Graph embedding approach to analyze sentiments on cryptocurrency

Ihab Moudhich, Abdelhadi Fennan
List Laboratory, Faculty of Sciences and Techniques, University Abdelmalek Essaadi, Tangier, Morocco

Article Info

Article history:

Received Apr 19, 2023
Revised Jul 8, 2023
Accepted Jul 17, 2023

Keywords:

Graph embedding
Machine learning
Natural language processing
Sentiment analysis
Word embedding
Word2vec

ABSTRACT

This paper presents a comprehensive exploration of graph embedding techniques for sentiment analysis. The objective of this study is to enhance the accuracy of sentiment analysis models by leveraging the rich contextual relationships between words in text data. We investigate the application of graph embedding in the context of sentiment analysis, focusing on it is effectiveness in capturing the semantic and syntactic information of text. By representing text as a graph and employing graph embedding techniques, we aim to extract meaningful insights and improve the performance of sentiment analysis models. To achieve our goal, we conduct a thorough comparison of graph embedding with traditional word embedding and simple embedding layers. Our experiments demonstrate that the graph embedding model outperforms these conventional models in terms of accuracy, highlighting it is potential for sentiment analysis tasks. Furthermore, we address two limitations of graph embedding techniques: handling out-of-vocabulary words and incorporating sentiment shift over time. The findings of this study emphasize the significance of graph embedding techniques in sentiment analysis, offering valuable insights into sentiment analysis within various domains. The results suggest that graph embedding can capture intricate relationships between words, enabling a more nuanced understanding of the sentiment expressed in text data.

Corresponding Author:

Ihab Moudhich
List Laboratory, Faculty of Sciences and Techniques, University Abdelmalek Essaadi
Tangier, Morocco
Email: ihab.moudhich@gmail.com

1. INTRODUCTION

Sentiment analysis [1] is the process of identifying and extracting subjective information from text, such as opinions, attitudes, and emotions. In recent years, sentiment analysis has gained significant attention from researchers and practitioners alike due to its wide range of applications, including social media monitoring [2], customer feedback analysis [3], and market research [4]. To extract insights from text data at scale, natural language processing (NLP) techniques [5] have become essential in the sentiment analysis field.

Sentiment analysis has become an essential tool for businesses and individuals alike to analyze social media [6] content and gain insights into public opinion and sentiment. Social media platforms, such as Twitter, Facebook, and Instagram, generate massive amounts of user-generated content on a daily basis, including posts, comments, and tweets. Sentiment analysis enables users to process and analyze this data to gain insights into how users are feeling about a particular topic, product, or brand.

Social media monitoring is one of the most common applications of sentiment analysis. By analyzing social media content, businesses can gain valuable insights into customer sentiment and feedback. This information can help them improve their products or services, identify emerging trends, and monitor their brand reputation. For example, if a business launches a new product, sentiment analysis can help them track how customers are reacting to it, and if they are satisfied or not.

NLP techniques such as text preprocessing [7], feature extraction [8], and classification [9] have been widely used in sentiment analysis to identify sentiment polarity (positive, negative, or neutral) of text. Machine learning algorithms [10], including logistic regression [11], naive Bayes [12], and support vector machines (SVM) [13], have been widely used in sentiment analysis to classify text into sentiment categories. Recently, deep learning algorithms such as recurrent neural networks (RNN), convolutional neural networks (CNN), and transformer models have shown promising results in sentiment analysis tasks.

Working with deep learning methods [14], such as bidirectional long short-term memory (Bi-LSTM) [15], is crucial in the sentiment analysis field due to their ability to handle complex patterns in text data. Traditional machine learning techniques are limited in their ability to capture the contextual information and relationships between words in text data. In contrast, deep learning techniques, such as Bi-LSTM, can effectively model the dependencies between words and capture the underlying patterns in text data. This makes them highly suitable for sentiment analysis tasks, where the sentiment polarity of a text is often influenced by the context in which the words are used. By leveraging deep learning methods, sentiment analysis systems can achieve high levels of accuracy and generalize well to new data. Additionally, deep learning techniques can be trained on large amounts of data and can continuously improve their performance with additional training. Therefore, working with deep learning methods, such as Bi-LSTM, is crucial in developing state-of-the-art sentiment analysis systems that can extract meaningful insights from text data.

