Combining Graph NN and LLM for Improved Text-Based Emotion Recognition

Xinhao Zou and Konstantin Markov(✉)

The University of Aizu, Fukushima 965-8580, Japan
{d8261110,markov}@u-aizu.ac.jp

Abstract. Text-based emotion analysis, an important task in Natural Language Processing (NLP), aims to identify and understand emotional tendencies in text. Recently, given their strong performance in text classification, Graph Neural Networks (GNNs) have been utilized in various emotion recognition studies. They have excellent structural modeling abilities but lack context encoding strength. On the other hand, Large Language Models (LLMs) such as BERT and GPT are specially designed to model the text context. Aiming to utilize both their advantages, we investigated several ways to combine GNNs with LLMs for the emotion recognition task. First, we used BERT to generate embeddings for the graph document nodes. Next, we extended the system to include a description of the input data's emotional content obtained from GPT as an additional node embedding. For experiments and system evaluation, we used the GoEmotions dataset. The results clearly show that combining GNN and LLM improves the emotion classification performance by 20% to 30% compared to when either GNN or LLM is used alone.

Keywords: Graph Neural Network · Large Language Model · Emotion analysis and recognition

1 Introduction

In recent years, emotion analysis has gradually emerged as an important field of Natural Language Processing (NLP) [15]. However, the traditional emotion analysis methods have significant limitations when dealing with complex context and text structure [3]. Sometimes it is difficult to combine context and situation [17,22], and the fixed vocabulary size may affect the performance. Some words often have different meanings in different contexts, which may limit the ability to predict the emotional content precisely [7,8,12]. A wide range of studies in this area use categorical emotion labels which makes the emotion recognition task a supervised classification task. However, continuous [9] or distributed [21] emotion representation has also been explored.

The deep learning technology has significantly transformed the NLP field and as a consequence the text-based emotion recognition methods. Large language

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
P. Koprinkova-Hristova and N. Kasabov (Eds.): AIMSA 2024, LNAI 15462, pp. 143–154, 2025.
https://doi.org/10.1007/978-3-031-81542-3_12

models such as BERT [5] and GPT [16] have been utilized in various studies for emotional analysis [1]. On the other hand, Graph Neural Networks (GNN) [25] has also been used for text processing [10,24]. Furthermore, a system combining GNN and BERT has achieved excellent results on the text classification task [13]. Following the same approach, we first built a GNN-based emotion recognition system where document node features are initialized from the corresponding document representation obtained from the BERT model. The results we obtained in terms of emotion recognition accuracy were about 10% higher than if we used only BERT or GNN models separately.

When the text documents are short and as in our case just a single sentence, there might not be enough words to estimate the emotion category reliably. However, using generative LLM such as GPT it is possible to extend the text length and use the LLM power to elaborate on the emotional content of the input sentence. This way we obtain additional text, longer than the original sentence and closely related to its emotional content. Next, we use again the BERT model to extract a representation vector for this additional text explanation. Finally, vectors from both the original text and its explanation are concatenated and used as node features for the GNN. This approach further improved the performance of our system by another 15% accuracy compared to the combined BERT/GNN case.

2 Graph Neural Network

Graph Neural Network (GNN) is a class of deep learning models designed for processing graph-structured data. Entities in graph-structured data are represented as nodes, and the relationships between entities are represented as edges. The goal of GNN is to learn useful representations from graph-structured data and utilize these representations for various tasks. In Graph Neural Networks, the adjacency matrix A is an important tool for describing graph structure. It is a two-dimensional matrix that represents the connection relationships between nodes in the graph. In the adjacency matrix, rows and columns represent nodes, and the matrix elements represent connections between nodes. For a graph, if there is a connection between node i and node j, the elements in the corresponding (i, j) and (j, i) positions in the adjacency matrix typically have a value of 1 or represent the weight of the connection. If there is no connection between the nodes, the corresponding element value is 0.

