



# Modeling Tweet Dependencies with Graph Convolutional Networks for Sentiment Analysis

Abdalsamad Keramatfar<sup>1</sup> · Hossein Amirkhani<sup>1</sup> · Amir Jalaly Bidgoly<sup>1</sup>

Received: 18 May 2021 / Accepted: 17 December 2021 / Published online: 13 February 2022  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Nowadays, individuals spend significant time on online social networks and microblogging websites, consuming news and expressing their opinions and viewpoints on various topics. It is an excellent source of data for various data mining applications, such as sentiment analysis. Mining this type of data presents several challenges, including the posts' short length and informal language. On the other hand, microblog posts contain a high degree of interdependence, which can help to improve sentiment classification based on text. This data can be represented as a graph, with nodes representing posts and edges representing the various relationships between them. By using recently developed deep learning models for graph structures, this approach enables efficient sentiment analysis of microblog posts. This paper utilizes graphs to represent microblog posts and their various relationships, such as user, friendship, hashtag, sentimental similarity, textual similarity, and common friends. It then employs graph neural networks to perform context-aware sentiment analysis. To make use of the knowledge contained in multiple graphs, we propose a stacking model that simultaneously employs multiple graph types. The findings demonstrate the relevance of sociological theories to the analysis of social media. Experimental results on HCR (a real-world Twitter sentiment analysis dataset), indicate that the proposed approach outperforms baselines and state-of-the-art models.

**Keywords** Graph neural network · Graph convolutional network · Sentiment analysis · Social media · Opinion mining · Deep learning

## Introduction

Social media platforms have become indispensable tools for people to communicate with one another in the modern era. Individuals write about almost every aspect of their lives, including their interests, experiences, accomplishments, and failures. These activities enable the analysis of public sentiment toward a variety of subjects, from products to politicians. Sentiment analysis [1] is a rapidly growing field of analysis that elicits people's emotional responses to various topics [2]. Additionally, the sentiment analysis research community has recently shifted its focus from online reviews toward social media [3].

The primary focus of social media sentiment analysis research has been on the document's textual content [4], implicitly assuming that the posts are independent. Social

media posts, on the other hand, are not actually independent. Additionally, performing sentiment analysis solely on textual data presents numerous difficulties in practice. For example, some social media platforms, such as Twitter, limit the length of posts (280 characters); as a result, textual features are available for analysis in fewer numbers [5–7]. Other issues arise from the way that people use social media, such as when they type in informal language [8].

While social media has its own set of constraints, it also provides additional information, such as various types of links between posts. According to the theory of sentiment consistency [9], a particular person's tweets are more likely to share similar sentiments than random tweets [5]. This relationship can be used in conjunction with contextual analysis. Furthermore, other theories such as homophily or assortative mixing [10, 11], and emotional contagion [12] demonstrate that related individuals share similar viewpoints [5]. According to the first theory, similarity fosters the formation of new relationships. As a result, individuals within the same networks exhibit a high degree of homogeneity. Moreover, research indicates that happy/sad people tend to

✉ Abdalsamad Keramatfar  
Akeramatfar@qom.ac.ir

<sup>1</sup> Computer Engineering and Information Technology  
Department, University of Qom, Qom, Iran

interact with other happy/sad people in online social networks [10]. According to the second theory, an individual's emotional state affects those around him or her. Additionally, experimental evidence indicates that social networks are capable of transmitting emotion [13].

Several other clues also exhibit connections between various posts. For instance, there is the hashtag concept, which is used to specify the subject of a post. Comments about specific products demonstrate varying degrees of sentimental consistency. For example, comments about the MacBook are rated higher than those about other laptops. As a result, tying together posts that share a common hashtag may aid in sentiment analysis. Furthermore, a relationship between posts can be established using text similarity based on various text representations, including Tf-IDF, Word2Vec [14], GloVe [15], and Universal Sentence Encoder (USE) [16].

Deep learning methods have recently achieved outstanding results in various natural language processing (NLP) tasks, including sentiment analysis [7]. Moreover, the machine learning community has developed deep learning-based methods for graph analysis, including graph-specific embeddings [17], node classification [18, 19], and link prediction [20, 21]. The central tenet of this field of research is that models should use the information contained in the relationships between entities. In other words, these models generate representations that can encode both local graph structures and node properties [19]. These representations can then be used within the network or for subsequent tasks such as node classification.

As an illustration, the following is a tweet that was correctly labeled by the ChebNet (User) but incorrectly labeled by the feed-forward text-based model (the models are described in the materials and methods section in more detail):

“#HCR Our greatest threat to society is when we stop caring about one another! It's a damn shame #TCOT #TEAPARTY Cant understand that”

While this is a positive tweet about healthcare reform, the text-based method underestimates its sentiment due to several negative features such as threat, stop, and shame. The ChebNet model, on the other hand, correctly predicts the sentiment because the tweep (the user who posted the tweet) has other positive tweets about HCR, including the following tweets:

“Call #Congress today. Tell them YES on #hcr! (202) 224-3121 #PUBLICOPTION”  
 “RT @TeresaS: Henry Aaron, David Cutler, Alice Rivlin, et al.: #hcr bill is crucial to reducing the deficit (PDF): <http://bit.ly/cAf351>”

Based on the findings above, this paper models social media sentiment analysis as a node classification problem and solves it using graph neural networks (GNNs). This paper makes the following contributions:

- We use a graph to model social media posts and their various types of relationships, such as user, friendship, common friends, common hashtag, text-based similarity, and sentimental similarity.
- We utilize deep learning models, including feed-forward neural network and graph convolutional network (GCN) in the context-aware social media sentiment analysis problem. To our knowledge, this is the first time that the GCN has been used for multi-thread context-aware sentiment analysis on social media.
- To enable the use of existing knowledge across multiple graphs, we provide a stacking model that is more accurate than any of the base models.
- We conduct extensive experimental studies on social media sentiment analysis using a variety of different edge sets.

The remainder of the paper is organized as follows. The following section, titled “[Related Research](#)”, discusses notable papers on deep learning and sentiment analysis in social media. The “[Materials and Methods](#)” section discusses the architecture of the models used. The “[Results](#)” section contains experimental results from a real-world sentiment analysis dataset, and the “[Conclusion](#)” section summarizes the study's findings.

## Related Research

This section discusses the papers that are related to the current study. The section is divided into two subsections. We begin by reviewing the research on social media sentiment analysis and then examine deep learning methods in this field.

### Social Media Sentiment Analysis

Social media is a valuable source of data for various data mining applications, including sentiment analysis. The majority of research on social media sentiment analysis has relied entirely on the text of the posts [4, 22]. Traditional machine learning models such as support vector machines, logistic regression, decision tree [23, 24], and their stacking [25, 26] were trained on various textual features such as n-grams and parts of speech. Other researchers [27] consulted sentimental dictionaries and knowledge bases, such as General Inquirer [28], SentiWordNet [29], WordNet-Affect [30], SenticNet [31], SentiStrength [32], and Bing Liu dictionary [33], which contain information about the polarity orientation of words and phrases. They primarily used a combining function to infer the sentiment of a post by analyzing the polarity of phrases. Furthermore, negations and intensifications were frequently taken into account [34]. A separate line of research combined lexicon-based and machine learning approaches [26, 32, 35]. There are reports of leveraging the

interdependence of social media posts to improve sentiment prediction accuracy. For example, several papers included a regularization term in the supervised loss function [5, 6].