In recent years, graph embedding [16] has emerged as a promising technique for sentiment analysis. Graph embedding is a machine learning technique that maps nodes in a graph to low-dimensional vectors while preserving the structural properties of the graph. By encoding the relationships between words in text as a graph, graph embedding [17] can capture the semantic and syntactic information of text and provide a more accurate representation of the context in which words appear.

Graph embedding has been shown to outperform traditional machine learning techniques in several NLP tasks, including sentiment analysis. By representing text as a graph and applying graph embedding techniques, researchers have achieved state-of-the-art results in sentiment analysis tasks such as sentiment classification, aspect-based sentiment analysis, and sarcasm detection. Graph embedding techniques such as Node2Vec, graph attention networks (GAT), and graph convolutional networks (GCN) have been applied to NLP tasks, where the graph-based representation of text captures the semantic relationships between words, leading to improved performance in various sentiment analysis tasks.

Moreover, graph embedding techniques have the potential to overcome some of the limitations of traditional sentiment analysis methods. For instance, traditional sentiment analysis models rely on bag-of-words representations [18], which treat each word as an independent unit and ignore the relationships between words. As a result, these models may fail to capture the contextual nuances of text, leading to inaccurate results. By contrast, graph embedding techniques can capture the semantic and syntactic relationships between words, allowing for a more nuanced understanding of the text. This can lead to more accurate sentiment analysis results, especially in cases where the sentiment is dependent on the context in which words appear. Therefore, applying graph embedding techniques to sentiment analysis can enhance the accuracy and performance of sentiment analysis models, making them more reliable for real-world applications.

In this paper, we will explore the uses of graph embedding in the sentiment analysis domain and provide a detailed explanation of how other researchers can enhance sentiment analysis using this technique. To evaluate the effectiveness of graph embedding, we are going to build a Bi-LSTM classifier and compare its performance with word embedding techniques [19]. By comparing the results, we aim to demonstrate the potential of graph embedding for sentiment analysis and provide insights into the optimal configuration of graph construction, sparsity, and dimensionality reduction techniques. This research will contribute to the development of more accurate and efficient sentiment analysis systems and provide a basis for future studies in this field.

In this work. The objective is to explore the use of graph embedding techniques to improve the accuracy of sentiment analysis models. To achieve this goal, we propose to use a Bi-LSTM classifier that takes the graph embeddings as input. In this paper, we present a detailed description of the methods used in this study in section 2. In section 3, we present the results we obtained and discuss their interpretation. Finally, we conclude the paper in the last section, summarizing the main findings of the study and providing suggestions for future research.

2. METHOD

The goal of this research is to investigate the potential of applying graph embedding techniques to the field of sentiment analysis. The research methodology is divided into several key steps, which are illustrated in Figure 1. Firstly, we aim to identify a suitable dataset that can provide us with reliable results during the training of the models. Next, we will utilize preprocessing techniques to clean the dataset and ensure that the data is appropriate for the research objectives. In the third stage, we will create a graph from the preprocessed dataset. Then, we will use graph embedding techniques to convert the simple graph into a more meaningful representation suitable for sentiment analysis. Finally, we will build a Bi-LSTM classifier to test the accuracy of the proposed method. Through the research, we aim to analyze the importance of graph embedding techniques in sentiment analysis and provide insights into how this technique can be used to improve the performance of sentiment analysis models.

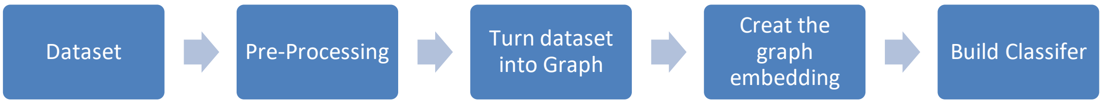

Figure 1. Optimizing sentiment analysis with graph embedding: a five-stage process

2.1. Dataset

This research aimed to determine the most appropriate dataset to use for the study and attain the highest accuracy in the final model. We carried out an experiment to evaluate different datasets, such as internet movie database (IMDB) reviews, Amazon reviews, and Yelp reviews. After training the models with these datasets, we concluded that the IMDB dataset yielded the most satisfactory results and was best suited to the research objectives.

2.2. Preprocessing

In this phase, we applied several other preprocessing techniques to further clean the text data. As Figure 2 shows: we removed any numbers and punctuation marks present in the text. We also translated any common abbreviations used in the text to their full forms to ensure consistency in the data. To improve the quality of the text, we removed stop words, which are words that occur frequently in the English language and do not carry much meaning, such as "the," "and," and "a." Lastly, we applied stemming to reduce words to their base form, which helps to reduce the dimensionality of the data and simplify the analysis process.