Each node encompasses a set of defining features. These features can vary based on the application. Edges between nodes might signify connections between entities with similar attributes, indicating some form of association or mutual influence. In this context, these features constitute feature vectors, serving as mathematical representations of nodes that capture various properties and characteristics within the graph structure.

The core idea of GNN is to aggregate neighbor information of nodes through the message-passing mechanism and update the feature representation of nodes. This process typically involves multiple iterations to capture information farther away in the graph. Eventually, each node's feature representation will contain information about its neighbors and more distant nodes.

2.1 Graph Convolutional Network

Consider a graph $G = (V, E)$ where V $(|V| = n)$ is a set of nodes, and E is a set of edges. Let $X \in \mathbb{R}^{n \times d}$ be a matrix containing all n nodes d-dimensional features and let A be the adjacency matrix.

GCN [11] is a basic variant of GNN that aggregates node neighbor information through graph convolution operations. The graph convolution operation of GCN can be expressed as:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{1}$$

where $H^{(l)}$ is the node representation matrix for layer l, $W^{(l)}$ represents the weight matrix of layer l, $\widetilde{A} = A + I$, $\widetilde{D}$ is the degree matrix of $\widetilde{A}$, and σ is a non-linearity such as ReLU [2]. Graphically, the operations in Eq.(1) are depicted in Fig. 1. For the first GCN layer, the node representation is initialized by the node feature matrix, i.e. $H^{(0)} = X$.

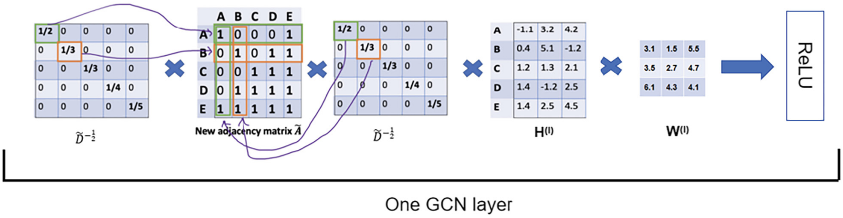


Fig. 1. Single GCN layer operation.

The above Eq.(1) describes the operation of a single GCN layer. Generally, several GCN layers are stacked for a complete GCN. The final layer representation $H^{(L)}$ is fed to a fully connected dense layer to obtain the model output. The overall structure of a multi-layer GCN is shown in Fig. 2:

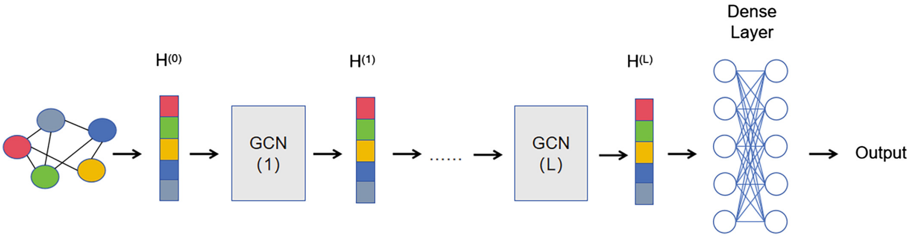


Fig. 2. The structure of a full GCN model.

2.2 GNN for Emotion Recognition

A collection of document texts can be represented by a graph considering the relationship between words and documents they are used in. Such a graph includes two sets of nodes: word nodes and document nodes, each representing one document or one unique word. Edges exist between each document node and word nodes corresponding to the words in that document. Generally, no other edges are needed, but following the approach taken in the TextGCN model [23], we also add weighted edges between word nodes.

With GNN, estimating the document's emotion is to find to which emotional class the corresponding node belongs. This is a semi-supervised task since all the node representations are used during training, but the loss calculation involves only the labeled training nodes. The structure of such a graph is given in Fig. 3 and the adjacency matrix is defined as in Eq.(2).

