
SENTIMENT ANALYSIS OF TWEETS USING TEXT AND GRAPH MULTI-VIEWS LEARNING

Loitongbam Gyanendro Singh

School of Electronics and Computer Science
University of Southampton
Southampton, United Kingdom
gyanendro.loitongbam@soton.ac.uk

Sanasam Ranbir Singh

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
North Guwahati, Assam, India
ranbir@iitg.ac.in

ABSTRACT

With the surge of deep learning framework, various studies have attempted to address the challenges of sentiment analysis of tweets (data sparsity, under-specificity, noise, and multilingual content) through text and network-based representation learning approaches. However, limited studies on combining the benefits of textual and structural (graph) representations for sentiment analysis of tweets have been carried out. This study proposes a multi-view learning framework (*end-to-end* and *ensemble-based*) that leverages both text-based and graph-based representation learning approaches to enrich the tweet representation for sentiment classification. The efficacy of the proposed framework is evaluated over three datasets using suitable baseline counterparts. From various experimental studies, it is observed that combining both textual and structural views can achieve better performance of sentiment classification tasks than its counterparts.

Keywords Sentiment analysis, Multi-view learning, Sequence learning model, Graph neural network.

1 Introduction

Given the surging popularity of Twitter, tweets have gained significant traction as subjects for sentiment analysis studies in recent times. Unlike regular text, sentiment analysis of tweets needs to handle inherent challenges like under-specificity due to limited characters (240 characters), informal writing styles, misspellings, code-switching, code-mixing contents, etc. Researchers have adopted various approaches such as sentiment-specific representation learning [1], tweet expansion [2], users relationship characteristics [3], multi-source information [4], ensembling [5], etc. to mitigate the above challenges. Earlier studies of sentiment analysis primarily focus on textual views; however, recent studies have shown the advantages of exploiting network embedding for sentiment analysis of tweets [2, 6, 7, 8]. In the studies [2, 6], authors construct a global network from a set of tweet corpus and learn the representation of required attributes, such as keywords, hashtags, users, etc., for sentiment classification. Whereas, studies [7, 8] construct a local network using a dependency parse tree of the individual tweet and learn a tweet representation for further classification. The above studies have shown that capturing structural information helps in enhancing sentiment analysis performance. It is also reported that network embedding is less sensitive to the social media-related noise mentioned above.

Though the above studies have shown the prospect of enhancing the performance of sentiment classification of tweets by incorporating both graphical and textual views, the studies have a few limitations. The studies of Singh et al. [2] and Lu et al. [6] have exclusively performed node embedding over the tweet corpus. The node embedding is subsequently incorporated into the individual textual view to capture the tweet's sequential information (i.e., textual views). These

⁰This paper has been accepted for publication in the *Knowledge and Information Systems* journal.

⁰This work is licensed under a Creative Commons Attribution 4.0 International License.

Tweet:Historic day for the Nation, #GST bill passed in Lok Sabha. #Congratulations to the nation, salute 2the vision of #PM @narendramodi ji

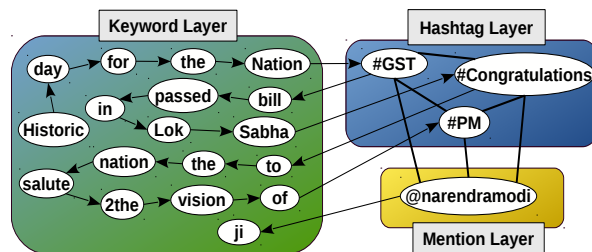


Figure 1: An example of representing a tweet to a heterogeneous multi-layer network structure

approaches do not consider incorporating the textual and structural views alongside. On the other hand, Meng et al. [7] and Zhang et al. [8] investigate using a dependency parse tree to represent a tweet in a network structure. However, adopting a dependency parse tree to represent a tweet may not always be feasible in the case of multilingual content (code switch and code mix) and informal textual constructs. In such a case, it is required to represent a tweet in a language-insensitive graph structure that captures the semantic relationship of the words in incorporating both structural and textual information. It is also observed that these studies combine textual and structural views progressively, where the representation of the words from text views is passed on to the structural views to represent the nodes (words). These approaches have not considered the notion of complementing both views simultaneously in parallel.

Motivated by the above observations (i.e., advantages of capturing structural information in the tweet, advantages of using network embedding), this paper proposes a multi-view based neural model to exploit both the textual and structural properties in parallel for an improved sentiment analysis system and attempts to understand two research questions – (i) *How informative is a graph-based representation of a tweet compared to text-based representation?* (ii) *Does the text-based representation, and graph-based representation complement each other?* This study considers using a heterogeneous multi-layer network proposed in the study of Singh et al. [2] to represent a tweet. A multi-layer network is a network formed by connecting different layers of networks. For example, a tweet or a collection of tweets can be represented as a heterogeneous multi-layer network by connecting layers of mention relations, hashtag relations, and keyword co-occurrence relations. Figure 1 shows an example of representing a tweet to a heterogeneous multi-layer network. Since the heterogeneous multi-layer network exploits co-occurrence characteristics rather than the linguistic structure, it makes the heterogeneous multi-layer network less sensitive to social media-related multilingual noise [2].

The primary distinction between the current research work to the study conducted by Singh et al. [2] lies in how the heterogeneous graph is portrayed within the latent embedding space (see Section 3.2). This study can be viewed as a further investigation of Singh et al.’s study [2], where the graph-based representation is learned using neural network-based approaches instead of random-walker-based methods. We study the synergy between graph-based and text-based representations, exploring their potential mutual enhancement. Furthermore, we investigate whether the heterogeneous graph captures the underlying linguistic information comparably to the graph formed using a language-specific dependency parser. Notably, constructing a language-specific dependency parser becomes complicated in the presence of multilingual text. Therefore, this study further explores whether the tweet graph should adopt a language-dependent or language-insensitive heterogeneous graph approach (refer to Section 3.2).

The proposed study represents two types of tweet views: *textual views* and *graphical views* and generates the embedding representation for each view using an appropriate embedding method. In this study, we use Convolution Neural Network (CNN) [9] and Bidirectional Encoder Representations from Transformers (BERT) [10] based representation learning for textual view and Deep Graph CNN (DGCNN) [11] and Segmented-Graph BERT (Seg-BERT) [12] for graphical view. The representations thus obtained are then integrated using an attention-based aggregator. The efficacy of the proposed model is evaluated using suitable baseline counterparts. From various experimental setups over three datasets, it is evident that the proposed multi-view model performs better sentiment analysis than

its single-view counterparts. Further, it is also observed that the proposed model is less sensitive to under-specificity, noise, and multi-lingual content. In summary, this paper has the following contributions:

- Propose a multi-view learning framework incorporating tweet text and graph views.
- Evaluate the performance of the proposed sentiment analysis framework compared to graph-based and text-based representations of tweets.
- Investigates whether the tweet graph is necessarily a language-dependent or language-insensitive heterogeneous graph.

The remainder of the paper is organized as follows. In Section 2, the literature related to this study is presented. Section 3 presents the proposed investigation study. The experimental setup is described in Section 4. The results and observations are analyzed in Section 5. Finally, the study of this paper concludes in Section 6.