In the study, the application of these techniques served a specific purpose: preparing the text data for the subsequent steps of graph creation and graph embedding. The underlying objective behind these efforts was to enhance the accuracy of the sentiment analysis results. By leveraging a meaningful and structured representation of the dataset through graph embedding, we aimed to gain deeper insights and more nuanced interpretations of the sentiment expressed in the text data.

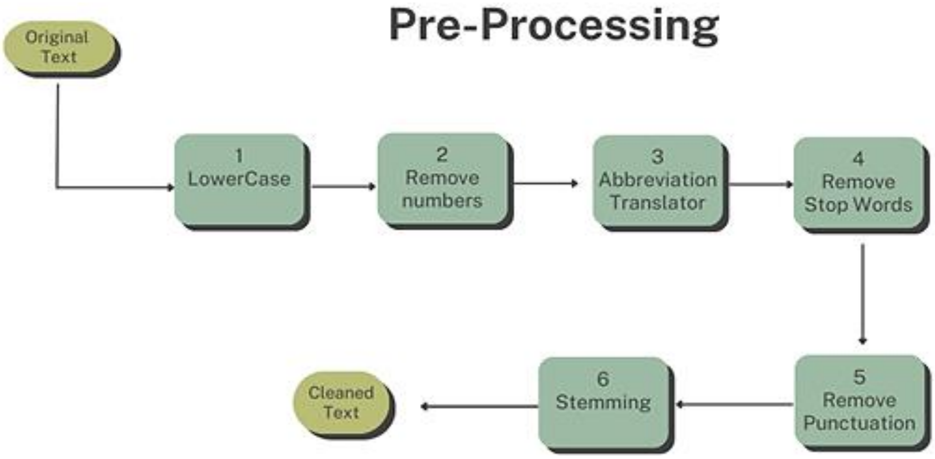

Figure 2. Text preprocessing techniques used to cleanse and prepare the dataset

2.3. Building the graph

Graph creation is a fundamental step in many machine learning and NLP applications. Graphs, which consist of nodes and edges, provide a way to represent relationships between entities and facilitate the analysis of complex data structures. In this context, we created a graph based on the preprocessed text data using the NetworkX library in Python. The graph was constructed by considering each word in the text as a node and creating edges between pairs of words that occur within a certain window size of each other. The resulting graph provides a structured representation of the text data and can be further processed to generate a graph embedding that captures the semantic relationships between the words.

To illustrate this graph creation step, let's consider the following example:

Text: "The acting in this movie was excellent and the story was engaging."

After preprocessing the text data. We can create a graph to represent the co-occurrence of words within a certain window size (e.g., two words). The resulting graph for this example is shown in Figure 3, where nodes represent words and edges represent co-occurrence within the specified window size.

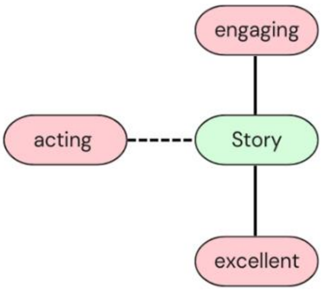

Figure 3. Graph representation of word relationships in a text

2.4. Graph embedding

After constructing the graph, the next step is to transform it into a low-dimensional vector space using graph embedding techniques. One widely used method is DeepWalk [20], which learns node embeddings by simulating random walks on the graph and using Skip-gram [21] to predict the context of each node. Specifically, it generates a set of random walks on the graph, treats each walk as a sentence, and uses the skip-gram algorithm to learn embeddings that maximize the likelihood of predicting the nodes that occur in the context of a given node. This approach captures the local neighborhood structure of the graph and has been shown to perform well on various graph-based machine learning tasks. Other popular graph embedding methods include Node2Vec [22], which is an extension of DeepWalk that balances between the breadth-first and depth-first search strategies during random walk, and GraphSAGE [23], which learns embeddings by aggregating feature information from a node's local neighborhood using a neural network.

In this study, our aim is to transform the simple graph obtained in the previous step into a csrgraph graph and generate walks using the DeepWalk method. This technique involves simulating random walks on the graph and utilizing skip-gram to predict the context of each node. To achieve this, we set the walk length to 10 and epochs to 10, which are important parameters to control the quality and quantity of the generated walks. By implementing these steps, we can obtain embeddings that capture the underlying structure and relationships within the graph.