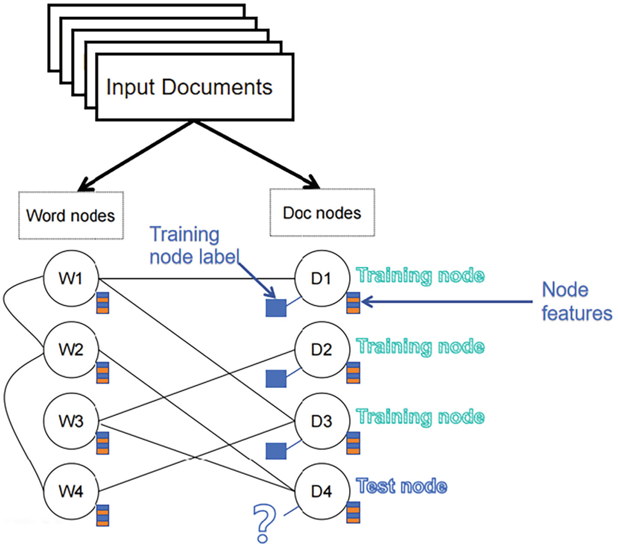


Fig. 3. Graph structure representing documents collection.

$$A_{i,j} = \begin{cases} PPMI(i,j), & i,j \text{ are words and } i \neq j \\ TF\text{-}IDF(i,j), & i \text{ is document, } j \text{ is word} \\ 1, & i = j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The word-document edge weights are the word term frequency-inverse document frequency (TF-IDF) value and the word-word edge weights are set to the positive point-wise mutual information (PPMI) value:

$$PPMI(\omega_i, \omega_j) = \max(\log \frac{P(\omega_i, \omega_j)}{P(\omega_i)P(\omega_j)}, 0) \tag{3}$$

where the word probabilities are estimated from the normalized word counts. In the absence of additional information about words and documents, node features are initialized with an identity matrix, $X = I_{n+m}$, where n is the number of documents and m is the number of unique words, i.e. the vocabulary size.

3 Large Language Models (LLMs)

3.1 BERT-Based Emotion Recognition

BERT is a pre-trained language model built on the Transformer's encoder architecture. There are several variants such as RoBERTa, DistilBERT, etc. which have the same structure but differ in the way they are trained. BERT outputs vector representations of the input words as well as a single vector representing the whole text document, the CLS output. Using this output it is easy to build an emotion classifier as shown in Fig. 4. A couple of linear dense layers take the CLS output and produce the estimated class probabilities using a softmax function. To achieve better performance on the downstream task, in our case - emotion recognition, pre-trained models are usually fine-tuned [19] in a supervised manner using the task data. The same approach can be used with RoBERTa or DistilBERT models [1].

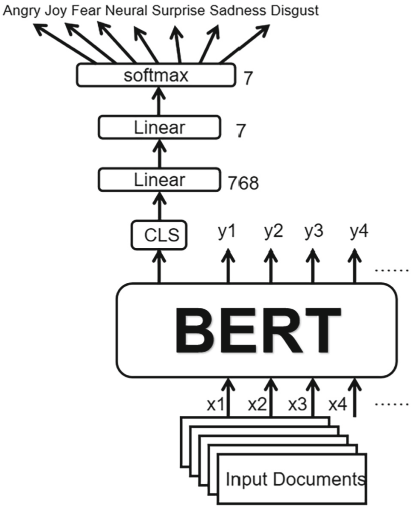
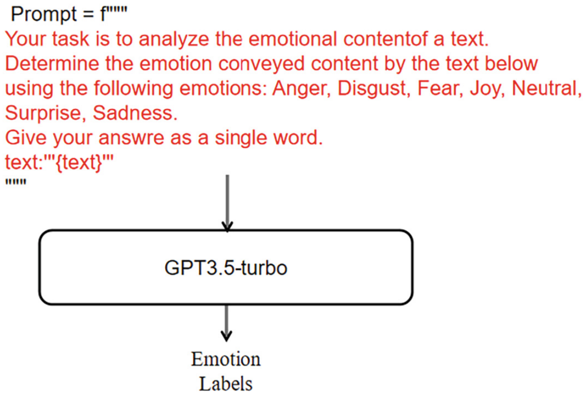


Fig. 4. Schematic diagram of BERT-based emotion recognition model.