2 Related studies

Sentiment analysis, in general, has experienced significant development as a result of cutting-edge methods that help us better comprehend the subtleties of context and textual emotions [13, 14, 15, 16, 17, 18, 19, 20, 21]. These studies highlight the importance of capturing sentiment in complex textual opinions. The multi-level neural network technique uses graph structures to combine co-occurrence and semantic similarity graphs to account for local and global variables, improving sentiment analysis results. Studies [19, 20] have introduced models that delve into the power of graph-based attention networks, adeptly capturing syntactic complexities, semantic relations, and sentiment polarities tied to specific aspects. Notably, several studies [19, 15, 16, 20, 13] explore the utilization of dependency parsers to construct graph structures of opinions, enabling a deeper understanding of the relationships between opinion words, aspects, and context. Various studies [14, 15, 18, 21] propose multifaceted solutions for more accurate aspect-level sentiment analysis. To improve sentiment prediction performance, these models combine diverse graph structures, use aspect-aware attention processes, include contextual and emotional knowledge, and use the interactions between various elements. In this collective effort, researchers highlight the value of incorporating structural and contextual information, affective knowledge, and advanced graph-based techniques to advance the understanding and prediction of sentiment expressions across diverse domains.

Several studies have explored the fusion of textual and graph-based perspectives for sentiment analysis tasks. This section provides a concise overview of the relevant literature that aligns with the proposed study’s objectives. Nguyen et al. [17] demonstrate the ability of multi-layered networks and Graph Neural Networks (GNNs) to understand sentiment in social media texts better by bridging the gap in social media text analysis, hence enhancing sentiment analysis approaches. Recent investigations have employed graph representation-based methodologies with text-based representations for aspect-based sentiment analysis tasks. Significant among them are studies by Chen et al.[22], Zhang et al.[8], and Meng et al. [7]. These scholars have embraced the Graph Convolutional Network (GCN) for learning the node features in the aspect-based sentiment classification tasks by transforming the opinionated text into a tree using a dependency parser of the English language. In the work of Zhang et al.[8], a GCN is applied to the dependency tree, with node features generated by a Long Short-Term Memory (LSTM) model to capture contextual nuances. Aspect-specific features are extracted through selective masking of the GCN output, followed by attention mechanisms for sentiment classification. A similar trajectory is pursued by Meng et al.[7], who leverage BERT embeddings to acquire contextual node attributes for GCN. Conversely, Chen et al. [22] embrace a multi-view learning framework for aspect-based sentiment classification. Their approach melds GCN over the dependency tree and LSTM over the word sequence, harmonizing the two streams through concatenation to enable aspect-based sentiment classification. The difference between the studies mentioned above and our proposed research lies in the application of network representation. The cited works revolve around using a dependency parser to map input text into a graph structure structurally. However, given the highly multilingual nature of tweets, relying solely on dependency trees becomes untenable. Unlike the previous study paradigms, this study examines the potential of harnessing the interconnections among hashtags, mentions, and conventional tokens in tweets. These connections are treated as constituent layers in a heterogeneous multi-layer network, driving sentiment classification across the entire tweet landscape rather than limiting the scope to aspect-based sentiment analysis.

In a different direction but related, Lu et al. [6] consider GCN and BERT to generate the word embeddings. Their study considers vocabulary graphs to generate node embedding using GCN

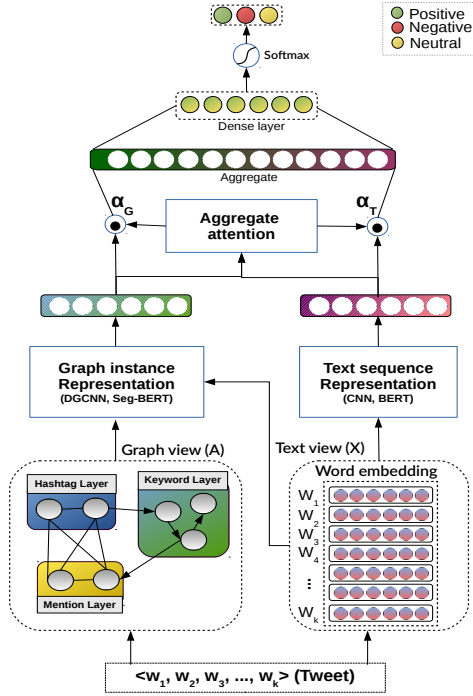


Figure 2: Proposed framework for sentiment classification of tweet by incorporating text and graph views through text and graph representation models. A and X represent the word embedding and adjacency matrices of the input tweet, and α_i represent the weighted representation of the graph (G) and text (T) representations

and pre-trained BERT embedding for text-based representation. The two-word embeddings are concatenated to generate the sentence representation via multi-head attention over the input word embeddings for the underlying sentiment classification task. Their approach does not complement the representation of text and graph views. Yao et al. [23] perform text classification using GCN by representing the text corpus to a heterogeneous network with the document as one type of node and the informative keywords connecting them. Their study applies GCN over the single structure, which requires the training and testing document to be present in the heterogeneous graph for generating the representation of the document. Hence, it is not feasible to represent unseen tweets for the sentiment classification task. Unlike the above studies, this paper represents tweets for sentiment classification by simultaneously learning text and graph view representation.

In contrast to the above studies, our study uses a language-insensitive graph, which is a significant difference. Many researchers have adopted dependency parsers to build the word graph for sentiment analysis tasks. However, considering dependency parse trees for tweet representation is language sensitive and has issues with informal text constructions and multilingual text (such as code-switching and code-mixing). In such a scenario, it is essential to have a language-insensitive graph structure that can capture word associations in their semantic context while combining structural and textual knowledge. Amid these changes, our work aims to provide a distinctive viewpoint on sentiment analysis, addressing issues with multilingual and under-specificity challenges via language-insensitive graphs and improving sentiment analysis approaches.

3 Proposed study

Given a tweet T with n words $(w_1, w_2, w_3, \dots, w_n)$, the objective of this paper is to incorporate semantic relation of words represented in different views (textual and graph) through a multi-view representation model. The text-view is represented using text embedding methods such as CNN and BERT. The graph-view is represented using graph embedding methods such as DGCNN and Seg-BERT. Figure 2 shows a high-level architecture of the proposed framework.