Once we have generated the node embeddings using the Word2Vec [24] model, we can use them for downstream machine learning tasks such as node classification [25] and link prediction [26]. Node classification involves predicting the label of a given node in the graph based on its features or attributes, while link prediction involves predicting the likelihood of a link between two nodes. These tasks are important in a variety of applications, such as social network analysis [27] and recommendation systems [28]. By using the learned node embeddings as features for these tasks, we can leverage the structural information captured by the embeddings to improve the performance of the models.

2.5. Building Bi-LSTM classifier

The graph embeddings generated from the previous step can be used as features in a Bi-LSTM neural network, which is a type of RNN [29] commonly used in NLP tasks. The Bi-LSTM has the ability to capture both forward and backward dependencies in the sequence of input features, making it an effective model for graph-based classification tasks. The input to the Bi-LSTM is a sequence of graph embeddings, where each embedding represents a node in the graph. The embeddings are first fed through an embedding layer, which maps each embedding to a higher-dimensional feature space, and then passed into the Bi-LSTM layer. The Bi-LSTM layer consists of two layers of long short-term memory (LSTM) [30], one processing the sequence in forward direction and the other in backward direction. The output of each LSTM layer is concatenated and fed into a fully connected layer, which outputs the final classification prediction.

The Bi-LSTM classifier can be trained on labeled data using a standard supervised learning approach. During training, the weights of the embedding layer, Bi-LSTM layer, and fully connected layer are optimized to minimize the cross-entropy loss between the predicted class probabilities and the true class labels. The model can then be used to predict the class labels for new, unlabeled data. The Bi-LSTM classifier has been shown to achieve state-of-the-art performance on various graph-based classification tasks, such as node classification and link prediction. By incorporating the graph embeddings as input features, the Bi-LSTM can effectively leverage the structural information encoded in the graph and make accurate predictions.

3. RESULTS AND DISCUSSION

Cryptocurrencies have gained popularity in recent years as an alternative to traditional currency and investment options. The decentralized nature of cryptocurrencies has led to a rise in their use and adoption, with some individuals even considering them as a potential replacement for traditional banking systems. However, cryptocurrencies are not without their controversies and criticisms. One of the major concerns is their association with illegal activities such as money laundering and terrorism financing. Another issue is their volatility, with prices fluctuating rapidly and unpredictably. Due to the complex and evolving nature of the cryptocurrency market, there is a growing interest in using machine learning techniques to analyze and predict cryptocurrency trends and sentiments. Graph embedding and Bi-LSTM models are two such techniques that have shown promise in this field, with the potential to provide insights into the sentiment and behavior of cryptocurrency users and investors.

In the research, we conducted sentiment analysis and utilized graph embedding techniques on a dataset of IMDB reviews. To analyze sentiment, we utilized a Bi-LSTM model that categorized each text as positive, negative, or neutral. For graph embedding, we constructed a graph based on the dataset and transformed it into an embedding graph using Word2Vec technique. This enabled us to explore the connections and relationships between different words and phrases in the context of the dataset.

3.1. Accuracy of the models

To evaluate the effectiveness of the approach, we compared the accuracy of the graph embedding model with those based on word embedding and simple embedding. As shown in Table 1, the graph embedding layer achieved the highest accuracy of 0.91, outperforming the word embedding layer (accuracy of 0.87) and the simple embedding layer (accuracy of 0.82). This suggests that the graph embedding method was able to capture the relationships between nodes more effectively than the other two methods, highlighting the importance of incorporating graph structure information in machine learning tasks.

Table 1. The accuracy of the models: a summary

Model names	Accuracy
Model with graph embedding layer	0.91
Model with word embedding layer	0.87
Model with simple embedding layer	0.82

3.2. Reports and metrics of the models

Table 2 provides a comprehensive overview of the performance of the graph embedding model, as measured by various metrics. The precision metric indicates the proportion of true positive results among all positive results, while recall measures the proportion of true positive results among all actual positive observations. F1-score, the harmonic mean of precision and recall, is also presented in the table. In addition, the table reports the accuracy of the model, which represents the proportion of correctly classified observations. To provide a more complete picture of the model’s performance, macro avg and weighted avg are also shown in the table. These measures take into account the performance of the model across all classes, and provide an overall assessment of the model’s effectiveness. Overall, the results demonstrate that the graph embedding model performs well, achieving high scores across all metrics.