## Deep Learning Methods in Social Media Sentiment Analysis

Deep learning has improved the accuracy of a variety of NLP tasks [22] including sentiment analysis [36]. Researchers used popular deep learning models in this line, including feed-forward neural networks [37, 38], recurrent neural networks (RNN) [22, 39], convolutional neural networks (CNN) [40], and mixture, ensemble, or stack of them [41–43]. The attention mechanism [44] has also been used in conjunction with long-short term memory (LSTM) [8, 39, 45, 46] to improve token combination and to determine the relevance of sentiment words in relation to various aspects of the sentence [47]. Word and document embedding has been a significant achievement of NLP. The issue with the pre-trained embeddings was that the embeddings were not originally tailored for sentiment analysis, and some semantically related but sentimentally distinct words were projected to close points in the embedding space. This motivated researchers to propose specific embeddings which include sentimental clues of words [48–51].

Apart from classical deep learning techniques, various other techniques have been used to improve sentiment prediction accuracy. For example, social media posts may contain video, image, or audio; consequently, multimodal sentiment analysis [52] has been another topic in social media sentiment analysis, combining audio and visual cues into deep structures [53, 54]. Due to the complexity of natural language, recent efforts have been made to combine symbolic and subsymbolic artificial intelligence [31] and to leverage the power of neural tensor networks for modeling relational data in conversational sentiment analysis [55]. Moreover, because deep models contain a large number of parameters and require a large amount of data, some researchers attempted to pretrain models using distantly supervised tweets [56, 57]. Multi-task learning has also been applied to sentiment analysis and related tasks such as sarcasm and personality detection [58–60]. Additionally, similar to traditional methods, several efforts have been made to leverage sentiment lexicons in conjunction with deep learning models [31, 40, 61].

As a result of GNNs' recent success in several fields, sentiment analysis research has also tapped into their potential. Examples include the use of GCNs to model the relationships between sentence aspects [62], the use of graph network embedding enabled by a variational auto-encoder [63], the use of syntax and knowledge graphs to augment the sentence representation for a given aspect [64], the construction of a word-document graph [65], and the multi-level text

representation via message-passing [66]. Finally, some deep learning researchers have improved sentiment analysis by incorporating a variety of contexts. Some researchers modeled the problem as a sequence classification problem. They classified sentiment using hierarchical LSTM architectures in several contexts, including reply/retweet, hashtag, and user [8, 22]. Recently, Zhang et al. [67] used Bidirectional Encoder Representation from Transformers (BERT) with Bidirectional LSTM (BiLSTM) and Conditional Random Field (CRF) layers on top of BERT to model the sequence of tweets.

## Materials and Methods

### Lasso

The least absolute shrinkage and selection technique is a type of regression analysis that employs L1 regularization. Its loss is formally calculated as follows:

$$L = \frac{1}{n} \text{supervised}_{loss} + \beta \text{regularization}, \quad (1)$$

where  $\beta$  denotes the weight of regularization, and the *supervised\_loss* is defined as follows:

$$\text{supervised}_{loss} = \frac{1}{2} \|X\theta - Y\|_F^2, \quad (2)$$

where  $X$ ,  $\theta$  and  $Y$  represent feature, weight, and label arrays, respectively. The regularization term is defined as:

$$\text{regularization} = \|\theta\|_1, \quad (3)$$

The aforementioned prevents the model from overfitting. Additionally, by removing unnecessary features, this term acts as an automatic feature selector.

### Graph Regularization

Graph regularization is a technique for incorporating graph information into supervised learning techniques. The idea is to add a regularization term to the regular loss to reduce the difference between connected node predictions. It accomplishes this by minimizing the following loss:

$$L = \text{supervised}_{loss} + \alpha \text{graph}_{regularization} + \beta \text{regularization}, \quad (4)$$

where  $\alpha$  denotes the weight of graph regularization. The only difference with Lasso is the addition of the second term, which is defined as:

$$\text{graph}_{regularization} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n A_{ij} \|X_i\theta - X_j\theta\|^2, \quad (5)$$

where  $A_{ij}$  denotes the weight of the edge between nodes  $i$  and  $j$ .  $X_i\theta$  is the prediction for  $X_i$ .

Two baseline methods for graph regularization are presented: a sociological approach to handling noisy and short texts (SANT), which makes use of user and friend graphs [6], and Sentiment Analysis using Structure Similarity (SASS), which makes use of structural similarity in addition to the user graph [5]. If tweets  $i$  and  $j$  are connected by a friendship link, their structural similarity is computed as:

$$S_{ij} = \frac{|N_{u_i} \cap N_{u_j}|}{|N_{u_i} \cup N_{u_j}|} + 1, \tag{6}$$

Otherwise, their structural similarity is computed as:

$$S_{ij} = \frac{|N_{u_i} \cap N_{u_j}|}{|N_{u_i} \cup N_{u_j}|}, \tag{7}$$

where  $N_{u_i}$  denotes the set of  $u_i$ 's friends.

### Deep Learning

Before the advent of deep learning, supervised machine learning focused on learning weights to minimize the loss function using expert-engineered features. By contrast, deep learning fundamentally altered this paradigm by allowing the system to acquire representations that result in improved performance. Through their hidden layers, deep learning methods attempt to learn more abstract representations. The feed-forward neural network is a biologically inspired system of processing units resembling brain neurons [68]. Each layer of a feed-forward neural network performs as follows:

$$H^{(l+1)} = f(H^{(l)}) = g(b^{(l+1)} + \theta^{(l+1)}H^{(l)}), \tag{8}$$

where  $H^{(l+1)}$  denotes the activation of the layer  $l + 1$ ,  $b^{(l)}$  represents the bias vector, and  $\theta^{l+1}$  is the trainable parameter matrix of layer  $l + 1$ . Additionally,  $g$  is a non-linear function, typically sigmoid, ReLU or tanh. Note that  $H^{(0)}$  is the input feature vector to the network. Typically, the final layer contains a softmax function for classification purposes, which generates a probability distribution for the output classes:

$$\text{Softmax}(z)_i = \frac{\exp(z_i)}{\sum_{c=1}^C \exp(z_c)}, \tag{9}$$

where  $z_i$  denotes the  $i^{\text{th}}$  output of the final layer before applying softmax, and  $C$  represents the number of classes. The cross-entropy is the standard loss function used for classification as follows:

$$L = - \sum_{c=1}^C y_c \log(\text{Softmax}(z_c)), \tag{10}$$

where  $y_c$  denotes the  $c^{\text{th}}$  element of the one-hot label vector. Further detailed information is available in [69].