3.2 GPT-Based Emotion Recognition

Generative large language models such as GPT3.5 [16] have performed remarkably well in various text-processing tasks. Thus, analyzing the emotional content of a text and classifying it into several categories can be done using LLM alone [20]. The simplest approach would be to create a proper prompt and obtain the LLM answer. The architecture of an LLM-based emotion recognition system is shown in Fig. 5. The LLM is instructed to analyze the input document's emotional content and label it using a set of predefined emotion classes and to output the corresponding label.



```
Prompt = f"""
Your task is to analyze the emotional contentof a text.
Determine the emotion conveyed content by the text below
using the following emotions: Anger, Disgust, Fear, Joy, Neutral,
Surprise, Sadness.
Give your answre as a single word.
text:'''{text}'''
"""
```

Fig. 5. LLM-based emotion recognition system structure.

4 Combining LLM and GNN

4.1 BERT and GNN

In Sect. 2.2 we introduced the GNN for emotion recognition and defined the initial node features as an identity matrix $X = I_{\mathrm{n+m}}$. However, it is possible to use the BERT-derived input text representations as document node features which are much more discriminative and would ensure better performance [13]. In this case, the input feature matrix is defined as:

$$X = \begin{pmatrix} X_{doc} \\ 0 \end{pmatrix}_{(n+m)\times d} \tag{4}$$

where X_{doc} is the matrix of d-dimensional embeddings of all the n documents and the m word node feature vectors are set to zero. The block diagram of such a system is shown in Fig. 6. The BERT model is used similarly as in Sect. 3.1 but without the output softmax layer.

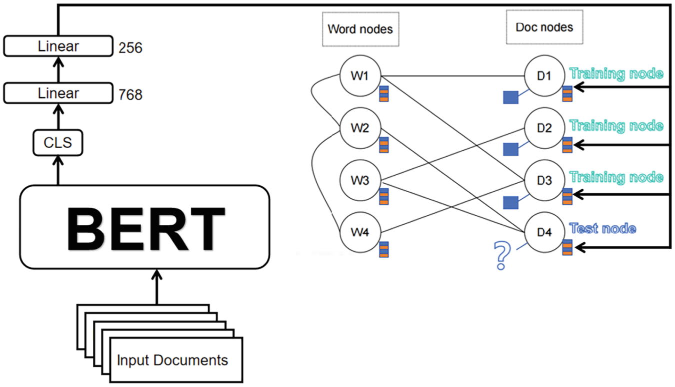


Fig. 6. GNN system with BERT-derived document node features.

To fully utilize the BERT power, the outputs of this system Z_{GNN} can be combined with the outputs of the BERT-only classifier Z_{BERT}:

$$Z = \lambda \cdot Z_{GNN} + (1 - \lambda) \cdot Z_{BERT} \tag{5}$$

where $\lambda \in (0, 1)$ is the linear combination hyper-parameter. Similarly to [13], in our experiments, we use GCN layers (see Sect.2.1) and denote this system as BertGCN.

