	XGBoost	56.2	56.6	57.2
	Semi-supervised	Our approach	72.8	72.1	71.3
25%	Supervised	XGBoost	56.9	57.3	58.1
	Semi-supervised	Our approach	73.1	72.8	72.2
30%	Supervised	XGBoost	56.3	56.9	57.9
	Semi-supervised	Our approach	73.3	73.1	72.9

Table 7 The effectiveness of the different methods for handling the imbalanced data in the proposed method

Method	AvgRec	F_1^{PN}	Accuracy
oversampling the underrepresented class	69.3	68.9	69.1
class weighted loss function	71.29	70.29	70.53
focal loss function	71.41	70.60	70.48
Our approach	71.28	70.34	70.52

Table 8 The impact of consistency regularization on the performance of the proposed method

Method	AvgRec	F_1^{PN}	Accuracy
consistency regularization	71.61	69.81	71.02
Our approach	71.28	70.34	70.52

training objective and pipeline. Hence, it could be considered as future work.

4.3.3 Sensitivity to hyperparameters

This section shows how the proposed approach performance is sensitive to the different values of the hyperparameters. To do so, we have done an experiment in which the obtained measures are reported for different values of the learning rate, including 0.1, 0.01, 0.001, 0.001, 0.0005, 0.0001, 0.00005, 0.00001. As shown, when the learning rate is set to 0.0001, the proposed method performs better on the average of the three measures (Table 9).

Table 9 The impact of the different values of the learning rate on the performance of the proposed method

Language Model	AvgRec	F_1^{PN}	Accuracy
0.1	56.1	57.2	55.6
0.01	62.9	61.8	60.0
0.001	69.5	69.1	69.9
0.0005	71.5	70.1	69.8
0.0001	71.2	70.3	70.5
0.00005	71.3	69.6	68.9
0.00001	70.8	69.2	69.1

It should be noted that we have used fixed values for the two other hyper parameters and have not utilized the grid-search optimization technique.

4.4 Sentiment140 dataset

The obtained results for the Tweets Airline dataset is shown in Table 10. When our algorithm uses the whole training sample, the obtained results are higher than all other approaches. Also, when only 10% of the training data were used, we improved the results obtained with respect to the other comparable approaches.

4.5 IMDB movie reviews

The obtained results of the proposed approach on the IMDB Movie reviews dataset are shown in Table 11. The comparison with baselines shows that the proposed method achieves better results compared to the other approaches. Our approach gets a 2.1% improvement over the stacking classifier when utilizing the whole training samples. Also,

Table 10 The obtained results of the proposed approach and its comparison with baselines on the Sentiment140 dataset

	Approach	AvgRec	F_1^{PN}	Accuracy
Supervised	XGBoost	64.8	63.9	65.4
	Our approach w.o. SS	64.2	63.7	64.7
Semi-supervised	[17]- CART	-	-	71.5
	[17]- ExtraTrees classifier	-	-	71.4
	[17]- HistGradientBoost classifier	-	-	71.4
	[17]- Stacking classifier	-	-	72.8
	Our approach w.o. EU	73.9	71.8	73.6
	Our approach (10% of training data)	75.3	72.0	74.2
	Our approach (All training data)	81.1	79.8	80.3

Table 11 The obtained results of the proposed approach and its comparison with baselines on the IMDB Movie reviews dataset

	Approach	AvgRec	F_1^{PN}	Accuracy
Supervised	XGBoost	58.6	57.5	58.6
	Our approach w.o. SS	59.8	59.0	59.8
Semi-supervised	[17]- CART	-	-	76.7
	[17]- ExtraTrees classifier	-	-	76.7
	[17]- HistGradientBoost classifier	-	-	76.7
	[17]- Stacking classifier	-	-	77.2
	Our approach w.o. EU	77.0	76.5	76.9
	Our approach (10% of training data)	78.6	77.9	78.1
	Our approach (All training data)	79.4	76.9	79.3

when it uses only 10% of the training samples as labeled samples, it achieves a 0.9% improvement over the stacking classifier.

5 Conclusion

In this paper, we have proposed a new approach for low-data sentiment analysis. To this end, we first utilized one of the pre-trained language models to create a feature representation for each raw tweet. Then, we design a network that learns to project the representation into a space such that tweets with similar sentiments have a similar representation and tweets with different sentiments have a distant representation. To do so, the network is learned via the triplet loss function. Then, we fed the labeled and unlabeled tweets into a semi-supervised approach to assign a label to some unlabeled tweets with high confidence. Then, the pseudo-labeled data are added to the labeled data, and it feeds into the embedding network again to learn more discriminative feature space.