### Graph Convolutional Neural Network

Convolutional neural networks were inspired by the early findings in the study of human visual system. Studies based on the functional magnetic resonance imaging and intracranial depth recordings demonstrated the correspondence between the hierarchy of the human visual areas and the CNN layers [70, 71]. The proposed method in this paper is based on GCNs, which generalize the convolution operation from the grid to graph data.

The independence of instances is a fundamental premise of existing machine learning algorithms. As previously stated, this is not the case with social media data. GNNs are a subclass of neural network architectures that use the graph structure to convolutionally aggregate data from neighborhoods [72, 73]. The fundamental concept is to generate a node's representation by aggregating its features with those of its neighbors. Convolution of a signal  $x$  with a filter  $g_\theta \in R^n$  is defined as:

$$x *_G g_\theta = U((U^T x) \odot (U^T g_\theta)), \tag{11}$$

where  $U = [u_0; u_1; \dots; u_{n-1}] \in R^{n \times n}$  denotes the eigenvectors matrix of the normalized graph Laplacian.  $U^T x$  denotes the graph Fourier transform of the signal  $x$ , and  $\odot$  is the element-wise multiplication operation. In this paper, we use two types of GNNs: the Chebyshev Spectral CNN (ChebNet) [74] and GCN-Kipf [19]. ChebNet [74] employs the filter of Chebyshev polynomials of the eigenvalues diagonal matrix as follows:

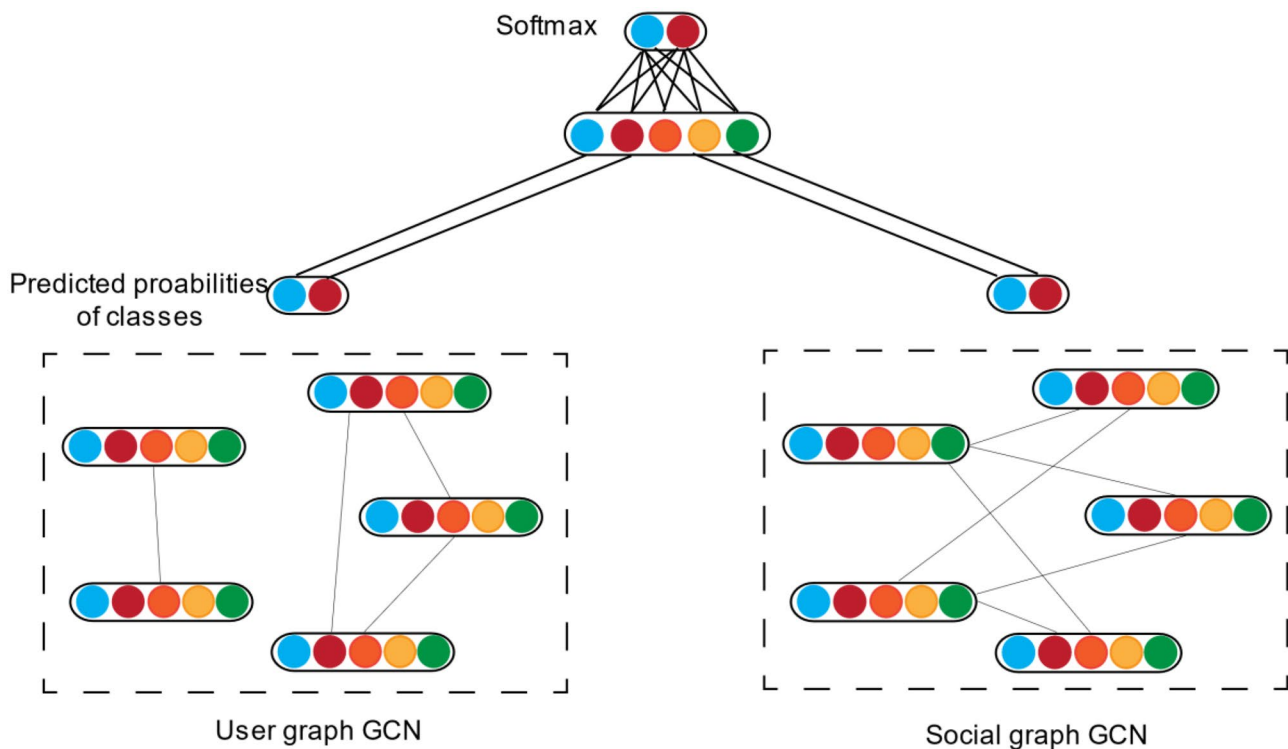
$$g_\theta \approx \sum_{i=0}^k \theta_i T_i(\tilde{\Lambda}) \tag{12}$$

where  $\tilde{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - I_n$ ,  $I_n$  denotes the identity matrix,  $\Lambda$  is the diagonal matrix of the eigenvalues, and  $\lambda_{max}$  represents the largest eigenvalue.  $T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$ ,  $T_0(x) = 1$ , and  $T_1(x) = x$ . Thus, filtering the signal  $x$  by  $g_\theta$  can be written as:

$$x *_G g_\theta \approx \sum_{i=0}^k \theta_i T_i(\tilde{L})x, \tag{13}$$

where  $\tilde{L} = \frac{2L}{\lambda_{max}} - I_n$ , and  $L$  denotes the Laplacian matrix.

GCN-Kipf [19] assumes  $k = 1$  (in the Chebyshev filter),  $\lambda_{max} = 2$ , and  $\theta = \theta_0 = -\theta_1$ . According to this formulation, each layer of the GCN-Kipf can be written as:



**Fig. 1** The proposed stacking model

$$H^{(l+1)} = f(H^{(l)}, A) = g\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} \theta^{(l+1)}\right), \quad (14)$$

where  $H^{(l)}$  denotes the activation of the layer  $l$ ,  $g$  is the ReLU function,  $\tilde{A}$  is the graph's adjacency matrix combined with self-neighborhood ( $\tilde{A} = A + I_n$ ),  $\tilde{D}$  represents the diagonal degree matrix where  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$  and  $\theta^{(l+1)}$  is a trainable weight matrix. The argument of  $g$  is referred to as the propagation rule, and it is the primary difference between different GCNs. Equation (14) employs a propagation rule representing each node as the normalized sum of its normalized neighbors' representations. Our experiments make use of GCNs with two hidden layers. Similar to feed-forward neural network layers, GCN layers can be stacked to extract high-level node representations. We use the TF-IDF representation of tweets' text as input for all machine learning models. We implement all models in TensorFlow except for GNNs, which are implemented in Python using the GCN package (<https://github.com/tkipf/gcn>).

### Stacking Model

Different graphs provide a node with different neighbors, resulting in different node representations in model. As a result, multiple predictions are generated for a single test node. A stacking model based on multiple GCNs is proposed to leverage the knowledge embedded in different graphs. A feed-forward neural network is trained on the

outputs of the two best classifiers on the validation set, including ChebNet (User) and GCN-Kipf (Social). These classifiers have an aggregation layer with 17 and 32 hidden nodes and are trained with a learning rate of 0.0007 and 0.01, respectively. The feed-forward network is composed of an input layer with four nodes (two for each base classifier, i.e., one for each of the positive and negative classes), a hidden layer with ten nodes (using the ReLU activation function), and an output layer with two nodes (one for each class) and a softmax activation. Adam is used for optimization. Figure 1 depicts the architecture of the proposed stacking model.