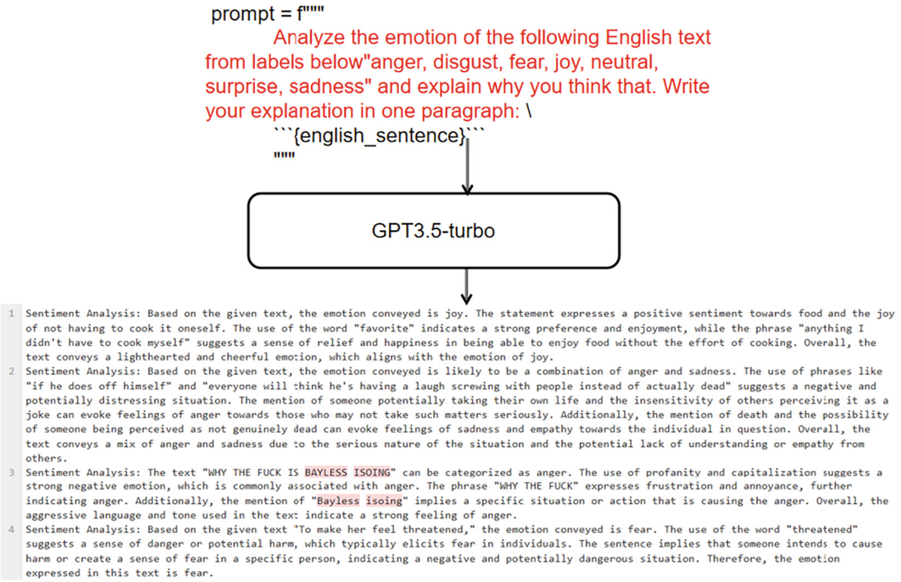
4.2 BERT, GNN, and GPT

The BERT model can provide highly discriminative document embeddings, but when the input text consists of a single sentence with a few words the classification task becomes more challenging. This problem can be alleviated using the generation abilities of models such as GPT. With an appropriate prompt, based on the GPT's reasoning power, the information about the document's emotional content can be significantly expanded. As shown in Fig. 7, we instruct the model to analyze the emotional content of the input text and provide a reasoning or explanation of its decision within one paragraph. The contextually rich explanations generated by GPT are transformed into representation vectors using BERT in the same way the input documents are processed. Representation vectors for the input and its GPT explanation are further concatenated and used as features for the document nodes schematically depicted in Fig. 8.

5 Experiment

5.1 Dataset

The dataset used in this study is the GoEmotions corpus [4,18]. It consists of 58K Reddit comments annotated manually. The comments are drawn from



~~~
prompt = f"""
        Analyze the emotion of the following English text
from labels below"anger, disgust, fear, joy, neutral,
surprise, sadness" and explain why you think that. Write
your explanation in one paragraph: \
        ```{english_sentence}```
        """
~~~

```
Sentiment Analysis: Based on the given text, the emotion conveyed is joy. The statement expresses a positive sentiment towards food and the joy
of not having to cook it oneself. The use of the word "favorite" indicates a strong preference and enjoyment, while the phrase "anything I
didn't have to cook myself" suggests a sense of relief and happiness in being able to enjoy food without the effort of cooking. Overall, the
text conveys a lighthearted and cheerful emotion, which aligns with the emotion of joy.
Sentiment Analysis: Based on the given text, the emotion conveyed is likely to be a combination of anger and sadness. The use of phrases like
"if he does off himself" and "everyone will think he's having a laugh screwing with people instead of actually dead" suggests a negative and
potentially distressing situation. The mention of someone potentially taking their own life and the insensitivity of others perceiving it as a
joke can evoke feelings of anger towards those who may not take such matters seriously. Additionally, the mention of death and the possibility
of someone being perceived as not genuinely dead can evoke feelings of sadness and empathy towards the individual in question. Overall, the
text conveys a mix of anger and sadness due to the serious nature of the situation and the potential lack of understanding or empathy from
others.
Sentiment Analysis: The text "WHY THE FUCK IS BAYLESS ISOING" can be categorized as anger. The use of profanity and capitalization suggests a
strong negative emotion, which is commonly associated with anger. The phrase "WHY THE FUCK" expresses frustration and annoyance, further
indicating anger. Additionally, the mention of "Bayless isoing" implies a specific situation or action that is causing the anger. Overall, the
aggressive language and tone used in the text indicate a strong feeling of anger.
Sentiment Analysis: Based on the given text "To make her feel threatened," the emotion conveyed is fear. The use of the word "threatened"
suggests a sense of danger or potential harm, which typically elicits fear in individuals. The sentence implies that someone intends to cause
harm or create a sense of fear in a specific person, indicating a negative and potentially dangerous situation. Therefore, the emotion
expressed in this text is fear.
```

Fig. 7. Generating explanation from GPT3.5 for four input documents.