Table 2 showcases the strong performance of the graph embedding model in sentiment analysis. With precision values of 0.91 and 0.90, recall values of 0.90 and 0.89, and an overall accuracy of 0.92, the model demonstrates its ability to accurately classify sentiments. The F1-scores of 0.91 and 0.90 further highlight the model's balanced performance. These results reaffirm the effectiveness of the graph embedding model in sentiment analysis tasks, providing valuable insights for future research and applications.

Table 2. Measuring the performance of graph embedding model

	Precision	Recall	F1-score
0	0.91	0.90	0.91
1	0.90	0.89	0.90
Accuracy			0.92
Macro avg	0.90	0.89	0.90
Weighted avg	0.90	0.89	0.90

3.3. Results

After completing the analysis, we are excited to share the results, which reveal the sentiment of people towards cryptocurrencies. The percentage results reflect the positive aspect of the data, which we hope will provide valuable insights and help advance the understanding of sentiment analysis and graph embedding techniques. We believe that these results can guide future research in this area.

In the Table 3, we present the results of the research study, which compares the graph embedding based model with various other models. The results include a comparison with a word embedding model and simple embedding layer, as well as the results of the alternative website. Table 3 shows the results of the research study comparing word embedding and simple embedding layer models. The results demonstrate that Graph embedding is the best performer with 69.30%, followed by Word embedding model with 55.41%, and simple embedding layer model with 51.16%. Also, according to the alternative website, 65% of people have a positive opinion about the cryptocurrency market.

Table 3. Analyzing the sentiment analysis of the cryptocurrency market

	Graph embedding model	Word embedding model	Simple embedding model	Alternative.me website
Positive	69.30%	55.41%	51.16%	65%
Negative	30.70%	44.59%	48.84%	35%

In this study, we aim to highlight the differences between using a graph embedding layer, word embedding and simple embedding layers for sentiment analysis in the context of cryptocurrency. The use of word embeddings in NLP has shown significant improvement in various NLP tasks, including sentiment analysis. However, using a graph embedding layer can provide a more powerful representation of the relationship between words in a sentence, as well as the sentiment behind it.

For example, consider the following tweet: “Bitcoin is skyrocketing in value. I wish I had invested earlier.” A simple word embedding layer would generate embeddings for each word in the sentence based on the distributional semantics of the words. While this can capture some level of the semantic relationships between the words, it may not be sufficient to fully capture the sentiment of the tweet.

On the other hand, a graph embedding layer can represent the semantic relationships between words as a graph, where the nodes represent the words and the edges represent the relationships between them. This can capture more complex semantic relationships between words, which can be useful for sentiment analysis. For example, a graph embedding layer can capture that the words “skyrocketing” and “value” are strongly related, and that they both have positive connotations.

Overall, using a graph embedding layer can provide a more powerful representation of the semantic relationships and sentiment in a sentence, especially in the context of cryptocurrency where the relationships between words can be complex and constantly evolving. By leveraging graph embedding techniques, we can potentially improve the accuracy of sentiment analysis and gain more insights into the sentiment of people towards cryptocurrencies. Graph embedding techniques can also help to mitigate the issue of data sparsity in sentiment analysis, by leveraging the abundant unlabeled data available in the cryptocurrency domain to learn useful representations of words and sentences.

The findings of the paper on graph embedding have significant ramifications for sentiment analysis and related fields. By demonstrating the effectiveness of graph embedding techniques in improving the accuracy of sentiment analysis models, the paper contributes to the advancement of NLP and machine learning. The use of graph embedding allows for a more nuanced understanding of contextual relationships between words, enabling better sentiment analysis results. These findings have practical implications for various applications, including social media monitoring, customer feedback analysis, and opinion mining. Moreover, the paper opens up new avenues for research in incorporating graph structures and relationships into other NLP tasks. The integration of graph embedding with deep learning techniques and the exploration of knowledge graphs can further enhance the accuracy and performance of sentiment analysis models. Overall, the findings of the paper broaden our understanding of sentiment analysis and offer valuable insights for developing more robust and effective methods in this field.