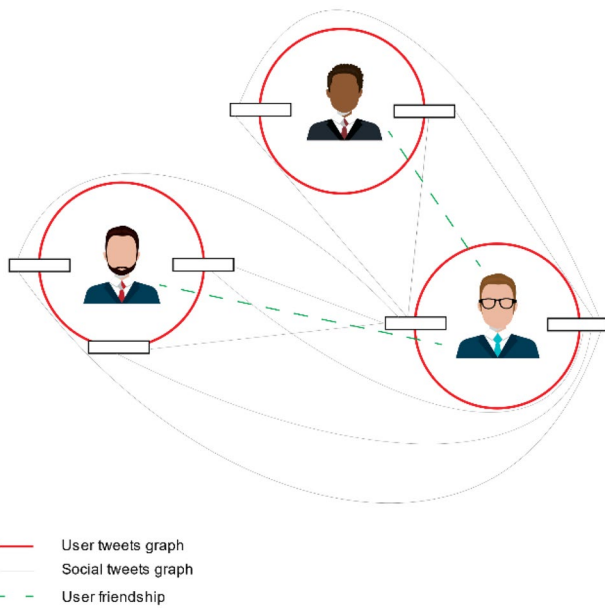
## Results

### Setup

Google Colaboratory<sup>1</sup> is used as the implementation environment, a free research tool equipped with a Tesla K80 GPU and 12G RAM. The experiments are conducted using the HCR dataset<sup>2</sup> [74], including tweets about the healthcare reform. This paper considers only tweets from

<sup>1</sup> <https://research.google.com/colaboratory/>

<sup>2</sup> <https://bitbucket.org/speriosu/updown/wiki/Home>



**Fig. 2** Illustration of user and social graphs

users who have at least one friend in the dataset. Sixty percent of the total data is used for training, 20% for validation, and 20% for testing. There are 988 negative tweets (73.62%) and 354 positive tweets (26.38%). Our experiments employ the following graphs:

- User: This graph connects all of the tweets from the same tweep. In other words,  $U_{ij} = 1$  if tweets  $i$  and  $j$  were both posted by the same tweep. There are 743 unique tweeps, with an average of  $\sim 1.80$  tweets per tweep (Fig. 2).
- Social: The friendship graph is based on data crawled by Kwak et al. [75], representing a snapshot of Twitter in 2009 (<http://an.kaist.ac.kr/traces/WWW2010.html>). Each tweet is connected to the tweets posted by the tweep’s followers or followings. In other words,  $S_{ij} = 1$  if tweet  $i$ ’s tweep is a follower/following of tweet  $j$ ’s tweep (Fig. 2).
- Common friends: Similarity detection via common friends is a well-established technique in social network analysis [5]. We construct a graph in which two tweets are connected if their respective tweeps share friends. In

other words,  $CF_{ij} = 1$  if tweets  $i$  and  $j$  were created by two tweeps who share at least one friend.

- Topic: This is a graph that connects tweets that use the same hashtag. In other words,  $T_{ij} = 1$  if tweets  $i$  and  $j$  share a hashtag.
- Sentimental Similarity: This graph connects tweets based on their sentiments to the most comparable tweets. SentiStrength [32] is used to determine the sentiment of tweets. It is a tool that was developed using a machine-learning-enhanced human-annotated sentiment lexicon. SentiStrength assigns each tweet a positive and negative sentiment score. The Cosine similarity between the sentiment vectors of tweets is calculated, and then the tweets with the highest similarity are connected. In other words, if tweet  $j$  is one of the  $n$  most similar tweets to tweet  $i$ , or vice versa, then  $SS_{ij} = 1$ .
- Text Similarity: As with the topic graph, text similarity can be used to connect tweets that are semantically related. We represent tweets using universal sentence encoder (USE), which embeds each tweet in 512-dimensional vector space [16]. The graph is constructed by repeatedly selecting the most similar tweets based on their cosine similarity. In other words,  $USE_{ij} = 1$  if tweet  $j$  is one of the  $n$  most closely related tweets to tweet  $i$ .

Table 1 presents the statistics for the graphs used in the experiments. User graph is the most sparsely connected, followed by Sentimental Similarity and USE. The topic graph is the most densed graph.

### Experiments

This section summarizes the experimental findings.

Table 2 shows the smoothness of the sentiment signal for each graph. The smoothness of a graph signal, in particular, indicates the degree to which a signal’s node values are related to their neighbors’ corresponding values. As a result, we use this concept to investigate the correlation between the structure of these graphs and the sentiment labels. The following defines the smoothness of a signal on a weighted undirected graph [75]:

**Table 1** Statistics of the used graphs

Graph	Number of edges	Average node degree	min degree of nodes	max degree of nodes
User	2591	3.86	0	34
Social	67440	100.5	1	564
Common friends	531041	791.41	1	1248
Topic	197240	293.94	0	1341
Sentimental Similarity	6710	9.86	5	539
USE	6710	8.88	5	43

**Table 2** The smoothness of the sentiment signal with respect to different graphs

Graph	GS	RS	ND
User	549	1248	0.560096
User + Social	14568	26471	0.449662
Social	14019	25326	0.446458
Sentimental Similarity	2352	2532	0.07109
USE	3959	4222	0.062293
Topic	80962	84821	0.045496
Common friends	181390	155995	-0.16279

$$\text{Smoothness} = S^T L S = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n A_{ij} (S_i - S_j)^2, \quad (15)$$

where  $S$  denotes the signal,  $L$  is the Laplacian of the underlying graph, and  $A_{ij}$  is the weight between node  $i$  and  $j$ . Lower smoothness values indicate that the underlying graph is more closely related to the signal. We provide the smoothness of a graph with the same number of nodes and edges but randomly chosen edges in the third column. The fourth column contains the normalized difference between the random signal smoothness and the graph smoothness, which is calculated as follows:

$$ND = \frac{RS - GS}{RS}, \quad (16)$$

where  $RS$  denotes the random signal smoothness, and  $GS$  is the graph smoothness. Increased  $ND$  values imply a stronger correlation between the sentiment signal and the graph to which it is associated. Table 2 demonstrates that the sentiment signal is smoother than the random graph for all graphs except the common-friends graph. Additionally, the fourth column demonstrates that the sentiment signal is smoother when applied to the user graph, followed by the hybrid User + Social graph and the social graph, which is consistent with sentiment consistency and homophily theories.

Table 3 shows all the hyperparameters used in different models. The values have been tuned on the validation set.