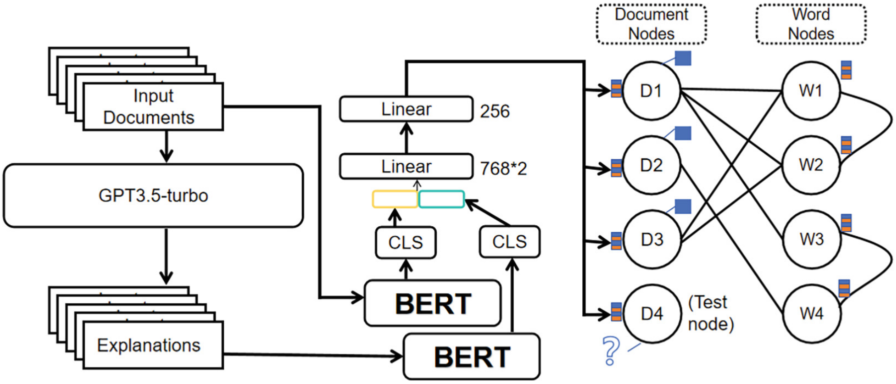


Fig. 8. Integration of GPT generated explanation representations into the BertGCN system.

popular English Reddit subpages and labeled with 27 emotion categories. We mapped those 27 categories to Ekman's 7 emotion classes [3,6] which include anger, disgust, fear, joy, neutral, surprise, and sadness. Comments comprising only a single emoticon were excluded from the dataset reducing its size to about 54K. The dataset was randomly split into the train, validation, and test sets with a ratio of 80:10:10%.

5.2 Evaluated Models

We evaluated and compared several configurations of single-model systems as well as combined GNN/LLM models.

- **GCN:** Only graph model with an adjacency matrix that doesn't include word-to-word edges. The nodes feature matrix is set to identity, $X = I$.
- **BERT:** Bert-based emotion classifier described in Sect. 3.1.
- **RoBERTa:** Same as the BERT model, but using the RoBERTa [14].
- **GPT3.5:** GPT3.5 based classification method described in Sect. 3.2.
- **TextGCN:** Same as GCN model, but with PPMI weighted word-to-word edges in the adjacency matrix. The nodes feature matrix is set to identity, $X = I$.
- **BertGCN:** Combined BERT and GCN model described in Sect. 4.1. The nodes feature matrix is defined as in Eq. (4).
- **RoBERTaGCN:** Same as BertGCN, but using RoBERTa instead of BERT.
- **RoBERTa + GPT3.5:** A model where the classification decision is done using only the concatenated representations of the input text and its GPT3.5 explanation. No GNN is applied.
- **RoBERTaGCN + GPT3.5:** RoBERTaGCN model including the GPT3.5 generated explanation representations as described in Sect. 4.2.

Each model's hyper-parameters were manually optimized on the validation dataset. Still, to ensure a fair comparison between them, we fixed some of the parameters such as GCN or linear layer size and number. The BERT-style models included 2 dense layers with sizes of 768 (or 1536) and 256 (or 7) respectively, and the GNN models had 2 GCN layers. A learning rate of 1e-4 for used for the BERT/RoBERTa fine-tuning, while the GNN parameters were updated with a learning rate of 1e-3. Between the linear layers in all models, there was a dropout layer inserted with a dropout probability of 0.5.