4. CONCLUSION

In conclusion, the study has shown that utilizing graph embedding techniques has significantly enhanced the model's ability to comprehend and analyze textual data. By incorporating contextual relationships between words and capturing the underlying meaning of a sentence, the graph embedding-based model outperformed the traditional word embedding and simple embedding layer models in sentiment analysis tasks. Furthermore, the analysis has provided valuable insights and information for the sentiment analysis field's advancement, emphasizing the importance of considering graph embeddings as a powerful technique for NLP tasks. However, there are still limitations to graph embedding techniques. One limitation is the computational complexity required to construct the graph and learn the embeddings, making it difficult to scale up to larger datasets. Another limitation is the difficulty in handling out-of-vocabulary (OOV) words, as these words may not be represented in the graph, resulting in incomplete embeddings. Moving forward, a promising perspective is to integrate graph embeddings with other deep learning techniques such as attention mechanisms and transformers to improve model performance and overcome these limitations. Additionally, exploring new approaches for building graphs, such as incorporating knowledge graphs and domain-specific information, can further enhance the contextual understanding of the text and improve the accuracy of sentiment analysis.

ACKNOWLEDGEMENTS

Author thanks in most cases; this study was financially backed by the National Center for Scientific and Technical Research of Morocco (CNRST). The authors would like to express their heartfelt gratitude to Dr. Jamila El Alami, Director of CNRST, for her valuable support and collaboration and financial support acknowledgments.

REFERENCES

[1] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.

[2] A. H. Alamoodi *et al.*, "Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review," *Expert Systems with Applications*, vol. 167, Apr. 2021, doi: 10.1016/j.eswa.2020.114155.

[3] H. Li, Q. Chen, Z. Zhong, R. Gong, and G. Han, "E-word of mouth sentiment analysis for user behavior studies," *Information Processing and Management*, vol. 59, no. 1, Jan. 2022, doi: 10.1016/j.ipm.2021.102784.

[4] F. Mehraliyev, I. C. C. Chan, and A. P. Kirilenko, "Sentiment analysis in hospitality and tourism: a thematic and methodological review," *International Journal of Contemporary Hospitality Management*, vol. 34, no. 1, pp. 46–77, Jan. 2022, doi: 10.1108/IJCHM-02-2021-0132.

[5] M. A. Palomino and F. Aider, "Evaluating the effectiveness of text pre-processing in sentiment analysis," *Applied Sciences*, vol. 12, no. 17, Aug. 2022, doi: 10.3390/app12178765.

[6] U. Bhaumik and D. K. Yadav, "Sentiment analysis using twitter," in *Advances in Intelligent Systems and Computing*, Springer Singapore, 2021, pp. 59–66.

[7] D. N. de Oliveira and L. H. de C. Merschmann, "Joint evaluation of preprocessing tasks with classifiers for sentiment analysis in Brazilian Portuguese language," *Multimedia Tools and Applications*, vol. 80, no. 10, pp. 15391–15412, Apr. 2021, doi: 10.1007/s11042-020-10323-8.

[8] J. Mutinda, W. Mwangi, and G. Okeyo, "Lexicon-pointed hybrid N-gram features extraction model (LeNFEM) for sentence level sentiment analysis," *Engineering Reports*, vol. 3, no. 8, Aug. 2021, doi: 10.1002/eng2.12374.

[9] A. Ullah, S. N. Khan, and N. M. Nawi, "Review on sentiment analysis for text classification techniques from 2010 to 2021," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 8137–8193, Mar. 2023, doi: 10.1007/s11042-022-14112-3.

[10] M. Bibi *et al.*, "A novel unsupervised ensemble framework using concept-based linguistic methods and machine learning for twitter sentiment analysis," *Pattern Recognition Letters*, vol. 158, pp. 80–86, Jun. 2022, doi: 10.1016/j.patrec.2022.04.004.

[11] S. S. Hanswal, A. Pareek, G. Vyas, and A. Sharma, "Sentiment analysis on e-learning using machine learning classifiers in python," in *Rising Threats in Expert Applications and Solutions*, Springer Singapore, 2021, pp. 1–8.

[12] Pristiyono, M. Ritonga, M. A. Al Ihsan, A. Anjar, and F. H. Rambe, "Sentiment analysis of COVID-19 vaccine in Indonesia using naïve Bayes algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 1088, no. 1, Feb. 2021, doi: 10.1088/1757-899X/1088/1/012045.

