Emotion Detection with GNN

Zhechao Jin, Yifei Li

Introduction

Emotion detection in text is a growing area of interest in Natural Language Processing (NLP) due to its wide-ranging applications in social media analysis, customer experience management, and mental health assessments. The ability to classify sentences into distinct emotions provides organizations and researchers with deeper insights into user behavior, sentiment trends, and overall emotional well-being. Understanding people's emotions, particularly negative ones, is essential for preventing mental health issues and their potential consequences. Tracking societal emotions, especially in response to significant events like new policies or legislation, can provide valuable insights for effective governance. By identifying spikes in negative emotions, authorities can proactively address concerns and mitigate risks such as polarization, political unrest, or even violence (Ismail et al., 2023).

Sentiment analysis and emotion analysis are two approaches that aim to interpret and recognize expressions and emotions conveyed in natural language text. While both methods extract insights from textual data, they differ significantly in their scope and focus. Sentiment analysis primarily categorizes text into three overarching sentiments: **positive**, **negative**, or **neutral**, providing a high-level understanding of the tone of the text. For example, sentiment analysis might determine whether a product review expresses satisfaction or dissatisfaction but does not delve into the underlying emotions.

On the other hand, emotion analysis offers a more nuanced perspective by identifying specific emotions such as **happiness**, **sadness**, **anger**, **surprise**, **fear**, etc. This method allows researchers and organizations to understand the deeper emotional states of users based on the text they provide in conversations, reviews, or social media posts (Madhuri & Lakshmi, 2021). Emotion analysis goes beyond simply categorizing sentiment to capture the intricacies of human emotional expression, making it particularly valuable in areas like mental health, customer experience management, and social media monitoring.

There are two primary methods for identifying emotions in text. The first involves lexicon-based approaches, where predefined dictionaries of words associated with specific emotions are used to classify text. The second method leverages machine learning and advanced techniques like deep learning, including the use of models such as Neural Networks and Transformers, to analyze emotions in a data-driven way (Khan et al., 2022). These models learn patterns and relationships within the text, allowing for more accurate detection of emotions. While sentiment analysis provides a foundational understanding of user intent, emotion analysis offers richer insights into users' inner feelings, which can be critical for applications that require emotional intelligence.

Traditional approaches to emotion detection, such as sequential models like Long short-term memory(LSTMs) or attention-based Transformers, have shown promise but often fail to capture deeper relationships between sentences or documents (Yao et al., 2018). This project leverages Graph Neural Networks (GNNs), a powerful class of models that operate on graph-structured data, to improve emotion detection. By treating sentences as nodes and their relationships (e.g., semantic similarity) as edges, GNNs can propagate and aggregate information

across the graph. This enables a richer understanding of emotional nuances, addressing challenges like overlapping emotions and limited context in short text.

Methods:

Our methodology for emotion detection integrates sentence embeddings, graph construction, and a Graph Neural Network (GNN) to classify sentences into six emotion categories: anger, joy, sadness, fear, love, and surprise. The process involves the following steps:

Sentence Embedding

To represent each sentence numerically, we use a pre-trained SentenceTransformer model, all-MiniLM-L6-v2, which generates a 384-dimensional embedding for each sentence. These embeddings are designed to capture both the semantic meaning and contextual nuances of the text (HuggingFace, n.d.), providing a strong foundation for subsequent graph-based analysis. Unlike traditional word-level embeddings (e.g., Word2Vec, GloVe), sentence embeddings provide a representation for the entire sentence as a single vector, making them particularly well-suited for tasks involving sentence-level understanding, such as emotion detection. The dataset primarily contains short sentences, which may not provide enough context for models relying on word-level embeddings. SentenceTransformers are fine-tuned on tasks like semantic similarity, making them effective for representing the meaning of shorter text snippets.

Graph Construction

We model the dataset as a graph, where: Nodes represent sentences. Edges capture the relationships between sentences, defined by their semantic similarity.

