Evaluating sentiment analysis and word embedding techniques on Brexit

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

Article Info

Article history:

Received Mar 3, 2023
Revised Jun 13, 2023
Accepted Jul 21, 2023

Keywords:

FastText
Machine learning
Sentiment analysis
Word embedding
Word2vec

ABSTRACT

In this study, we investigate the effectiveness of pre-trained word embeddings for sentiment analysis on a real-world topic, namely Brexit. We compare the performance of several popular word embedding models such global vectors for word representation (GloVe), FastText, word to vec (word2vec), and embeddings from language models (ELMo) on a dataset of tweets related to Brexit and evaluate their ability to classify the sentiment of the tweets as positive, negative, or neutral. We find that pre-trained word embeddings provide useful features for sentiment analysis and can significantly improve the performance of machine learning models. We also discuss the challenges and limitations of applying these models to complex, real-world texts such as those related to Brexit.

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 commonly used in the context of social media, as digital communication networks produce a significant amount of written content, it can be examined to discern the attitudes of those who utilize it. This can include analyzing the overall sentiment [2] of a particular brand or product or identifying sentiment towards specific topics or events. There are several challenges in applying sentiment analysis to social media data [3], including the informal and often abbreviated nature of the text, as well as the presence of slang, misspellings, and other forms of non-standard language. However, with the use of advanced natural language processing techniques [4], it is possible to accurately identify the sentiment of social media [5] posts and use this information to gain insights about the attitudes and opinions of users.

Sentiment analysis is a subfield of natural language processing that centers on utilizing machine-learning techniques [6] to recognize and extract objective information from written content. This information can include the emotional tone of the text, as well as the overall sentiment (positive, neutral, or negative) expressed by the writer. By applying sentiment analysis to large datasets of text [7], such as social media posts or customer feedback, organizations can gain insights into the opinions and emotions of their audience.

One of the main advantages of sentiment analysis is its ability to help organizations make decisions that are more informed by providing them with a deeper understanding of their customers needs and preferences. For instance, a company might use sentiment analysis to analyze customer feedback [8] and identify common trends or patterns that could be used to improve their products or services. Furthermore, sentiment analysis can be used to monitor social media platforms for mentions of a particular brand or product, allowing companies to quickly respond to customer complaints or concerns.

Sentiment analysis plays a crucial role in extracting valuable insights from vast volumes of text data. Leveraging machine learning techniques [9], organizations can effectively identify and analyze the sentiment expressed within text, enabling them to make more informed decisions. Moreover, this analytical approach empowers companies to enhance their products and services based on the feedback and sentiments expressed by their customers.

Sentiment analysis frequently employs machine-learning techniques to automatically discern the attitude in written content. These approaches are educated on a vast dataset of annotated text, where the annotations indicate the emotion of the text (e.g. positive, negative, and neutral). The machine-learning method [10] utilizes this training data to learn the patterns that are connected with diverse emotions, and can then be used to new, unseen text data to predict the emotion of the text.

In sentiment analysis, a wide range of machine learning algorithms can be employed [11]. These encompass traditional classification methods like support vector machines and decision trees, alongside more advanced neural networks including long short-term memory (LSTM) networks and convolutional neural networks (CNNs) [12]. The selection of the most suitable algorithm hinges on factors like the dataset's unique characteristics and the performance objectives set for the sentiment analysis system.

One of the most popular methods to represent words is known as word embedding [13]. Word embedding is a technique for representing words as vectors in a high-dimensional space. These word vectors capture the semantic meaning of the words, and the position of the vector in the space encodes the meaning of the word. Word embedding is a vital aspect of numerous natural languages processing assignments, including opinion mining and machine interpretation.

By using word embedding in conjunction with sentiment analysis [14], the sentiment analysis model can learn to associate specific words or phrases with certain sentiments. For example, a word-embedding model may learn to associate the word "terrible" with negative sentiment, while associating the word "wonderful" with positive sentiment. This can help the sentiment analysis model to predict the sentiment of a piece of text more accurately, even if the text contains words or phrases that the model has not seen before.

Pre-trained word embedding models serve as a valuable resource for natural language processing tasks [15], such as sentiment analysis, by utilizing their training on extensive text datasets. These models come equipped with a comprehensive understanding of semantic relationships between words, allowing them to offer meaningful word representations in the form of word vectors. As a result, they serve as a convenient starting point for sentiment analysis, enabling researchers and practitioners to leverage the pre-existing knowledge encoded within these models to enhance their sentiment analysis algorithms.

Employing pre-trained word embedding models in sentiment analysis can aid to augment the effectiveness of the emotion recognition model. Because the pre-trained word-embedding model has already learned the semantic connections between words, it can supply useful information to the sentiment analysis model regarding the significance of words and phrases in the text data. This can assist the emotion recognition model to recognize the emotion of the text more accurately.

A wide range of pre-trained word embedding models are readily accessible for various natural language processing tasks. Among the popular options are word to vec (word2vec) [16], global vectors for word representation [17], embeddings from language models (ELMo) [18], and FastText [19], each offering unique advantages and capturing different aspects of word semantics. These well-known pre-trained models have been widely adopted by researchers and practitioners to facilitate tasks like sentiment analysis, providing a solid foundation for understanding word meanings and contextual relationships within textual data.