The results of the baseline models are summarized in Table 4. Except for SANTI and SASS, which also use graph regularizations, these methods rely entirely on textual data. In general, sentiment analysis employs three approaches: lexicon-based [76], machine learning-based [22], and hybrid approach [77]. The first baseline is SentiStrength [32] which assigns a positivity score of (1:5) and a negativity score of 1 (− 1: − 5). In each case, the greater absolute value indicates a more sentimental text in that direction. The sum of these scores determines a tweet's classification. Because the data set contains only positive and negative tweets, when the result is 0, the tweet is classified randomly (proportional to the ratio of classes). As can be seen, this method achieves

**Table 3** Hyperparameters of the models

Model	Learning rate	Hidden layers nodes	Alpha
Lasso	0.01	-	-
Logistic regression	0.005	-	-
Feed-forward neural network	0.05	16	-
SANTI	0.005	-	0.01
SASS	0.003	-	0.05
GCN-Kipf (User)	0.03	24	-
ChebNet (User)	0.0007	17	-
GCN-Kipf (Social)	0.01	32	-
ChebNet (Social)	0.05	16	-
GCN-Kipf (Sentimental Similarity)	0.001	17	-
ChebNet (Sentimental Similarity)	0.05	16	-
GCN-Kipf (Topic)	0.001	17	-
ChebNet (Topic)	0.03	24	-
GCN-Kipf (USE)	0.002	22	-
ChebNet (USE)	0.03	32	-
GCN-Kipf (Common friends)	0.001	17	-
ChebNet (Common friends)	0.05	21	-
Stacking model	0.01	5	-

a level of accuracy of ~65%. It is worth noting that this method is unaware of the data distribution. Lasso improves this baseline to 73.60. Logistic regression outperforms other techniques. A slightly improved result is obtained by adding a hidden layer (16 units) to the logistic regression and constructing a feed-forward neural network. The results achieved using graph regularization methods are comparable to those using other baselines (except for SentiStrength), where SANTI [6] achieves a ~75% accuracy, and SASS [5] achieves a ~78% accuracy.

Table 5 presents the experimental results for the graph convolutional models. Convolutional graph models outperform textual models and regularization methods for graphs. The findings suggest that leveraging tweet relationships via GCNs (rather than the graph regularization method) is more advantageous. Additionally, ChebNet outperforms GCN-Kipf in most cases, which is likely because GCN-Kipf is a special case of ChebNet. Due to sentiment

**Table 4** Experimental results of the baseline models

Model	Accuracy
SentiStrength	65.42
Lasso	73.60
Logistic regression	77.32
Feed-forward neural network	78.06
SANTI	75.09
SASS	78.00

**Table 5** Experimental results of GCN-Kipf and ChebNet using different graphs and the stacking model

Model	GCN-Kipf	ChebNet
User	76.95	81.04
Social	81.41	79.18
Sentimental Similarity	75.09	81.04
Social + User	80.66	80.29
Topic	75.09	78.43
USE	75.83	76.20
Common friends	75.09	78.43
Stacking	82.15	

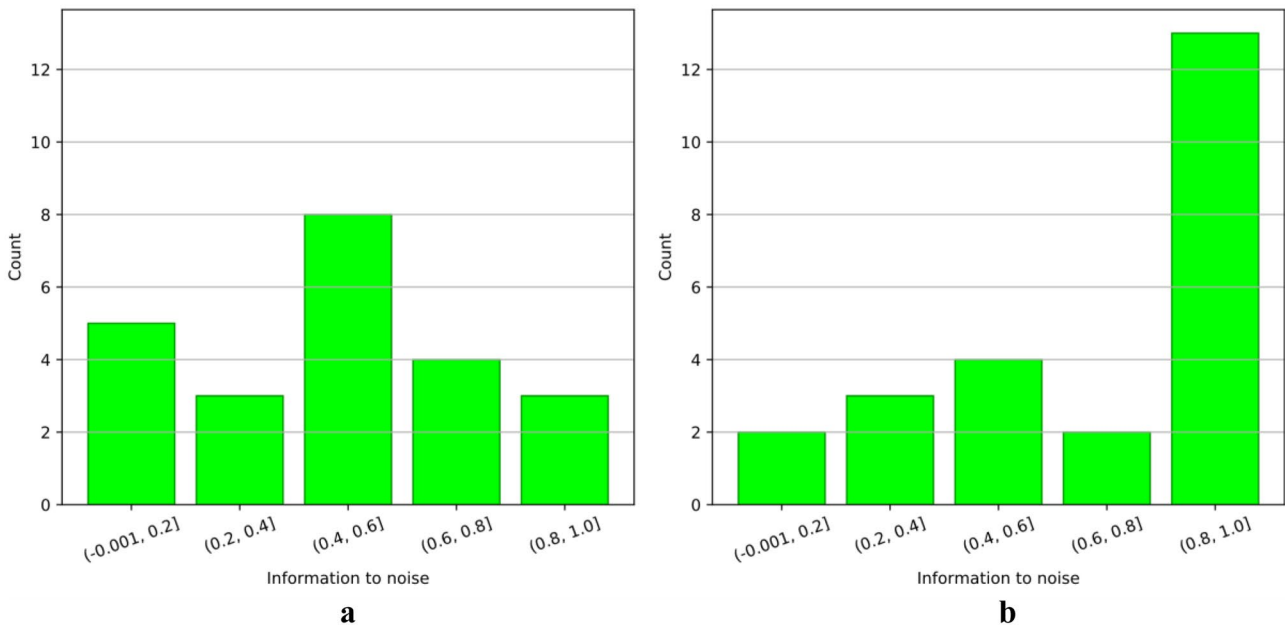
contagion and existence many social edges, the social graph achieves the best result.

Furthermore, the User and Sentimental Similarity graphs perform well (in the ChebNet model) due to the consistency and the knowledge injected into the model by the lexicon. The primary reason the social graph outperforms the user graph is due to the former’s more significant number of edges. Previous research with graph regularization has produced comparable results [5].

In comparison to other graphs, the topic, USE, and common-friends graphs all produce unsatisfactory results. Despite its dense nature, the topic graph is not particularly useful. Perhaps this is because when confronted with a particular subject, individuals can experience a range of emotions. For instance, some people regard #hcr positively, while others do not. This situation is exacerbated in the USE