The semantic similarity between two sentences is measured using cosine similarity between their embeddings, which measures the angle between two embedding vectors in a high-dimensional space. Cosine similarity ensures that sentences with similar meanings, regardless of their magnitudes, are treated as related. Edges are created only if the similarity exceeds a predefined threshold, ensuring the graph focuses on meaningful connections. The resulting graph consists of nodes (sentence embeddings), edges (semantic similarity), and edge weights (similarity scores).

Graph Neural Network

We implement a Graph Convolutional Network (GCN) to process the graph and classify each node (sentence) into its respective emotion category. The GCN aggregates information from neighboring nodes, enabling it to capture relationships between sentences and improve classification performance. The architecture includes:

- An input layer that takes the sentence embeddings as node features.
- Graph convolutional layers that propagate and aggregate information across the graph.
- A final fully connected layer to classify each node into one of the six emotion categories.

Training

The model is trained using the training portion of the dataset. Negative Log-Likelihood Loss is used to optimize the model, as it is well-suited for multi-class classification tasks. The

Adam optimizer is applied with a learning rate of 0.01, and dropout regularization is used to prevent overfitting. Unlike traditional models, GNNs leverage graph structures to propagate and aggregate information across nodes. This means that the model learns not only from individual sentence embeddings but also from the relationships between sentences in the graph. The edges and their associated weights play a critical role in influencing the flow of information during training, enhancing the model's ability to classify nuanced emotions. This approach provides a solid foundation for effective emotion detection using graph-based learning.

Evaluation

The model is evaluated on a test set using accuracy as the primary metric. Misclassified examples can be analyzed to identify patterns and refine the model further. This methodology provides a feasible foundation for emotion detection, combining the strengths of sentence embeddings and graph-based learning to address the complexities of the task.

Dataset:

The dataset used in this project, the Emotions dataset for NLP from Kaggle (Praveen, 2024), focuses on classifying short text sentences into one of six emotions: anger, joy, sadness, fear, love, and surprise. The target variable in this dataset is the emotion label, which captures the underlying sentiment of each sentence. The task of classifying emotions goes beyond simple sentiment analysis (positive/negative) by addressing more nuanced emotional states, making this an important application in NLP.

This dataset is particularly valuable because it provides a varied representation of multiple emotion classes, ensuring that models trained on it can generalize well to real-world scenarios where diverse emotional expressions are common. The sentences in the dataset are concise, making it a good testbed for models to understand contextual and semantic nuances. Despite its benefits, analyzing this dataset presents several challenges. First, overlap between emotions can cause misclassifications—for instance, sentences expressing "love" might share linguistic features with "joy," making it difficult for models to distinguish between the two. Second, the dataset contains short sentences with limited context, which makes it harder for models to capture complex emotional cues compared to longer texts. Finally, while the dataset is balanced across classes, it does not account for the ambiguity in human emotions, where the same sentence could evoke different emotions in different people.

The most common approach done by others using this dataset is a Long Short-Term Memory (LSTM) network to process the text sequentially and classify it into various emotion categories. Word embeddings like GloVe are used as input to the LSTM model, which captures long-term dependencies in the text. While LSTMs are good at capturing sequential dependencies, they may struggle with non-linear relationships between words or sentences that aren't in strict sequential order. Another approach was to use TF-IDF to convert textual data into numerical feature vectors. Then, a K-Nearest Neighbors (KNN) classifier is applied to classify each text into one of the emotion categories. The KNN algorithm identifies the emotion of a sentence based on the majority class of its nearest neighbors in the feature space. KNN only considers local neighbors for classification, while GNNs can model both local and global relationships between sentences using graph structures.

Several datasets similar to the Emotions dataset for NLP have been studied in the past, each contributing unique insights into emotion detection and sentiment analysis. One prominent example is the GoEmotions dataset, developed by Google, which contains over 58,000 carefully

labeled Reddit comments spanning 27 distinct emotion categories, including nuanced emotions like "pride" and "embarrassment" (Alon & Ko, n.d.). While GoEmotions provides a broader range of labels compared to the Emotions dataset, it introduces challenges such as overlapping classes and increased complexity, which requires more advanced NLP models and more complex real word applications.