5.3 Results

We evaluated the emotion recognition performance of all the models described in the previous section using standard metrics such as Accuracy, Recall, Precision, and F1-score. Our test set consists of about 5000 sentences. The Precision, Recall, and F1-score results of all the models are shown in Table 1. They are divided into two groups: 1) Single model group including GCN, TextGCN, and BERT, RoBERTa, and GPT3.5 models; and 2) Combined model group which includes BertGCN, RoBERTaGCN, RoBERTa+GPT3.5, and RoBERTaGCN+GPT3.5 models. The results clearly show that the combined models achieve much better performance than the single models. The GPT3.5 only result was lower than our expectations, and this is probably due to the 0-shot prompt configuration. However, when used for explanation generation in the combined models GPT3.5 provides a boost in the performance of up to about 15%. The accuracy scores of all the models are given in Fig. 9.

Table 1. Classification results (%) of all the models in terms of Precision, Recall, and F1-score metrics.

Model type		Precision	Recall	F1-score
Single Model	GCN	61.99	52.33	43.81
	GPT3.5	56.23	48.24	50.98
	TextGCN	57.55	57.77	56.38
	BERT	58.57	68.84	65.25
	RoBERTa	60.15	65.82	64.29
Combined	BertGCN	68.19	68.25	67.96
	RoBERTaGCN	68.31	68.49	68.09
	RoBERTa + GPT3.5	81.76	81.78	81.71
	RoBERTaGCN + GPT3.5	**84.44**	**84.51**	**84.36**

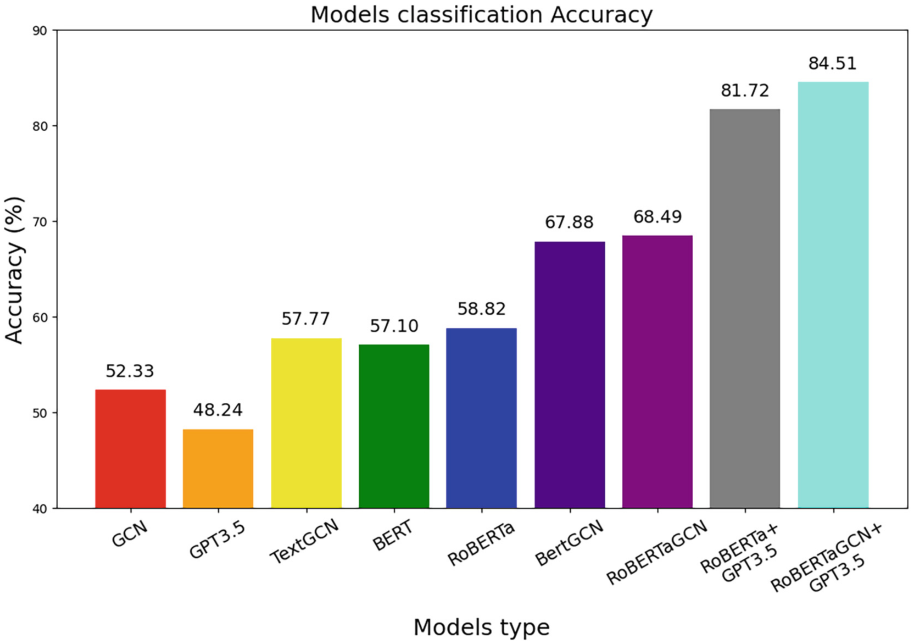


Fig. 9. Emotion classification accuracy (%) of all the models.