We have done a huge experiment to evaluate the proposed approach from different aspects. We explored the role of the pre-trained language model, and the results showed that XLNet could provide more discriminative representation. The results show that the proposed approach could significantly improve performance with respect to the comparable base approaches by using only 10 percent of the training set of SemEval2017 as labeled data and the remaining ones as unlabeled data. Also, we evaluated the performance of the proposed approach by using all training data as labeled data and test data as unlabeled data. The results indicate that our approach performed better than comparable approaches. The results obtained on IMDB Movie Reviews and Sentiment 140 were better than those of comparable models.

One of the advantages of the proposed approach is that the embedding update module propagates the error-correcting signals to a semi-supervised module. It is done through similarity matrix computation using the updated feature representation vector. Also, the semi-supervised module updated the set of labeled data and fed it into the embedding update module to update the representation space.

Our approach currently involves complex computations to calculate the similarity matrices. Also, it needs more memory requirement. This step of the algorithm is well-suited for parallelization due to its nature (computing the similarity between each pair is independent of the other pairs). Specifically, as future work, we can distribute the computational workload across multiple threads, processors, or computing nodes, enabling concurrent execution of independent tasks. This parallelization strategy has the potential to significantly reduce training time, especially when utilizing modern parallel computing frameworks.

Acknowledgements We acknowledge anonymous reviewers for the valuable comments on the original manuscript.

Authors contribution SHR: conceptualization, data curation, result analysis, methodology, writing, review & editing. ASH&MH: result analysis, project administration, review & editing. JT: conceptualization, supervision, project administration, review & editing.

Data availability The sample data for SSSA are available at: https://alt.qcri.org/semeval2014/task9/

Declarations

Competing interests The authors declare that they have no competing interests.

Ethical and informed consent for data used All of the authors agree to the ethical and informed consent for data used.

References

1. Sebastiani F, Esuli A (2006) SentiWordNet: a publicly available lexical resource for opinion mining. in 5th international conference on language resources and evaluation. In: European language resources association (ELRA), pp 417–422
2. Gupta R (2019) Data augmentation for low resource sentiment analysis using generative adversarial networks. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 7380–7384
3. Goodfellow I et al (2014) Generative adversarial nets. Adv Neural Inf Process Syst 27
4. Meetei LS et al (2021) Low resource language specific pre-processing and features for sentiment analysis task. Lang Resour Eval 55(4):947–969
5. Yu Y, Zhang D (2022) Few-shot multi-modal sentiment analysis with prompt-based vision-aware language modeling. In: 2022 IEEE international conference on multimedia and expo (ICME), pp 1–6
6. Triantafillou E et al (2018) Meta-learning for semi-supervised few-shot classification. International conference on learning representations. arXiv preprint arXiv:1803.00676
7. Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440
8. Yang X et al (2022) A survey on deep semi-supervised learning. IEEE Trans Knowl Data Eng 35(9):8934
9. Dong-Hyun L (2013) Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning. ICML, p 896
10. McLachlan GJ (1975) Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. J Am Stat Assoc 70(350):365–369
11. Tanha J, Van Someren M, Afsarmanesh H (2014) Boosting for multiclass semi-supervised learning. Pattern Recogn Lett 37:63–77
12. Xie Q et al (2020) Self-training with noisy student improves ImageNet classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10687–10698
13 Zhang B et al (2021) Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. Adv Neural Inform Proc Syst 34:18408
14. Yadav A, Vishwakarma DK (2020) Sentiment analysis using deep learning architectures: a review. Artif Intell Rev 53(6):4335–4385
15. Wankhade M, Rao ACS, Kulkarni C (2022) A survey on sentiment analysis methods, applications, and challenges. Artif Intell Rev 55(7):5731–5780
16. Wu C et al (2019) Semi-supervised dimensional sentiment analysis with variational autoencoder. Knowl-Based Syst 165:30–39
17. Tesfagergish SG, Kapočiūtė-Dzikienė J, Damaševičius R (2022) Zero-shot emotion detection for semi-supervised sentiment analysis using sentence transformers and ensemble learning. Appl Sci 12(17):8662
18. Tiwari D, Nagpal B (2022) KEAHT: A Knowledge-Enriched Attention-Based Hybrid Transformer Model for Social Sentiment analysis. New Gener Comput 11:1–38
19. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
20. Almalis I, Kouloumpris E, Vlahavas I (2022) Sector-level sentiment analysis with deep learning. Knowl-Based Syst 258:109954
21. Dang NC, Moreno-García MN, De la Prieta F (2020) Sentiment analysis based on deep learning: A comparative study. Electronics 9(3):483
22. Chen Z, Qian T (2022) Retrieve-and-edit domain adaptation for end2end aspect based sentiment analysis. IEEE/ACM Trans Audio, Speech Language Proc 30:659–672
23. Singh C et al (2022) A deep learning approach for sentiment analysis of COVID-19 reviews. Appl Sci 12(8):3709
24. Wu H, Shi X (2022) Adversarial soft prompt tuning for cross-domain sentiment analysis. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2438-2447)
25. Tan KL et al (2022) RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network. IEEE Access 10:21517–21525
26. Mathew L, Bindu VR (2022) Efficient transformer based sentiment classification models. Informatica 46(8)
27. Zhang T, Gong X, Chen CP (2021) BMT-Net: Broad multitask transformer network for sentiment analysis. IEEE Trans Cybern 52(7):6232–6243
28. Chen T et al (2020) Big self-supervised models are strong semi-supervised learners. Adv Neural Inf Process Syst 33:22243–22255
29. Zheng M et al (2022) Simmatch: semi-supervised learning with similarity matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14471–14481
30. Sohn K et al (2020) Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Adv Neural Inf Process Syst 33:596–608
31. Mikolov T et al (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26
32. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. InProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532–1543).
33. Devlin J, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
34. Yang Z (2019) XLNet: generalized autoregressive pretraining for language understanding for language understanding. In: Advances