In the remaining part of this section, italic lowercases (e.g., w_i , s), bold lowercases (e.g. \mathbf{x}_i , \mathbf{h}), and bold uppercases (e.g. \mathbf{W}) are used to denote scalars, vectors, and matrices respectively. A tweet T is represented in text-view as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ where \mathbf{X}_i (i^{th} row of the matrix \mathbf{X}) represents the embedding of the word w_i of dimension d . This study considers FastText embedding [24] to generate the initial semantic word embeddings. However, the proposed framework can be applied to any word or node embedding method. The semantics of the word sequence relations are captured using a text representation model F_{seq} that transforms the text-view \mathbf{X} to a vector \mathbf{z}_{seq} , i.e., $\mathbf{z}_{seq} = F_{seq}(\mathbf{X}, \theta_{seq})$ where θ_{seq} is the model learning parameter. Since hashtags and mentions are added by the author of the tweet, capturing the relation of hashtags, mentions, and normal tokens will be of great interest as hashtags and mentions can link tweets to similar topics or themes. To capture the semantic relations of the words, the tweet T is represented in graph-view as a heterogeneous multi-layer graph via an adjacency matrix representation $\mathbf{A}_{n \times n}$ to accommodate the relation of hashtags, mentions, and normal keywords present in the tweet. The process of representing T to the heterogeneous multi-layer graph is discussed in Section 3.2. The semantics of the relations of words are captured using graph instance representation learning model F_{graph} that transformed $\mathbf{A}_{n \times n}$ to a vector \mathbf{z}_{graph} using its corresponding word embedding \mathbf{X} as nodes features, i.e., $\mathbf{z}_{graph} = F_{graph}(\mathbf{A}, \mathbf{X}, \theta_{graph})$ where θ_{graph} is the model learning parameter. This study exploits CNN and BERT models as the text representation model (F_{seq}) for capturing the local semantics of tweets. While DGCNN and Seg-BERT models are considered as the graph representation model (F_{graph}) to capture the semantic relations of the tokens in tweets. The text and graph representation models considered in this study are further discussed in Sections 3.1 and 3.2.

Given a text-view representation \mathbf{z}_{seq} and graph-view representation \mathbf{z}_{graph} of a tweet T , the two views are integrated using the *Scaled Dot-Product Attention* mechanism [25]. Given a query tweet, the idea is to assign attention weights to text-view and graph-view over the input query as the average of both representations. The purpose of the attention mechanism is to capture the right amount of information for each view to represent the input tweet. We define the query of the attention by an element-wise average of the \mathbf{z}_{seq} and \mathbf{z}_{graph} representations, i.e.,

$$\mathbf{z}_{avg}[i] = \frac{\mathbf{z}_{seq}[i] + \mathbf{z}_{graph}[i]}{2} \quad (1)$$

The attention weight vector of the text view is defined as:

$$\alpha_{seq} = \text{Softmax}\left(\frac{\mathbf{z}_{avg} \cdot \mathbf{z}_{seq}^T}{\sqrt{|\mathbf{z}_{avg}|}}\right) \quad (2)$$

Similarly, the attention weight vector of the graph view is defined as:

$$\alpha_{graph} = \text{Softmax}\left(\frac{\mathbf{z}_{avg} \cdot \mathbf{z}_{graph}^T}{\sqrt{|\mathbf{z}_{avg}|}}\right) \quad (3)$$

The text-based and graph-based representations generated using the above methods can be incorporated in an *end-to-end* or *ensemble* fashion to generate the final representation of the tweet for the sentiment classification task. In the *end-to-end* framework, as the name suggested, both the text and graph representation methods are learned together for the tweet classification task. While in *ensemble* framework, the text and graph representation methods are learned individually and fused in parallel to generate the final tweet representation. The two views are integrated by concatenating the weighted representation of each view. The weighted representations help select the informative representation of each view that supports generating the final representation of the tweet. The concatenation of the two views can be defined as follows:

$$\mathbf{z}_{agg} = \alpha_{seq} \cdot \mathbf{z}_{seq} \oplus \alpha_{graph} \cdot \mathbf{z}_{graph} \quad (4)$$

The sentiment classifier is trained using a dense layer with the Relu activation function. It can be mathematically defined as:

$$\mathbf{s} = \text{Softmax}(\text{Relu}(\mathbf{W} \cdot \mathbf{z}_{agg} + \mathbf{b})) \quad (5)$$

where \mathbf{W} and \mathbf{b} and weight and bias parameters of the dense layer. We use *Categorical Cross-Entropy* loss function defined in Equation 6 and *Adam Optimizer* as the optimization technique for training the proposed framework.

$$\Delta = -\frac{1}{l} \sum_{i=1}^l \sum_c \mathbf{t}_{ic} \log(s_{ic}) \quad (6)$$

where c is the number of sentiment classes, \mathbf{t}_{ic} is the c^{th} ground truth class for the tweet, l is the total number of training samples, and s_{ic} is the predicted probability on sample i for the c^{th} class.

3.1 Text representation model

Given a tweet, a text-view can be generated using suitable text embedding methods. In this paper, we have investigated text representation using CNN [9] and BERT [10]. This section discusses CNN and BERT-based embedding briefly.

3.1.1 Convolution Neural Network

From earlier studies [2, 26], it is reported that CNN captures local semantics better than recurrent-based models for sentiment classification tasks, especially for short text. CNN aims to capture intricate linguistic patterns and hierarchical structures inherent in textual data. The convolutional filters slide over the sequence of word embeddings $\mathbf{X} \in \mathbb{R}^{n \times d}$. The i^{th} row of the matrix \mathbf{X} represents the embedding of the i^{th} word in the tweet. Each filter f focuses on a specific sequence of adjacent words, allowing the network to capture different levels of linguistic granularity. To capture spatial properties of h consecutive words in the tweet, we apply convolution over the matrix \mathbf{X} considering kernels of size $h \times d$. The convolution operation at time t can be defined as:

$$conv_i^{(f)}(\mathbf{X}, h) = ReLu(\mathbf{W}^{(f)} \cdot \mathbf{X}_{t:t+h-1} + b^{(f)}) \quad (7)$$

where $\mathbf{W}^{(f)}$ is the kernel matrix for the filter f and $b^{(f)}$ is the corresponding bias. We consider padding and apply filter f with a stride size 1 to obtain a convolution vector $\mathbf{c}^{(f)}$ for the tweet matrix \mathbf{X} . The elements of $\mathbf{c}^{(f)}$ vector are defined as follow:

$$\mathbf{c}_i^{(f)} = conv_i^{(f)}(\mathbf{X}, h) \quad (8)$$

After applying *maxpooling*, we obtain a vector \mathbf{z}^f to represent the tweet using the filter f i.e.,

$$\mathbf{z}^{(f)} = maxpooling(\mathbf{c}^{(f)}) \quad (9)$$

We consider 128 filters. The 128 \mathbf{z}^f vectors obtained from 128 filters are concatenated to obtain the vector representation of the textual view of the tweet represented by \mathbf{X}

$$\mathbf{z} = \mathbf{z}^1 \oplus \mathbf{z}^2 \oplus \dots \oplus \mathbf{z}^{128} \quad (10)$$

For ease of reference, we can define the whole operation as:

$$\mathbf{z} = CNN(\mathbf{X}, \theta) \quad (11)$$

where θ denotes the required hyper-parameters of the CNN model such as k filters, h convolution window size. We apply 2-layers of the CNN model with the same parameters over the input \mathbf{X} to represent the input tweet, i.e.,

$$\mathbf{z}_{cnn} = CNN(CNN(\mathbf{X}, \theta), \theta) \quad (12)$$

In order to reduce the size of \mathbf{z}_{cnn} vector, the \mathbf{z}^i vector is further transformed to a scalar by applying *global maxpooling* over \mathbf{z}^i .

3.1.2 Bidirectional Encoder Representations from Transformers

Majority of the recent studies on text embedding consider using Bidirectional Encoder Representations from Transformers, more commonly known as BERT [10]. Earlier studies have considered using BERT as a pre-trained model [7, 6]. However, it is inefficient to use a pre-trained BERT model if it does not match the current domain of interest leading to out-of-vocabulary issues [27]. This study considers building BERT from scratch to overcome the inefficiency caused by using pre-trained BERT models.