Another widely studied dataset is the ISEAR (International Survey on Emotion Antecedents and Reactions) dataset, which contains self-reported emotional experiences labeled with seven basic emotions such as "anger," "joy," and "fear." Unlike the Emotions dataset, which focuses on short text data, ISEAR entries are typically longer and more descriptive, offering rich contextual information but requiring more preprocessing.

Results

The model achieved an overall accuracy of 71.75% on the test dataset, correctly classifying approximately 1435 samples out of 2000. This accuracy indicates a moderate level of performance in the emotion classification task.

Label	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Support**
Anger	0.640000	0.692913	0.640000	0.665406	275
Fear	0.660714	0.675799	0.660714	0.668172	224
Joy	0.837410	0.724782	0.837410	0.777036	695
Love	0.421384	0.549180	0.421384	0.476868	159
Sadness	0.750430	0.775801	0.750430	0.762905	581
Surprise	0.393939	0.650000	0.393939	0.490566	66
Overall	0.717500				2000

Table 1: Performance Metrics for Emotion Classification by Category

A detailed analysis of performance across individual emotion classes revealed significant variations in the model's ability to classify different emotions:

Anger: The model demonstrates reasonably well performance on anger with balanced precision and recall. F1-Score suggests it maintains a good trade-off between precision and recall but the recall shows it misses some true instances.

Fear: Fear performs slightly better than anger, with both recall and F1-Score around 66%, indicating stable classification.

Joy: Joy is the best-performing category, with high accuracy and F1-Score. The recall indicates the model effectively identifies most joy samples.

Love: Love exhibits the weakest performance among all categories, with low Accuracy and F1-Score. The Recall is especially low, indicating many love samples are missed.

Sadness: Sadness performs well with balanced Precision and Recall, resulting in a stable F1-Score. The model is effective in identifying sadness samples, as indicated by the high recall.

Surprise: Surprise is the weakest category after love, with very low accuracy and F1-Score. The low recall indicates that many surprise samples are not captured by the model.

State-of-the-art (SOTA) methods

Apart from the combination of Sentence-BERT and Graph Neural Network (GNN) used in this project, there are several state-of-the-art (SOTA) methods widely used for emotion classification. These include transformer-based models such as BERT and RoBERTa, which excel at capturing contextual semantics, as well as hybrid approaches that integrate CNNs and RNNs to leverage both local and sequential features. Additionally, multi-task learning frameworks and attention-based mechanisms, like Hierarchical Attention Networks, have also shown strong performance by focusing on shared representations or highlighting important textual elements. Ensemble methods, which combine predictions from multiple architectures, further push the boundaries of performance by leveraging model diversity.

Comparison with Existing Work

The methodology and findings of this project align closely with those presented in the study on emotion classification using GNN. Both works employ Graph Neural Networks (GNNs) alongside transformer-based models, such as Sentence-BERT, to leverage semantic representations for emotion classification. This shared approach underscores the consensus that semantic embeddings capture nuanced emotional content more effectively than syntactic features, particularly in text-based analysis.

Similar to this project, the referenced study demonstrates strong performance in well-represented categories (e.g., Joy and Sadness), while underrepresented emotions (e.g., Love and Surprise) remain challenging due to data imbalance. Both analyses highlight the limitations posed by sparse training data in certain classes, leading to reduced accuracy and recall in those categories.

However, a notable difference lies in the strategies proposed to address these challenges. While the approach applied in this project relies on fixed cosine similarity thresholds for graph construction, the referenced study emphasizes the importance of dynamic graph-building methods and advocates for data augmentation and external knowledge integration as solutions to mitigate data imbalance. These suggestions align with the gaps identified in our work and provide potential avenues for future improvements.

Conclusion

In conclusion, the findings of the referenced study are largely consistent with our results, reinforcing the effectiveness of semantic-driven GNN approaches for emotion classification. The study further identifies key enhancements, such as adaptive graph construction and data enrichment, which could address limitations in handling underrepresented categories and improve overall model performance.

The proposed model succeeded because it combined Sentence-BERT embeddings with GNNs to effectively capture both the semantic nuances of individual sentences and the relational information between them. This hybrid approach allowed the model to leverage the strengths of both transformer-based embeddings and graph-based relational modeling. However, the model struggled with categories like Love and Surprise due to data imbalance and overlapping features with other emotions. Additionally, the fixed similarity threshold used for constructing graph edges may have limited the model's ability to capture complex inter-sentence relationships, leading to misclassifications. These challenges highlight the need for more dynamic graph construction and better handling of underrepresented categories.