in neural information processing systems. arXiv preprint arXiv:1906.08237
35. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22:888–905
36. Schroff F et al (2015) Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 815–823
37. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. In: CS224N project report, vol 1, 12. Stanford
38. Sebastiani F (2015) An axiomatically derived measure for the evaluation of classification algorithms. In: Proceedings of the 2015 international conference on the theory of information retrieval, pp 11–20
39. Nakov P et al (2016) Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. Lang Resour Eval 50(1):35–65
40. Nguyen DQ, Vu T, Nguyen AT (2020) BERTweet: A pre-trained language model for English Tweets. arXiv preprint arXiv:2005.10200.
41. Baziotis C, Pelekis N, Doulkeridis C (2017) Datastories at semeval-2017 task 4: deep LSTM with attention for message-level and topic-based sentiment analysis. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 747–754
42. Cliche M (2017) BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs. Association for Computational Linguistics, pp 573–580

Shima Rashidi was born in Iran in 1989. She received the B.E. and M.E. degrees in computer science from the University of Tabriz, Tabriz, Iran, in 2011 and 2013, respectively. Now, She is a Ph.D. student at the University of Science and Technology, Tehran, Iran. Currently she is an assistant lecturer in a university of Human development, Kurdistan region, sulaymaniyah, Iraq. Her main areas of research interest are text mining, semi supervised learning, social network analysis.

Jafar Tanha was born in Bonab, Iran. He received the B.Sc. and M.Sc. degrees in computer science from the Amirkabir University of Technology (Polytechnic), Tehran, Iran, in 1999 and 2001, respectively, and the Ph.D. degree in computer science-artificial intelligence from the University of Amsterdam (UvA), Amsterdam, The Netherlands, in 2013. He joined with the INL Institute, Leiden, The Netherlands, as a Researcher, from 2013 to 2015. Since 2015, he has been with the Department of Computer Engineering, Payame Noor University, Tehran, where he was an Assistance Professor. He has held lecturing positions with the Iran.

Arash Sharifi was born in Tehran, Iran. He received the Bachelor's degree from Islamic Azad University, South Tehran Branch. And he got Master's degree from Islamic Azad University, Science and Research Branch and PhD from Islamic Azad University, Science and Research Branch. Now, he is an Assistant Professor in SRBIAU. His main areas of research interest are Machine Learning, Soft Computing and Deep Learning.

Mehdi Hosseinzadeh (Member, IEEE) received the B.S. degree in computer hardware engineering from Islamic Azad University, Dezful Branch, Iran, in 2003, and the M.Sc. and Ph.D. degrees in computer system architecture from Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2005 and 2008, respectively. He is the author/co-author of more than 250 publications in technical journals and conferences. His research interests include SDN, information technology, data mining, big data analytics, e-commerce, e-marketing, and social networks.