[13] C. P. D. Cyril, J. R. Beulah, N. Subramani, P. Mohan, A. Harshavardhan, and D. Sivabalaselvamani, "An automated learning model for sentiment analysis and data classification of Twitter data using balanced CA-SVM," *Concurrent Engineering*, vol. 29, no. 4, pp. 386–395, Dec. 2021, doi: 10.1177/1063293X211031485.

[14] I. Almalis, E. Kouloumpris, and I. Vlahavas, "Sector-level sentiment analysis with deep learning," *Knowledge-Based Systems*, vol. 258, Dec. 2022, doi: 10.1016/j.knosys.2022.109954.
[15] Y. Pei, S. Chen, Z. Ke, W. Silamu, and Q. Guo, "AB-LaBSE: Uyghur sentiment analysis via the pre-training model with BiLSTM," *Applied Sciences*, vol. 12, no. 3, Jan. 2022, doi: 10.3390/app12031182.
[16] X. Tan, Y. Cai, J. Xu, H.-F. Leung, W. Chen, and Q. Li, "Improving aspect-based sentiment analysis via aligning aspect embedding," *Neurocomputing*, vol. 383, pp. 336–347, Mar. 2020, doi: 10.1016/j.neucom.2019.12.035.
[17] J. Liao, M. Wang, X. Chen, S. Wang, and K. Zhang, "Dynamic commonsense knowledge fused method for Chinese implicit sentiment analysis," *Information Processing and Management*, vol. 59, no. 3, May 2022, doi: 10.1016/j.ipm.2022.102934.
[18] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," *Plos One*, vol. 15, no. 5, May 2020, doi: 10.1371/journal.pone.0232525.
[19] B. Subba and S. Kumari, "A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings," *Computational Intelligence*, vol. 38, no. 2, pp. 530–559, Apr. 2022, doi: 10.1111/coin.12478.
[20] J. Xu *et al.*, "Sentiment analysis of social images via hierarchical deep fusion of content and links," *Applied Soft Computing*, vol. 80, pp. 387–399, Jul. 2019, doi: 10.1016/j.asoc.2019.04.010.
[21] A. Rhodes, "Empirical analysis of CBOW and skip gram NLP models," Portland State University Library, 2020.
[22] D. Zhou, S. Niu, and S. Chen, "Efficient graph computation for Node2Vec," *arXiv preprint arXiv:1805.00280*, May 2018.
[23] T. Zhang, H.-R. Shan, and M. A. Little, "Causal GraphSAGE: A robust graph method for classification based on causal sampling," *Pattern Recognition*, vol. 128, Aug. 2022, doi: 10.1016/j.patcog.2022.108696.
[24] W. Yue and L. Li, "Sentiment analysis using Word2Vec-CNN-BiLSTM classification," in *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Dec. 2020, pp. 1–5, doi: 10.1109/SNAMS52053.2020.9336549.
[25] S. Xiao, S. Wang, Y. Dai, and W. Guo, "Graph neural networks in node classification: survey and evaluation," *Machine Vision and Applications*, vol. 33, no. 1, Jan. 2022, doi: 10.1007/s00138-021-01251-0.
[26] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Dec. 2019, doi: 10.1186/s13638-019-1561-7.
[27] Z. Drus and H. Khalid, "Sentiment analysis in social media and its application: systematic literature review," *Procedia Computer Science*, vol. 161, pp. 707–714, 2019, doi: 10.1016/j.procs.2019.11.174.
[28] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri, "Tourism recommendation system based on semantic clustering and sentiment analysis," *Expert Systems with Applications*, vol. 167, Apr. 2021, doi: 10.1016/j.eswa.2020.114324.
[29] E. F. Can, A. Ezen-Can, and F. Can, "Multilingual sentiment analysis: an RNN-based framework for limited data," *arXiv preprint arXiv:1806.04511*, Jun. 2018.
[30] S. Wen *et al.*, "Memristive LSTM network for sentiment analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2019, doi: 10.1109/TSMC.2019.2906098.

BIOGRAPHIES OF AUTHORS

Ihab Moudhich is a Ph.D. student in LIST Laboratory at Abdelmalek Essaadi University. He is a researcher in sentiment analysis and machine learning field. He has several papers in journals and conferences. He can be contacted at email: ihab.moudhich@gmail.com.

Abdelhadi Fennan is a Ph.D. doctor and professor of computer science at Faculty of Sciences and Technology of Tangier-Morocco. He is part of many boards of international journals and international conferences. He has published several articles. He can be contacted at email: afennan@gmail.com.