This analysis demonstrated the effectiveness of integrating transformer-based models and GNNs for text-based emotion classification. It highlighted the importance of contextual embeddings (from Sentence-BERT) for capturing nuanced emotional meanings and the value of GNNs for modeling inter-sentence relationships. However, it also revealed limitations such as sensitivity to data imbalance and the reliance on fixed similarity thresholds for graph construction. These findings emphasize the importance of balanced datasets and adaptive graph-building techniques in achieving robust performance across all emotion categories.

To further improve the analysis, several enhancements could be implemented. Addressing data imbalance through data augmentation, particularly for underrepresented categories like Love and Surprise, could significantly enhance model performance. A more dynamic approach to graph construction, such as utilizing learnable or context-sensitive methods for edge weight determination instead of fixed similarity thresholds, would better capture complex inter-sentence relationships. Additionally, fine-tuning the Sentence-BERT model on emotion-specific tasks could yield embeddings more aligned with the nuances of this domain. Exploring advanced GNN architectures, such as Graph Attention Networks (GATs) or GraphSAGE, may further improve the model’s ability to represent and differentiate emotional contexts. If multimodal data, such as audio or visual features, becomes available, integrating them would likely complement textual information and provide a more holistic understanding of emotions. Finally, implementing robust evaluation techniques, including cross-validation, would ensure the model generalizes well across diverse datasets, improving reliability and applicability. These steps collectively offer a roadmap for refining the analysis and overcoming current limitations.

References

[1] D. Alon and J. Ko, “Goemotions: A dataset for fine-grained emotion classification,” Google Research, https://research.google/blog/goemotions-a-dataset-for-fine-grained-emotion-classification/ (accessed Nov. 29, 2024).

[2] I. Ameer, N. Bölücü, G. Sidorov, and B. Can, “Emotion classification in texts over graph neural networks: Semantic representation is better than syntactic,” *IEEE Access*, vol. 11, pp. 56921–56934, 2023. doi:10.1109/access.2023.3281544

[3] L. Ismail, N. Shahin, H. Materwala, A. Hennebelle, and L. Frermann, “ML-NLPEMOT: Machine Learning-Natural Language Processing Event-based emotion detection proactive framework addressing mental health,” *IEEE Access*, vol. 11, pp. 144126–144149, 2023. doi:10.1109/access.2023.3343121

[4] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” arXiv.org, https://arxiv.org/abs/1809.05679 (accessed Oct. 25, 2024).

[5] S. I. Khan, F. B. Aziz and M. M. Uddin, "Emotion Detection from Multilingual Text and Multi-Emotional Sentence using Difference NLP Feature Extraction Technique and ML Classifier," *International Journal of Advanced Networking and Applications,* vol. 14, *(3),* pp. 5429-5435, 2022.

[6] Praveen, “Emotions dataset for NLP,” Kaggle, https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp/data (accessed Nov. 30, 2024).

[7] S. Madhuri and S. V. Lakshmi, "Detecting Emotion from Natural Language Text Using Hybrid and NLP Pre-trained Models," *Turkish Journal of Computer and Mathematics Education,* vol. 12, *(10),* pp. 4095-4103, 2021.

[8] “Sentence-transformers/all-minilm-L6-V2 · hugging face,” sentence-transformers/all-MiniLM-L6-v2 · Hugging Face, https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed Nov. 30, 2024).

[9] I. Ameer, N. Bölücü, G. Sidorov and B. Can, "Emotion Classification in Texts Over Graph Neural Networks: Semantic Representation is Better Than Syntactic," in IEEE Access, vol. 11, pp. 56921-56934, 2023, doi: 10.1109/ACCESS.2023.3281544.

Appendix: https://github.com/ZekeJin97/NLP_Final

Statement of contributions:

Evenly distributed, Zhechao mainly for intro+dataset+method, Yifei for related work, results and discussion.