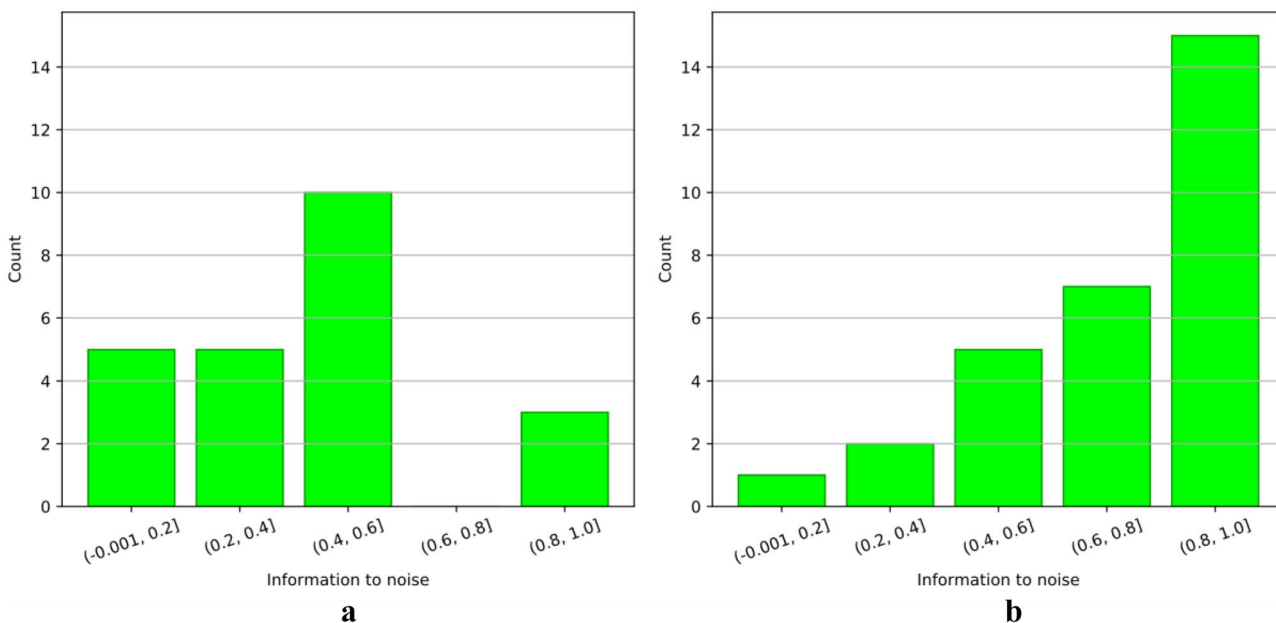
graph, where the results indicate that text similarity alone may not be sufficient to distinguish sentiment classes, as two identical texts one containing only one negative word can have the opposite sentiment. The common-friends graph is not as precise as the direct friendship graph. This finding could imply that individuals share similar views with their friends but are less likely to share them with their friends’ friends. As illustrated in Table 5, the proposed stacking model outperformed all other models in the experiments due to the model’s use of two concurrent graphs, which reduces the model’s error rate.

Finally, to better understand the GCN models’ behavior, we calculate the information-to-noise ratio, which is essentially the ratio of neighbors with similar labels to the target node to all neighbors. It has been demonstrated that the success of GNNs is because the information received from neighbors is greater than the noise received [78]. As a result, we examine the relationship between prediction accuracy and the information-to-noise ratio of nodes in the user and social graphs. Figure 3 depicts the obtained results for the User GCN graph. The chart on the left illustrates the histogram of the information-to-noise ratio for nodes in which the feed-forward (context-free) model is successful, but the ChebNet (User) (contextual) model fails. The chart on the right depicts the opposite situation, in which the User GCN model succeeds, but the feed-forward model fails. This figure shows no clear correlation between the information-to-noise ratio and feed-forward model performance, whereas ChebNet (User) performs significantly better in nodes with a higher information-to-noise ratio. This is due to the GCN



**Fig. 3** Performance evaluation of the average information-to-noise ratio: **a** nodes where the feed-forward model is successful, but ChebNet (User) fails, and **b** nodes where ChebNet (User) succeeds but feed-forward fails



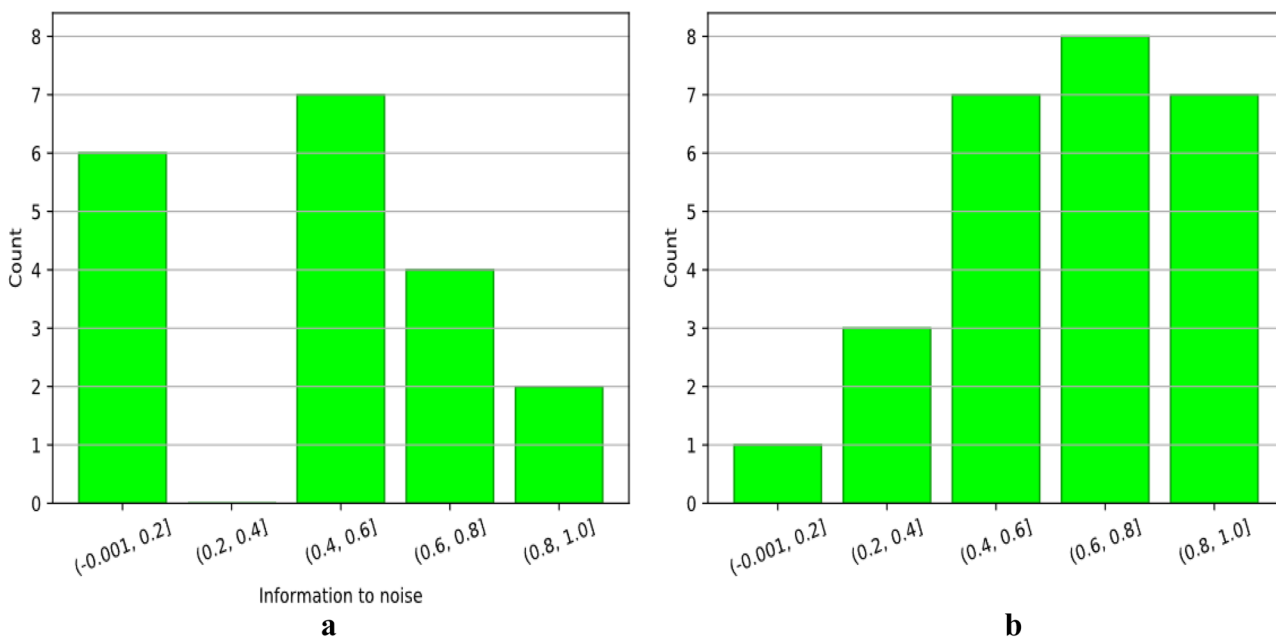


**Fig. 4** Performance evaluation of the average information-to-noise ratio: **a** nodes in which the feed-forward model succeeds but GCN-Kipf (Social) fails, and **b** nodes where GCN-Kipf (Social) is successful but feed-forward fails

model’s nature. When neighbors’ features are more similar to those of the target node, GCN provides more useful information for the target node’s contextual representation, resulting in improved performance on the target node.

Similarly, Fig. 4 depicts the information-to-noise of the Social GCN model. The chart on the left depicts the histogram of the information-to-noise ratio for nodes where the feed-forward model succeeds but the Social GCN model

fails, while the chart on the right depicts the opposite. Once again, the GCN model performs better with smoother context tweets. The average information-to-noise ratio of the user and social graphs in the feed-forward model (left) and stacking model (right) are depicted in Fig. 5. The figure demonstrates that the stacking model performs better with samples with a higher information-to-noise ratio, contrary to what the context-free model predicts. GCN may receive data



**Fig. 5** Performance evaluation of the mean of the average of information-to-noise ratio: **a** nodes where the feed-forward model succeeds but stacking model fails, and **b** nodes where stacking model is successful but feed-forward fails

or noise from other nodes. Aggregating nodes that belong to the same class provides information by bringing their representations closer together and increasing the likelihood that they assign to the same class. On the contrary, noise is introduced due to the aggregation of nodes belonging to other classes.