Given a tweet representation $\mathbf{X} \in \mathbb{R}^{n \times d}$, the BERT model captures the semantic information of the word sequences by relying only on the attention-weighted representation of the words. The word order relation is incorporated into the initial word embedding \mathbf{X} by adding element-wise positional embedding. The position embedding for each word position pos can be defined as:

$$\mathbf{P}_{pos,i} = \begin{cases} \sin(pos/10000^{2i/d}) & \text{if } i \in (1, d) \text{ is even} \\ \cos(pos/10000^{2i/d}) & \text{otherwise} \end{cases} \quad (13)$$

There are l number of transformer blocks stacked on top of the other in the BERT architecture. The initial input to the first transformer block is the sum of word embedding \mathbf{X} and positional embedding

\mathbf{P} , i.e., $\mathbf{Z}_0 = \mathbf{X} + \mathbf{P}$. To capture the different aspects of tweet semantics, a transformer block t can have mh multi-head attention layers. For each attention head $i \in (1, mh)$ in a transformer block t , three matrices are generated using dense layer over the input \mathbf{Z}_t serving as the *query*, *key*, and *value* to find the attention-weighted representation using the *Scaled Dot-Product Attention* mechanism [25], i.e.,

$$\begin{aligned}\mathbf{Q}_i &= \mathbf{W}_{qi} \cdot \mathbf{Z}_t \\ \mathbf{K}_i &= \mathbf{W}_{ki} \cdot \mathbf{Z}_t \\ \mathbf{V}_i &= \mathbf{W}_{vi} \cdot \mathbf{Z}_t\end{aligned}\quad (14)$$

where $\mathbf{Q}_i, \mathbf{K}_i$, and \mathbf{V}_i is a linear transformation of the input \mathbf{Z}_t through three different weight parameters $\{\mathbf{W}_{qi}, \mathbf{W}_{ki}, \text{ and } \mathbf{W}_{vi}\} \in \mathbb{R}^{n \times n}$. The output of each attention head $i \in (1, mh)$ in a transformer block t can be defined as:

$$\mathbf{Y}_t^{(i)} = \text{Softmax}\left(\frac{\mathbf{Q}_t^{(i)} \cdot \mathbf{K}_t^{T(i)}}{\sqrt{|\mathbf{Q}_t^{(i)}|}}\right) \mathbf{V}_t^{(i)} \quad (15)$$

The attention-weighted outputs of the multi-head attention layer are concatenated to generate the semantic representation using a dense layer with *Relu* activation function as output for the transformer block t , i.e.,

$$\mathbf{Z}_{t+1} = \text{Relu}(\mathbf{W} \cdot \mathbf{Y}_{1:mh} + \mathbf{B}) \quad (16)$$

where $\mathbf{W} \in \mathbb{R}^{n \times n \cdot mh}$ and $\mathbf{B} \in \mathbb{R}^{n \times d}$ are the weights and biased parameter matrices, \mathbf{Z}_{t+1} represent the output of the t transformer block. The output of the last transformer block, i.e., \mathbf{Z}_{l+1} is considered as the final representation of the input tweet T to the BERT model. To represent in the vector space, \mathbf{Z}_{l+1} is being flattened into $\mathbf{z}_{bert} \in \mathbb{R}^{n \cdot d \times 1}$ vector for sentiment classification. For ease of reference, the whole operation can be defined as:

$$\mathbf{z}_{bert} = \text{BERT}(\mathbf{Z}_0, \theta) \quad (17)$$

where θ represents the hyper-parameters such as l number of encoders, mh number of multi-head attentions, d hidden layer dimensions. We have considered the same hyperparameters used in the original BERT setup, i.e., $l = 8$ transformer blocks and $mh = 8$ multi-head attentions.

3.2 Tweet graph construction

Using the same intuition of the study of Singh et al. [2], a tweet is represented as a heterogeneous multi-layer network by considering the link between hashtags, mentions, and normal tokens that co-occur in a tweet. Figure 1 illustrates how a tweet is represented in a heterogeneous multi-layer network. It can be highlighted that before transforming the tweet to the heterogeneous multi-layer network, a preprocessing step, such as removing stopwords, normalizing keywords (converting to lowercase, removal of URL links), and so on, could be performed. This study performs the removal of the stopwords and normalization of keywords before transforming the tweet to the heterogeneous multi-layer network structure.

In this study, the heterogeneous multi-layer network is represented using three types of undirected relations i.e. mention–mention (MM), hashtag–hashtag (HH), mention–hashtag (MH) or hashtag–mention (HM) and five directed relations i.e. keyword \rightarrow keyword (KK), keyword \rightarrow hashtag (KH), hashtag \rightarrow keyword (HK), keyword \rightarrow mention (KM), and mention \rightarrow keyword (MK). The directed edges are considered to capture the sequence relation of normal tokens. Accommodating all the eight types of relations of a tweet with n tokens can be represented in the adjacency matrix as:

$$\mathbf{A}_{n \times n} = \begin{bmatrix} \mathbf{B}^{HH} & \mathbf{B}^{HM} & \mathbf{B}^{HK} \\ \mathbf{B}^{MH} & \mathbf{B}^{MM} & \mathbf{B}^{MK} \\ \mathbf{B}^{KH} & \mathbf{B}^{KM} & \mathbf{B}^{KK} \end{bmatrix} \quad (18)$$

where $MH = HM$ and \mathbf{B}^r represent the adjacency matrix representation of the relation $r \in \{HH, MH, HM, MM, KK, KH, HK, KM, MK\}$.

The key difference between this study from that of Singh et al. [2] is in utilizing the supra-adjacency matrix $\mathbf{A}_{n \times n}$ to represent the structural information of the tweet graph. Singh et al.'s study encompasses using a biased random walker to generate alternative naturalistic word sequences via language modeling from the tweet graph to yield multifaceted representations of the tweet, followed by using a text embedding model (CNN, LSTM) to represent the random walk sequences and the original tweet

Table 1: Characteristics of the experimental datasets

Dataset	Pos	Neg	Neu	Total	Topics	Domain
<i>Socetial</i>	16,375	17,047	9,000	42,422	<i>Kashmir Unrest, Pathankot Attack, Surgical Strike, GSTN</i>	Social Issue
<i>SemEval-2016</i>	1,296	2,491	276	4,063	Atheism, Climate Change, Feminist Movement, Hillary Clinton, Legalization of Abortion	Social Issue
<i>SemEval-2013</i>	5,115	2,017	6,099	13,231	General Discussion	–

in latent space. In such a case, the relations of the nodes in the graph are not fully captured due to the noise created using a random walk and language model algorithm. To address the aforementioned issue, this study explores graph embedding techniques to harness node relationships and capture latent information of the heterogeneous graph. Recent studies of graph instance representation learning [12, 11] have shown promising results in capturing the latent representation of the graph. Therefore, to capture the insight relation of hashtags, mentions, and normal keywords, this study considers the graph representation learning methods [12, 11].