6 Conclusion

In this study, we investigated the text-based emotion recognition performance of various GNN-based and LLM-based models as well as several ways to combine them. For the model training and evaluation, we used the GoEmotion dataset reducing the original 27 emotion classes to Ekman's 7 basic categories. All the GNN-based models were trained from scratch while all the LLM-based models but the GPT3.4 were fine-tuned. Evaluation experiments clearly showed that combining GCN and LLM models is effective and improves the performance by up to 20%. Single GCN model results were slightly worse than single BERT or RoBERTa, but higher than GPT3.5 which we attribute to the simple prompt we used. However, when the representations of the GPT3.5 derived explanations were included in the combined models a significant boost in the performance was observed. The reason might be the fact that those explanations were much longer than the input texts and their vector embeddings became easier to discriminate.

There are probably other possible ways to combine the Graph Neural Networks with large language models which based on our findings would also lead to improved performance. The main goal of such approaches is to be able to efficiently merge different views of the input document's information content.

References

1. Adoma, A.F., Henry, N.M., Chen, W.: Comparative analyses of BERT, RoBERTa, DistilBERT, and XLNet for text-based emotion recognition. In: 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pp. 117–121. IEEE (2020)
2. Agarap, A.F.: Deep learning using Rectified Linear Units (ReLU). arXiv preprint arXiv:1803.08375 (2018)
3. Coppini, S., Lucifora, C., Vicario, C.M., Gangemi, A.: Experiments on real-life emotions challenge Ekman's model. Sci. Rep. **13**(1), 9511 (2023)
4. Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., Ravi, S.: GoEmotions: a dataset of fine-grained emotions. arXiv preprint arXiv:2005.00547 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Ekman, P., et al.: Basic emotions. Handb. Cogn. Emot. **98**(45–60), 16 (1999)
7. Fung, P., Ngai, G., Yang, Y., Chen, B.: A maximum-entropy Chinese parser augmented by transformation-based learning. ACM Trans. Asian Lang. Inf. Process. (TALIP) **3**(2), 159–168 (2004)
8. Gao, S., Sethi, A., Agarwal, S., Chung, T., Hakkani-Tur, D.: Dialog state tracking: a neural reading comprehension approach. arXiv preprint arXiv:1908.01946 (2019)
9. Gunes, H., Pantic, M.: Automatic, dimensional and continuous emotion recognition. Int. J. Synthetic Emot. (IJSE) **1**(1), 68–99 (2010)
10. Huang, L., Ma, D., Li, S., Zhang, X., Wang, H.: Text level Graph Neural Network for text classification. arXiv preprint arXiv:1910.02356 (2019)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Li, Y., Chan, J., Peko, G., Sundaram, D.: Mixed emotion extraction analysis and visualisation of social media text. Data Knowl. Eng. **148**, 102220 (2023)
13. Lin, Y., et al.: BertGCN: transductive text classification by combining GCN and BERT. arXiv preprint arXiv:2105.05727 (2021)
14. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

15. Nandwani, P., Verma, R.: A review on sentiment analysis and emotion detection from text. Soc. Netw. Anal. Min. **11**(1), 81 (2021)
16. OpenAI: GPT-3.5: Language model (2021). https://chat.openai.com
17. Shi, W., Xue, G., He, S.: Literature review of network public opinion research from the perspective of sentiment. Doc. Inf. Knowl. **39**(1), 105–118 (2022)
18. Singh, G., Brahma, D., Rai, P., Modi, A.: Text-based fine-grained emotion prediction. IEEE Trans. Affect. Comput. (2023)
19. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, 18–20 October 2019, pp. 194–206. Springer (2019)
20. Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., Ikeuchi, K.: Bias in emotion recognition with ChatGPT. arXiv preprint arXiv:2310.11753 (2023)
21. Wang, X., Zong, C.: Distributed representations of emotion categories in emotion space. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2364–2375 (2021)
22. Xie, Y., Li, J., Pu, P.: Uncertainty and surprisal jointly deliver the punchline: exploiting incongruity-based features for humor recognition. arXiv preprint arXiv:2012.12007 (2020)
23. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7370–7377 (2019)
24. Zhang, X., Zhao, J., LeCun, Y.: Character-level Convolutional Networks for text classification. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
25. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020)