## Conclusion

The purpose of this study was to perform sentiment analysis utilizing tweet contexts. Previously, this procedure was carried out using graph regularization [6]. GCNs were chosen for sentiment analysis on Twitter due to recent advances in graph signal processing [79] which enable a more accurate application of deep learning on graphs. We analyzed contextualized Twitter sentiment using GCNs. The stacking of contextual graphs, including user and friendship, was investigated using two distinct types of GNNs, ChebNet [74] and GCN-Kipf [19]. We obtained promising experimental results on a real-world Twitter sentiment analysis dataset, outperforming both text- and graph-based models. Additional research is required to examine other GNN models including models that utilize multiple graphs end-to-end. Additionally, as the paper's final section implies, GCNs rely on a high information-to-noise ratio, which indicates that their performance on low-information-to-noise nodes may be suboptimal. Sentiment classifiers that make better use of context are promising candidate models for future research.

## Declarations

**Ethics Approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Conflict of Interest** The authors declare no competing interests.

## References

- Cambria E, Poria S, Gelbukh A, Thelwall M. Sentiment analysis is a big suitcase. *IEEE Intell Syst*. 2017;32(6):74–80.
- Li D, Wang Y, Madden A, Ding Y, Tang J, Sun GG, et al. Analyzing stock market trends using social media user moods and social influence. *J Assoc Inf Sci Technol*. 2019;70(9):1000–13.
- Keramatfar A, Amirkhani H. Bibliometrics of sentiment analysis literature. *J Inf Sci*. 2019;45(1):3–15.
- Hussain A, Cambria E, Poria S, Hawalah AYA, Herrera F. Information fusion for affective computing and sentiment analysis. *Inf Fusion*. 2021;71:97–8.
- Zou X, Yang J, Zhang J. Microblog sentiment analysis using social and topic context. *PLoS one*. 2018;13(2):e0191163.
- Hu X, Tang L, Tang J, Liu H. Exploiting social relations for sentiment analysis in microblogging. In: *Proceedings of the sixth ACM international conference on Web search and data mining*. Rome, Italy; 2013. p. 537–46.
- Sánchez-Rada JF, Iglesias CA. Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison. *Inf Fusion*. 2019;52:344–56.
- Feng S, Wang Y, Liu L, Wang D, Yu G. Attention based hierarchical LSTM network for context-aware microblog sentiment classification. *World Wide Web*. 2019;22(1):59–81.
- Abelson RP. Whatever became of consistency theory? *Pers Soc Psychol Bull*. 1983;9(1):37–54.
- McPherson M, Smith-Lovin L, Cook JM. Birds of a feather: homophily in social networks. *Annu Rev Sociol*. 2001;27(1):415–44.
- Bollen J, Gonçalves B, Ruan G, Mao H. Happiness is assortative in online social networks. *Artif Life*. 2011;17(3):237–51.
- Hatfield E, Cacioppo JT, Rapson RL. Emotional contagion. *Curr Dir Psychol Sci*. 1993;2(3):96–100.
- Kramer ADI, Guillory JE, Hancock JT. Experimental evidence of massive-scale emotional contagion through social networks. *Proc Natl Acad Sci*. 2014;111(24):8788–90.
- Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations*. Arizona, USA; 2013. p. 1–12.
- Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha, Qatar; 2014. p. 1532–43.
- Cer D, Yang Y, Kong S-Y, Hua N, Limtiaco N, St. John R, et al. Universal sentence encoder for English. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics; 2018. p. 169–74.
- Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, USA: Association for Computing Machinery; 2014. p. 701–10.
- Hou Y, Zhang J, Cheng J, Ma K, Ma RT, Chen H, et al. Measuring and improving the use of graph information in graph neural networks. In: *International Conference on Learning Representations*. New Orleans, USA; 2019. p. 1–16.
- Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations*. Toulon, France; 2017. p. 1–14.
- Zhang M, Chen Y. Link prediction based on graph neural networks. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Montréal, Canada; 2018. p. 5171–81.
- Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: *The Semantic Web*. Monterey, USA. 2018. p. 593–607.
- Keramatfar A, Amirkhani H, Jalaly Bidgoly A. Multi-thread hierarchical deep model for context-aware sentiment analysis. *J Inf Sci*. 2021;1–12.
- Shoeb M, Ahmed J. Sentiment analysis and classification of tweets using data mining. *Int Res J Eng Technol*. 2017;4(12):1471–4.
- Khatua A, Khatua A, Cambria E. Predicting political sentiments of voters from Twitter in multi-party contexts. *Appl Soft Comput*. 2020;97:106743.
- Sarkar K. A stacked ensemble approach to Bengali sentiment analysis. In: *Intelligent Human Computer Interaction*. Daegu, Korea; 2020. p. 102–11.
- Rani S. Hybrid model using stack-based ensemble classifier and dictionary classifier to improve classification accuracy of Twitter sentiment analysis. *Int J Emerg Trends Eng Res*. 2020;8(7):2893–900.
- Taboada M, Brooke J, Tofigoski M, Voll K, Stede M. Lexicon-based methods for sentiment analysis. *Comput Linguist*. 2011;37(2):267–307.
- Stone PJ, Hunt EB. A computer approach to content analysis: studies using the General Inquirer system. In: *Proceedings of the*