3.2.1 Network expansion

This study investigates *whether adding semantically related tokens into the tweet-graph can enrich the representation of the tweet*. To expand a tweet graph, the semantically related nodes of all tokens in the tweet are retrieved using cosine similarity over the word embeddings generated using FastText (FT) [24] and Sentiment Hashtag Embedding (SHE) [1] methods. We select the top 20 tokens having high cosine similarity scores to the tokens present in the tweet as semantically relevant nodes of the tweet. These 20 nodes are added to the tweet graph by introducing an undirected edge with all the nodes. For ease of reference, such a node expansion approach is considered as semantic *Node Expansion* (NE).

Further, we investigate *whether adding semantically related and sentiment polarized tokens into the tweet-graph can enrich the representation of the tweet or not*. For this study, the previously selected semantically similar nodes through NE are filtered by selecting only the sentiment-polarized tokens. To select the sentiment-polarized tokens, this study exploits the SHE method to classify the sentiment of the 20 semantically relevant nodes. Then, the sentiment-polarized node expansion is performed by dividing the 20 nodes into three different sentiment sets, i.e., positive, negative, and neutral. The dominating sentiment set, i.e., majority of the nodes having the same sentiment, is selected for sentiment-polarized node expansion. For ease of reference, this study considers such expansion approach as *Sentiment polarized Node Expansion* (SNE).

3.3 Graph representation model

Recent studies on graph instance representation learning [12, 11] have shown promising results in capturing the latent representation of the graph. We can apply graph instance representation learning methods such as Deep Graph Convolution Neural Network (DGCNN) [11] and Segmented-Graph BERT (Seg-BERT) [12] over $\mathbf{A}_{n \times n}$ to represent it in vector space for the graph classification task.

3.3.1 Deep Graph Convolution Neural Network

Zhang et al. [11] have used Graph Convolution Neural network (GCN) [28] for graph classification tasks. Compared to the study of Kipf and Welling [28], which work on a single structure, this method can represent graphs of arbitrary structures. They proposed an algorithm named *SortPooling* similar to the Weisfeiler-Lehman node coloring algorithm [29] for sorting vertex features to learn the global graph topology.

Given a graph $\mathbf{A}_{n \times n}$ and feature matrix (word embedding) $\mathbf{X} \in \mathbb{R}^{n \times d}$, we can apply multiple stacks of *GCN* at time t to output \mathbf{Z}^t as

$$GCN(\mathbf{Z}^{t-1}, \mathbf{A}) = ReLu(\tilde{\mathbf{A}}\mathbf{Z}^{t-1}\mathbf{W})$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times n}$ is the adjacency matrix with added self-loops (identity matrix), i.e., $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{Z}^0 = \mathbf{X}$, $\mathbf{W} \in \mathbb{R}^{d \times c}$ is the neural weight parameters¹ shared with all the graphs and h is the number of *GCN* layers. For learning global node features, the output of each *GCN* layers are concatenated

¹For generalization we set $c = d$

Table 2: Performance of sentiment classifiers over the *Societal* dataset.

Single-view methods	<i>Societal</i>		SemEval-2016		SemEval-2013	
	Accuracy	F-Macro	Accuracy	F-Macro	Accuracy	F-Macro
T (CNN)	77.16	76.08	73.41	47.26	64.42	61.98
T (BERT)	76.92	75.78	68.59	37.20	55.72	48.20
DGCNN	74.89	72.27	74.31	48.38	62.03	56.59
Seg-BERT	77.39	75.71	70.33	41.48	56.60	51.32
Multi-view methods	<i>Societal</i>		SemEval-2016		SemEval-2013	
	Accuracy	F-Macro	Accuracy	F-Macro	Accuracy	F-Macro
CNN+DGCNN (End-to-end)	78.70	76.83	74.31	47.92	64.84	62.19
CNN+DGCNN (Ensemble)	79.34	77.03	74.55	48.78	66.00	62.79
BERT+Seg-BERT (End-to-end)	73.36	72.12	68.81	52.26	60.42	54.76
BERT+Seg-BERT (Ensemble)	75.37	73.81	70.11	52.63	62.20	58.71
T+MLN (CNN)	76.69	73.97	72.22	53.63	63.49	60.16

row-wise i.e. $\mathbf{Z} = \mathbf{Z}^{1:h}$ and apply *SortPooling* over \mathbf{Z} i.e., $\mathbf{Z}_{sp} = \text{SortPooling}(\mathbf{Z})$. The output \mathbf{Z}_{sp} is fed to *CNN* layer to generate the graph representation via *MaxPooling*, i.e.,

$$\mathbf{z}_{dgcnn} = \text{MaxPool}(\text{CNN}(\mathbf{Z}_{sp}, \theta_{graph}))$$

where θ_{graph} is the learning parameters of CNN. We use the same parameters considered in the text representation model (refer to Section 3.1.1).

3.3.2 Segmented-Graph BERT

Zhang et al. [12] have used BERT architecture to encode graph information given node features such as word embeddings (\mathbf{X}), latent representation of adjacency neighborhood matrix (\mathbf{A}), node degree matrix (\mathbf{D}), and node global role matrix (\mathbf{WL}) pre-computed using Weisfeiler-Lehman algorithm [29]. We feed these features as input to the BERT model, i.e.,

$$\mathbf{Z}_0 = \mathbf{X} + \mathbf{A} + \mathbf{D} + \mathbf{WL} \quad (19)$$

Hence, we can learn graph instance representation of a graph similar to the normal BERT model, which captures semantic relations of the nodes in the graph as

$$\mathbf{z}_{segbert} = \text{BERT}(\mathbf{Z}_0, \theta_{graph}) \quad (20)$$

where θ_{graph} is the learning parameters of BERT. We use the same parameters considered in the text representation model (refer to Section 3.1.2).

4 Experimental setup

4.1 Dataset

To evaluate the efficacy of the proposed framework, this study considers a *Societal* dataset used in [26, 2] for the sentiment classification task. This dataset contains 1,505 under-specified tweets (tweets having less than five tokens) and 1,626 multilingual tweets (code mix of Hindi and English languages). The *Societal* dataset is curated over four topics that happened in India, namely Kashmir Unrest, Pathankot Attack, Surgical Strike, and GSTN². Table 1 shows the characteristics of the training dataset considered in this study.

4.2 Baseline classifiers

To evaluate the performance of the proposed framework, we consider four single-view classifiers, i.e., *CNN*, *BERT*, *DGCNN*, and *Seg-BERT*, and two multi-view classifiers, i.e., *T+MLN* and *VGCN-BERT* as baseline models for comparison.

- **CNN**: The output of \mathbf{z}_{cnn} of CNN model over the input \mathbf{X} is considered as the tweet representation for sentiment classification task in Equation 5.
- **BERT**: The output of \mathbf{z}_{bert} of BERT model over the input \mathbf{X} is considered as the tweet representation for sentiment classification task in Equation 5.

²[https://en.wikipedia.org/wiki/Goods_and_Services_Tax_\(India\)](https://en.wikipedia.org/wiki/Goods_and_Services_Tax_(India))

Table 3: Hyperparameters settings

Hyperparameters	Value
Optimizer	Adam [30]
Learning rate	0.0001
Training Epochs	40
Batch size	64
Dropout rate	0.1
# Filters (CNN, GCN)	128
Word embedding dimension	60
Maximum sequence length	40

- **DGCNN**: The output of \mathbf{z}_{dgcnn} of DGCNN model over the input \mathbf{X} is considered as the tweet representation for sentiment classification task in Equation 5.
- **Seg-BERT**: The output of $\mathbf{z}_{seg-bert}$ of Seg-BERT model over the input \mathbf{X} is considered as the tweet representation for sentiment classification task in Equation 5.
- **T+MLN**: The work of Singh et al. [2] is considered one of the baseline methods for incorporating graph and text information.

4.3 Hyper-parameters

The selection of hyperparameters is critical in determining the behavior of the models used for effective sentiment analysis. The number of layers, the size of hidden units, and the dropout rates of the GCN all significantly influence the network’s ability to learn graph-based representations. Similarly, parameters such as filter sizes, pooling methods, and the number of layers impact the CNN model’s capacity to capture local and global information within text sequences. On the other hand, the BERT model relies on hyperparameters like learning rate, batch size, and sequence length during training to impact its ability to contextualize words effectively. Table 3 shows the hyperparameters considered in this study.

5 Results and Observation

In this section, we investigate the efficacy of the proposed framework over the baseline methods through the two research questions – (i) *How informative is a graph-based representation of a tweet as compared to that of the text-based representation?* (ii) *Does the text-based and graph-based representations complement each other?* The efficacy of the proposed framework is investigated over the *Societal* dataset using a 10-fold cross-validation strategy. Table 2 shows the performance of the classifiers over *Societal* and *SemEval* datasets for the sentiment classification task. The under-specified and multilingual tweets are excluded from the *Societal* dataset for this analysis study. These tweets are considered to investigate whether the proposed model can address the challenge of social media noises.

5.1 How informative is a graph-based representation of a tweet compared to a text-based representation?

The first part of Table 2, i.e., single-view methods, shows the performances of the single-view classifiers. In the societal dataset, it is observed that the best performance achieved by a single-view classifier is up to 77.39% accuracy with an F-Macro of 75.71% using Seg-BERT over the heterogeneous tweet graph. While, the performance of the sentiment classifier built over text representation, i.e., text-view, can achieve the best performance up to 77.16% accuracy using CNN. Similarly, in the SemEval 2016 dataset, it is observed that the graph-based classifier DGCNN can achieve the best up to 74.31% with 48.38% F-Macro score while the text-based classifier CNN can achieve a comparable performance accuracy of 73.41% and 47.26% F-Macro score over the same dataset. In contrast, it is observed that the CNN classifier can achieve the best performance of 64.42% accuracy with an F-macro score of 61.98% over the SemEval 2013 dataset, while the DGCNN classifier can achieve a best up to 62.03% with a 56.59% F-Macro score over the SemEval 2013 dataset. It is observed that the CNN-based classifier can perform better than the BERT-based classifier.

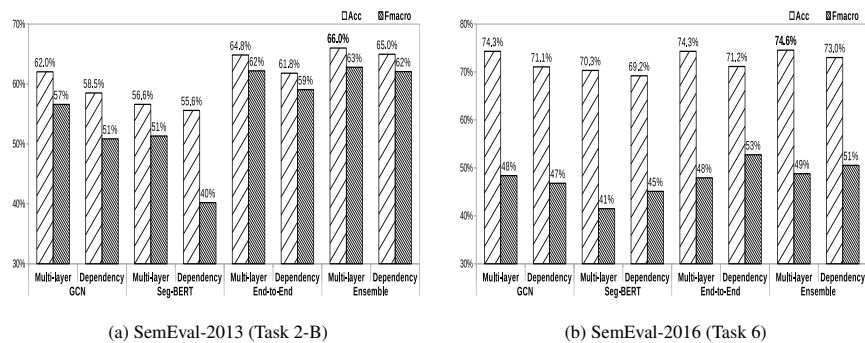


Figure 3: Performance of classifiers over SemEval 2013 and 2016 challenge datasets. End-to-end and Ensemble classifiers are the combination of CNN and DGCNN methods.

The primary reason is due to the smaller amount of training samples. However, it is observed that using the Societal dataset (a larger experimental dataset), the graph-based BERT classifier (i.e., Seg-BERT) with the additional graph-based information can enhance the classifier’s performance. The above experiments over different corpus sizes establish that the graph-based representation can achieve comparable performance with the text-based approach. The following section further investigates using the proposed framework to comprehend if the text and graph views complement each other.

5.2 Do the text-based and graph-based representations complement each other?

Table 2 second part shows the performance result of incorporating graph and text views for the sentiment classification task. It is evident from the Table 2, i.e., multi-view methods, that incorporating both the text and graph views have significantly improved the performance of sentiment classifiers for both *end-to-end* and *ensemble* than the single-view based methods. The *ensemble* frameworks using CNN and DGCNN methods can achieve the best performance of up to 79.34% accuracy and 77.03% F-Macro scores. In contrast, its *end-to-end* framework can achieve up to 78.70% accuracy and 76.35% F-Macro score. It is observed that the performance of multi-view classifiers using BERT and Seg-BERT could not improve the performance compared to its individual classifier performances over *Societal* and SemEval 2016 datasets. One of the reasons for not performing well compared to the individual view is that Seg-BERT takes both the text and graph information while encoding graph representation. In contrast, BERT takes only the text information to encode sequence representation. Hence, the tweet representation generated using Seg-BERT has redundant information. Adding BERT information in the multi-view framework has created a noisy representation of the tweet due to the losses while training the multi-view framework. Among the baseline methods for incorporating multi-views, the T+MLN classifier can achieve the best up to 76.69% accuracy and 73.97% F-macro score. It is also observed that the best performance of the single-view and multi-view classifiers over the SemEval-2016 dataset is relatively comparable. However, a clear difference between single-view and multi-view classifiers’ best performance is observed in the *Societal* and SemEval-2013 datasets. One of the reasons for underperforming is due to the small size corpus. Compared to the *Societal* and SemEval-2013 datasets, the corpus size of the SemEval-2016 dataset is minimal; therefore, the node information in the tweet graphs of this dataset is not fully incorporated. With a larger corpus, the graph representation learning method can benefit the global properties of the nodes. As a result, the performance of the *end-to-end* and *ensemble*-based classifiers have significantly improved using DGCNN. This study shows that the node’s properties in the tweet graph can inherently be captured with a larger corpus. Further, this study shows that incorporating text and graph views can better enrich the tweet representation for sentiment classification tasks than individual classifier performance. Therefore, from the above investigation, it is evident that both text and graph views complement the representation of the tweet for sentiment classification.