- May 21–23, 1963 spring joint computer conference. Detroit, USA: Association for Computing Machinery; 1963. p. 241–56.
29. Baccianella S, Esuli A, Sebastiani F. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: European Language Resources Association (ELRA). 2010. p. 2200–4.
  30. Strapparava C, Valitutti A. Wordnet affect: an affective extension of wordnet. Lisbon, Portugal: Lrec; 2004. p. 1083–1086.
  31. Cambria E, Li Y, Xing FZ, Poria S, Kwok K. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. 2020. p. 105–14.
  32. Thelwall M, Buckley K, Paltoglou G, Cai D, Kappas A. Sentiment strength detection in short informal text. *J Am Soc Inf Sci Technol*. 2010;61(12):2544–58.
  33. Hu M, Liu B. Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. Seattle, USA; 2004. p. 168–77.
  34. Jurek A, Mulvenna MD, Bi Y. Improved lexicon-based sentiment analysis for social media analytics. *Secur Inform*. 2015;4(1):1–13.
  35. Gupta I, Joshi N. Enhanced twitter sentiment analysis using hybrid approach and by accounting local contextual semantic. *J Intell Syst*. 2020;29(1):1611–25.
  36. Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: a survey. *WIREs Data Min Knowl Discov*. 2018;8(4):e1253.
  37. Despotovic V, Tanikic D. Sentiment analysis of microblogs using multilayer feed-forward artificial neural networks. *Comput Inform*. 2017;36(5):1127–42.
  38. Vassilev A. Bowtie-a deep learning feedforward neural network for sentiment analysis. In: International Conference on Machine Learning, Optimization, and Data Science. Cham: Springer; 2019. p. 360–71.
  39. Ma Y, Peng H, Cambria E. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: Thirty-second AAAI conference on artificial intelligence. New Orleans, USA; 2018. p. 5876–5883.
  40. Shin B, Lee T, Choi JD. Lexicon integrated CNN models with attention for sentiment analysis. In: 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. Copenhagen, Denmark; 2017. p. 149–58.
  41. Akhtar MS, Ekbal A, Cambria E. How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble. *IEEE Comput Intel Mag*. 2020;15(1):64–75.
  42. Behera RK, Jena M, Rath SK, Misra S. Co-LSTM: convolutional LSTM model for sentiment analysis in social big data. *Inf Process Manag*. 2021;58(1):102435.
  43. Merello S, Ratto AP, Oneto L, Cambria E, editors. Ensemble application of transfer learning and sample weighting for stock market prediction. In: 2019 International Joint Conference on Neural Networks (IJCNN). 2019.
  44. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego, USA; 2016. p. 1480–9.
  45. Wang Y, Huang M, Zhu X, Zhao L. Attention-based LSTM for aspect-level sentiment classification. In: Conference on empirical methods in natural language processin. g2016. p. 606–15.
  46. Kardakis S, Perikos I, Grivokostopoulou F, Hatzilygeroudis I. Examining attention mechanisms in deep learning models for sentiment analysis. *Appl Sci*. 2021;11(9):1–14.
  47. Liu Q, Zhang H, Zeng Y, Huang Z, Wu Z. Content attention model for aspect based sentiment analysis. In: Proceedings of the 2018 World Wide Web Conference. Lyon, France; 2018. p. 1023–32.
  48. Maas A, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. Portland, USA; 2011. p. 142–50.
  49. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B. Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014. p. 1555–65.
  50. Li Y, Pan Q, Yang T, Wang S, Tang J, Cambria E. Learning word representations for sentiment analysis. *Cognit Comput*. 2017;9(6):843–51.
  51. Naderalvojud B, Sezer EA. Sentiment aware word embeddings using refinement and senti-contextualized learning approach. *Neurocomputing*. 2020;405:149–60.
  52. Poria S, Majumder N, Hazarika D, Cambria E, Gelbukh A, Hussain A. Multimodal sentiment analysis: addressing key issues and setting up the baselines. *IEEE Intell Syst*. 2018;33(6):17–25.
  53. Karimvand AN, Chegeni RS, Basiri ME, Nemati S. Sentiment analysis of Persian instagram post: a multimodal deep learning approach. In: 7th International Conference on Web Research (ICWR). Tehran, Iran; 2021. p. 137–41.
  54. Peng W, Hong X, Zhao G. Adaptive modality distillation for separable multimodal sentiment analysis. *IEEE Intell Syst*. 2021;36(3):82–9.
  55. Li W, Zhu L, Cambria E. Taylor’s theorem: a new perspective for neural tensor networks. *Knowl Based Syst*. 2021;228:107258.
  56. Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. Santiago, Chile; 2015. p. 959–62.
  57. Cliche M. BB\_twttr at SemEval-2017 Task 4: Twitter sentiment analysis with CNNs and LSTMs. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics; 2017. p. 573–80.
  58. Barnes J, Velldal E, Øvrelid L. Improving sentiment analysis with multi-task learning of negation. *Nat Lang Eng*. 2021;27(2):249–69.
  59. Majumder N, Poria S, Peng H, Chhaya N, Cambria E, Gelbukh A. Sentiment and sarcasm classification with multitask learning. *IEEE Intell Syst*. 2019;34(3):38–43.
  60. Yang Li, Amirmohammad Kazameini, Yash Mehta, Cambria E. Multitask learning for emotion and personality detection. *IEEE Trans Affect Comput*. 2021;1(1):1–8.
  61. Dashitpour K, Gogate M, Li J, Jiang F, Kong B, Hussain A. A hybrid Persian sentiment analysis framework: integrating dependency grammar based rules and deep neural networks. *Neurocomputing*. 2019;380:1–10.
  62. Zhao P, Hou L, Wu O. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowl Based Syst*. 2020;193:105443.
  63. Jin Z, Zhao X, Liu Y. Heterogeneous graph network embedding for sentiment analysis on social media. *Cognit Comput*. 2021;13(1):81–95.
  64. Zhou J, Huang JX, Hu QV, He L. SK-GCN: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. *Knowl Based Syst*. 2020;205:106292.
  65. Zhu X, Zhu L, Guo J, Liang S, Dietze S. GL-GCN: Global and local dependency guided graph convolutional networks for aspect-based sentiment classification. *Expert Syst Appl*. 2021;186:115712.
  66. Liao W, Zeng B, Liu J, Wei P, Cheng X, Zhang W. Multi-level graph neural network for text sentiment analysis. *Comput Electr Eng*. 2021;92:107096.
  67. Lei J, Zhang Q, Wang J, Luo H. BERT based hierarchical sequence classification for context-aware microblog sentiment

- analysis. In: International Conference on Neural Information Processing. 2019. p. 376–86.
68. Sairamya NJ, Susmitha L, Thomas George S, Subathra MSP. Chapter 12 - Hybrid approach for classification of electroencephalographic signals using time–frequency images with wavelets and texture features. In: Hemanth DJ, Gupta D, Balas VE, editors. *Intelligent Data Analysis for Biomedical Applications*. Cambridge: Academic Press; 2019. p. 253–73.
69. Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge, USA: MIT press; 2016.
70. Eickenberg M, Gramfort A, Varoquaux G, Thirion B. Seeing it all: Convolutional network layers map the function of the human visual system. *Neuroimage*. 2017;152(1):184–94.
71. Kuzovkin I, Vicente R, Petton M, Lachaux J-P, Baciú M, Kahane P, et al. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Commun Biol*. 2018;1(1):107.
72. Zhang S, Tong H, Xu J, Maciejewski R. Graph convolutional networks: a comprehensive review. *Comput Soc Netw*. 2019;6(1):11.
73. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst*. 2019;32(1):4–24.
74. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: *Advances in neural information processing systems*. Barcelona, Spain; 2016. p. 3844–52.
75. Stanković L, Daković M, Sejdić E. Introduction to graph signal processing. In: Stanković L, Sejdić E, editors. *Vertex-Frequency Analysis of Graph Signals*. Cham: Springer International Publishing; 2019. p. 3–108.
76. Asgarian E, Kahani M, Sharifi S. HesNegar: Persian Sentiment WordNet. *Signal Data Process*. 2018;15(1):71–86.
77. Mudinas A, Zhang D, Levene M. Combining lexicon and learning based approaches for concept-level sentiment analysis. In: *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*. Beijing, China; 2012. p. 1–8.
78. Chen D, Lin Y, Li W, Li P, Zhou J, Sun X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, USA; 2020. p. 3438–45.
79. Shuman D, Narang S, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process Mag*. 2013;30(3):83–98.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.