5.3 Heterogeneous multi-layer network v/s Dependency tree

Building a dependency parser is expensive and not feasible in a multilingual context. Therefore, this section investigates *if there is a need for a language-dependent dependency parser to construct the tweet graph*. In this study, we investigate the performance of the proposed model using the language-insensitive multilayer network to the language-specific dependency graph. We consider an

off-the-shelf dependency parser in English language³ to construct the tweet graph. Since SemEval datasets are English language datasets, we consider these datasets for the experimental study. The sentiment classification performance of tweets is evaluated over two variant representations of the tweet, i.e., the tweet represented using the dependency parser and the heterogeneous multi-layer network. Figure 3 shows a performance comparison of the single-view and multi-view classifiers over the tweet graph using the dependency parser and the heterogeneous graph. It is evident from the figures that the performances of the single-view classifiers, i.e., DGCNN and Seg-BERT, over the tweet representation using the heterogeneous graph have better classification accuracy than using the dependency graph. It is observed that the best-performing classifiers (i.e., *ensemble* classifier) for both graph representations are relatively comparable. The *ensemble* classifier trained over the SemEval 2013 dataset using the heterogeneous multi-layer network can achieve the best of up to 66% accuracy while using the dependency graph can achieve up to 65% accuracy. Similarly, the *ensemble* classifiers trained over the SemEval 2016 dataset using the heterogeneous multi-layer network can achieve the best of up to 75% accuracy while using the dependency graph can achieve up to 73% accuracy. This study shows that the heterogeneous multi-layer network is language invariant and performs better than language-dependent word graph structure.

Further, *to investigate whether the heterogeneous multi-layer graph is less sensitive to social media-related noises*, we investigate the performance of the proposed framework over the under-specified and multilingual tweets in the following subsections. This study considers tweets with less than five tokens as under-specified tweets. To investigate the performance of the proposed framework over these tweets, the best performing *ensemble*-based classifier (as observed in Table 2), i.e., the ensemble of CNN and DGCNN classifiers and the single-view classifiers, are considered. The performances of the single-view classifiers are compared with the *end-to-end* and *ensemble* frameworks using the under-specified and multilingual tweets. Furthermore, we investigate the performance of sentiment classifiers on classifying tweets by adding semantically relevant tokens and sentiment-polarized tokens to the tweet graph through NE and SNE approaches. For this study, the classifiers are not re-trained over the expanded tweet graphs. Instead, the representation of tweets is generated from the expanded graph for comparison. For ease of reference, we use the notation *classifier+NE* to indicate the classifier uses the expanded graph generated using either NE or SNE approaches for sentiment classification.

5.4 Performance of sentiment classification over under-specified tweets

As mentioned above, to investigate the performance of the proposed framework over the under-specified tweets, the best performing *ensemble*-based classifier (in Table 2), i.e., the ensemble of CNN and DGCNN classifiers and the single-view classifiers, i.e., CNN, BERT, DGCNN, Seg-BERT are considered for comparison. From the Figure 4(a), it is observed that the proposed framework using *ensemble*-based method outperforms the individual-view-based classifiers by achieving the best accuracy up to 70.60% and F-macro score of 66.40%. While the *end-to-end* framework is able to achieve up to 69.32% accuracy and 62.30% F-macro score. The best performance of a single-view classifier is using CNN classifier, which can achieve up to 65.93% accuracy and 60.40% F-macro score, followed by BERT with 64.89% and 62.19% accuracy and F-macro scores, respectively. Among the graph-based approach, DGCNN can achieve up to 63.51% accuracy and 58.80% F-macro score, while Seg-BERT achieves up to 61.89% and 53.99% accuracy and F-macro scores, respectively. This study shows that incorporating both views can better represent tweets than representations of their individual views.

Further, after performing semantic *Node Expansion* (NE) over the under-specified tweet graph, it is evident from the Figure 4(c) that the performance of the classifiers significantly improves. It is observed that with NE, the performance of DGCNN+NE and Seg-BERT+NE improves to 77.62% and 74.56% accuracies, respectively. Incorporating the text and graph-views using NE in the *end-to-end* framework (i.e., *end-to-end+NE*) further improves the classifier performance by 78.42% accuracy. With *ensemble* framework (i.e., *ensemble+NE*), the classifier performance improves up to 78.82% accuracy. Furthermore, after performing *Sentiment polarized Node Expansion* (SNE), the best performance we can achieve is up to 78.85% accuracy using the ensemble classifier, i.e., Ensemble+SNE. It shows that adding semantically related polarized sentiment nodes in the tweet graph can further enrich the tweet representation even without re-training the classifiers. From this study, it is evident that the proposed framework can address the problem of the under-specificity of tweets with a high margin compared to the performance of the single-view classifiers.

³<https://spacy.io/usage/linguistic-features>

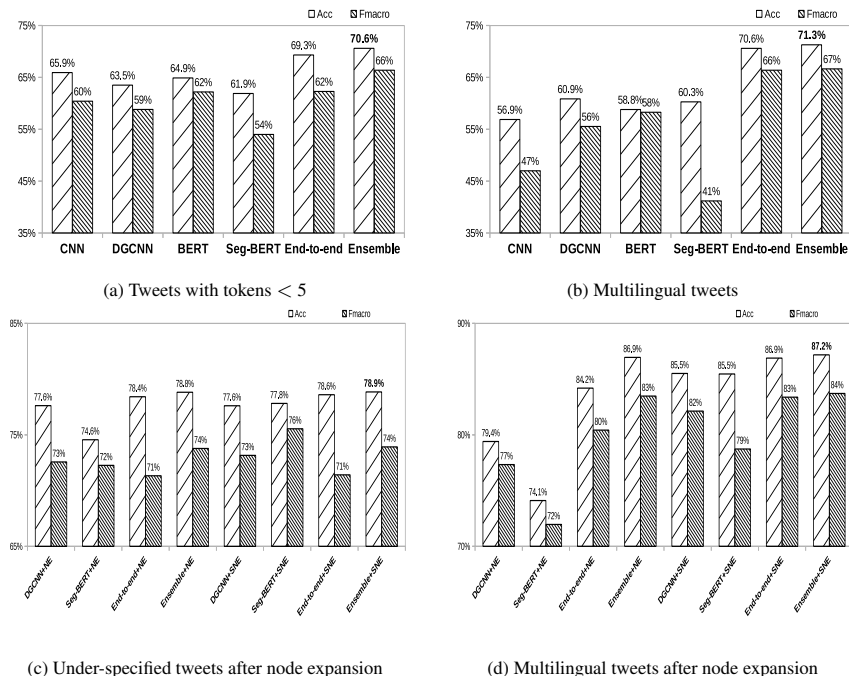


Figure 4: Performance of classifiers over different under-specified and multi-lingual tweet categories. End-to-end and Ensemble classifiers are combination of CNN and DGCNN methods; *classifier+NE* is the *classifier* performance of tweet classification over node expansion graph; *classifier+SNE* is the *classifier* performance of tweet classification over sentiment polarized node expansion graph

5.5 Performance of sentiment classification over multilingual tweets

This section investigates the performance of the proposed framework over multilingual tweets. Similar to the under-specified tweets evaluation study, the best performing *ensemble*-based classifier, i.e., the ensemble of CNN and DGCNN classifiers and the single-view classifiers using SHE, are compared. It is observed from Figure 4(b) that incorporating both the text and graph views in the *ensemble* framework can achieve up to 71.28% accuracy and 66.64% F-macro score. While the *end-to-end*-based classifier can achieve up to 70.59% accuracy and 66.38% F-macro. Among the single-view classifiers, the DGCNN classifier has achieved the highest of 60.86% accuracy and 55.54% F-macro, followed by Seg-BERT with 60.28% 41.18% accuracy and F-macro scores, respectively. The BERT classifier can achieve up to 59% accuracy and 58% F-macro scores, while the CNN classifier can achieve up to 57% accuracy and 47% F-macro scores. This shows that incorporating both text and graph views can better represent a tweet than representing its individual views.

Further, with node expansion of the tweet-graph, the improvement of the performance of classifiers is evident in Figure 4(d). The DGCNN and Seg-BERT classifier over the NE of tweet-graph can achieve up to 79.41% and 74.1% accuracies, respectively. Further, incorporating text representation over the expanded graph using *end-to-end* and *ensemble* frameworks improves the performance of the classifiers by achieving up to 84.19% and 86.95% accuracy, respectively. Furthermore, with SNE of the tweet-graph, the best performance we can achieve is up to 87.17% accuracy using *ensemble* of text representation and 86.95% *end-to-end* representation of SNE of tweet-graph. This study shows that the proposed framework of incorporating text and graph views can better enrich the tweet representation than its single-view representation. The tweet representation can be enriched further by adding semantically related sentiment-polarized nodes in the tweet graph. It is also evident that the proposed framework can address the problem of multilingual tweets by incorporating both text and graph views in the multi-view learning framework.

5.6 Limitation of the proposed framework

While the classifiers do show improved performance with the integration of SNE, a closer examination of Figure 4(c, d) reveals that the classifier performances for NE and SNE of tweet-graphs exhibit relatively comparable outcomes. The relatively lower effectiveness of SNE could potentially arise

from its method of filtering sentiment-polarized nodes. This technique selects nodes primarily based on the dominant sentiment, resulting in a similar count of nodes chosen through SNE and the entirety of nodes chosen through NE (refer to Section 3.2.1). This equality in node selection quantities might contribute to the observed similarity in classifier performance between NE and SNE. Exploring possibilities to uncover sentiment-polarized nodes that are more particularly related to the input tweet may give further options to improve sentiment classification performance. Therefore, developing an approach for determining sentiment-polarized nodes that are relevant to the input tweet may help the sentiment classification task perform more effectively.

6 Conclusion and Future work

This study proposes a multi-view learning framework for sentiment classification of tweets to address under-specificity, noise, and multilingual content by representing tweets using text and graph representation learning methods. To incorporate text and graph views in the multi-view learning framework, this study explores both *end-to-end* and *ensemble* based classifiers. It is observed from various experimental studies that the performance of the tweet sentiment classifier improves significantly after incorporating both text and graph views than its individual-view classifiers. The *ensemble* based classifier can perform better than *end-to-end* based classifier on incorporating both the views. Further, it is observed that the proposed framework can perform better than its counterpart in addressing multilingual and under-specified tweets. Moreover, after performing node expansion over the tweet graph, the performance of the classifiers improves further through semantic (NE) and sentiment polarized node expansion (SNE). To further enhance the performance of the sentiment classification task, retrieving relevant sentiment polarized nodes of the input tweet can be a future scope of the study.

Acknowledgments

We extend our sincere appreciation to the Indian Institute of Technology Guwahati (IITG) for providing the resources that facilitated the research and development of this work. The contributions of the Open Source Intelligence (OSINT) Group at the Department of Computer Science and Engineering, IITG, have been invaluable in shaping the outcomes of this project. We are grateful for the support and collaboration that enriched the research experience.

References

- [1] L. G. Singh, A. Anil, and S. R. Singh. She: Sentiment hashtag embedding through multitask learning. *IEEE Transactions on Computational Social Systems*, 7(2):417–424, 2020.
- [2] Loitongbam Gyanendro Singh, Anasua Mitra, and Sanasam Ranbir Singh. Sentiment analysis of tweets using heterogeneous multi-layer network representation and embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8932–8946, 2020.
- [3] Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. Microblog sentiment classification via recurrent random walk network learning. In *IJCAI*, volume 17, pages 3532–3538, 2017.
- [4] Guang-You Zhou and Jimmy Xiangji Huang. Modeling and mining domain shared knowledge for sentiment analysis. *ACM Transactions on Information Systems (TOIS)*, 36(2):1–36, 2017.
- [5] Nora Al-Twairish and Hadeel Al-Negheimish. Surface and deep features ensemble for sentiment analysis of arabic tweets. *IEEE Access*, 7:84122–84131, 2019.
- [6] Zhibin Lu, Pan Du, and Jian-Yun Nie. Vgcn-bert: augmenting bert with graph embedding for text classification. In *European Conference on Information Retrieval*, pages 369–382. Springer, 2020.
- [7] Fanyu Meng, Junlan Feng, Danping Yin, Si Chen, and Min Hu. Sentiment analysis with weighted graph convolutional networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 586–595, 2020.
- [8] Chen Zhang, Qiuchi Li, and Dawei Song. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4568–4578. Association for Computational Linguistics, 2019.

- [9] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, 2014. Association for Computational Linguistics.
- [10] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.
- [11] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [12] Jiawei Zhang. Segmented graph-bert for graph instance modeling. *arXiv preprint arXiv:2002.03283*, 2020.
- [13] Yuting Ma, Rui Song, Xue Gu, Qiang Shen, and Hao Xu. Multiple graph convolutional networks for aspect-based sentiment analysis. *Applied Intelligence*, 53(10):12985–12998, 2023.
- [14] Bengong Yu and Shuwen Zhang. A novel weight-oriented graph convolutional network for aspect-based sentiment analysis. *The Journal of Supercomputing*, 79(1):947–972, 2023.
- [15] Lvxiaowei Xu, Xiaoxuan Pang, Jianwang Wu, Ming Cai, and Jiawei Peng. Learn from structural scope: Improving aspect-level sentiment analysis with hybrid graph convolutional networks. *Neurocomputing*, 518:373–383, 2023.
- [16] Bin Liang, Hang Su, Lin Gui, Erik Cambria, and Ruifeng Xu. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, 235:107643, 2022.
- [17] Anna Nguyen, Antonio Longa, Massimiliano Luca, Joe Kaul, and Gabriel Lopez. Emotion analysis using multilayered networks for graphical representation of tweets. *IEEE Access*, 10:99467–99478, 2022.
- [18] Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis. *Information Sciences*, 589:416–439, 2022.
- [19] Haiyan Wu, Zhiqiang Zhang, Shaoyun Shi, Qingfeng Wu, and Haiyu Song. Phrase dependency relational graph attention network for aspect-based sentiment analysis. *Knowledge-Based Systems*, 236:107736, 2022.
- [20] Zheng Zhang, Zili Zhou, and Yanna Wang. Ssegcn: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4916–4925, 2022.
- [21] Wenxiong Liao, Bi Zeng, Jianqi Liu, Pengfei Wei, Xiaochun Cheng, and Weiwen Zhang. Multi-level graph neural network for text sentiment analysis. *Computers & Electrical Engineering*, 92:107096, 2021.
- [22] Junjie Chen, Hongxu Hou, Yatu Ji, and Jing Gao. Graph convolutional networks with structural attention model for aspect based sentiment analysis. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.
- [23] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.
- [24] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [26] Loitongbam Gyanendro Singh and Sanasam Ranbir Singh. Empirical study of sentiment analysis tools and techniques on societal topics. *Journal of Intelligent Information Systems*, pages 1–29, 2020.
- [27] Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. Domain adaptation challenges of bert in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, 2020.
- [28] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

- [29] AA Leman and B Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.
- [30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015.