We posit that integrating pre-trained word embeddings like GloVe, word2vec, ELMo, and FastText into sentiment analysis tasks will enhance accuracy and effectiveness. These embeddings, trained on extensive datasets, have already encapsulated semantic word relationships. By incorporating them into our sentiment analysis model, we anticipate improved accuracy in identifying and classifying sentiments compared to using a basic word-embedding layer. In this study, we aim to work and compare the results of the different pre-trained word embedding models on Brexit data, which are used here [20]. This paper is organized as such: i) Section 2 will list all the methods used in this study in detail; ii) Section 3 will summarize the results we got and their interpretation; and iii) The last section is a round-up of the paper and will conclude the paper.

2. METHOD

In this research, we aim to develop various methods to compare word-embedding techniques in the field of sentiment analysis [21]. We divide our architecture into 5 fundamental stages as illustrated in Figure 1. The first step is to identify an appropriate dataset [22] that can yield good results during the training of our models. Next, we proceed to preprocessing [23] with the aim of cleaning our data without damaging the accuracy of the final models. In the third stage, we commence building the various types of word vector representations [24] for our embedding layer to be used as input for the fourth stage, based on the pre-trained model. In the fourth stage, we create a neural network classifier [25] to train our final model that is based on the prior dataset and the vectors of the embedding layer. Finally, we apply the classifier to Brexit data, allowing us to compare it with the results of our previous study.

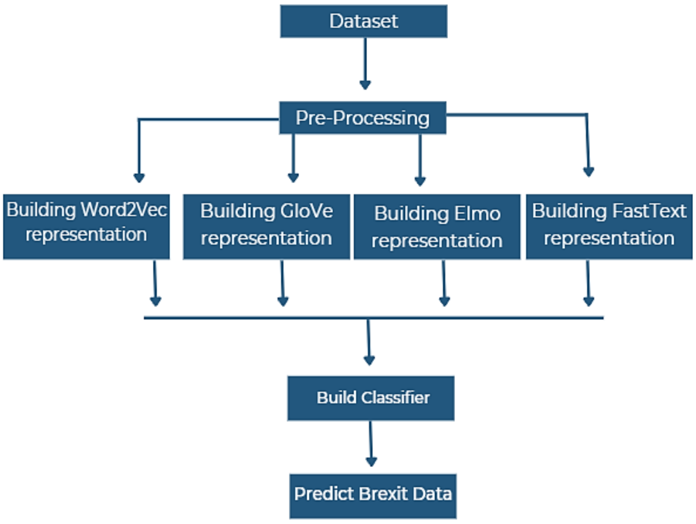

Figure 1. The foundational elements of our structures

2.1. Dataset

Within the scope of this study, we designed an experiment to assess different datasets with the aim of identifying the most appropriate one for our specific objectives, ultimately yielding the highest accuracy in our final model. Through the training process involving diverse datasets such as tweets, internet movie database (IMDb) reviews, Amazon reviews, and Yelp reviews, we determined that the tweets dataset emerged as the most optimal choice, delivering exceptional results that aligned closely with our requirements. Consequently, the utilization of the tweet's dataset proved to be instrumental in achieving our desired outcomes within this study.

2.2. Preprocessing

Before we start the main step of creating the models, we should apply preprocessing techniques to our dataset as described in Figure 2. We started by converting the text to lowercase. Then, we applied some regex to delete any HTML tags or links. Meanwhile, we tried to clean any specific characters such as numbers and punctuation. Moreover, before we applied lemmatization to the text, we tokenized the sentence. In this work, we tried not to complicate the preprocessing stage with other stemming and filtering techniques because it is hard to improve the final accuracy of the model when we apply many filters to one text.

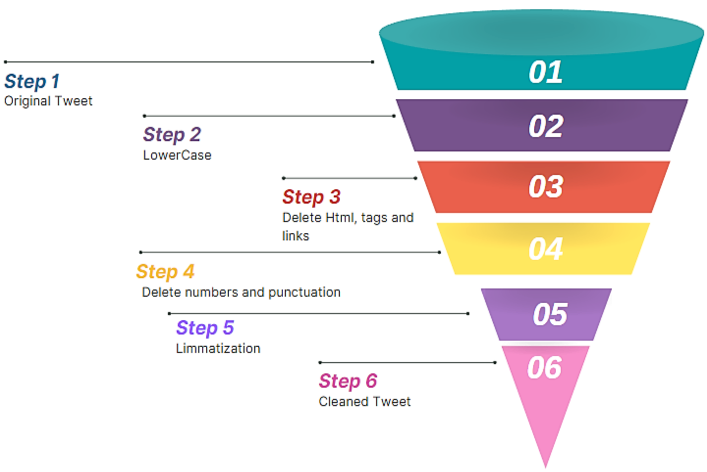

Figure 2. The data preparation techniques for our dataset

2.3. Word embedding

Word embedding, or word representation, is a technique used in natural language processing (NLP) [26]. Each word is represented in low-dimensional vectors based on numbers. When using word embedding, the semantic information of words can be captured from a large corpus. Word embeddings are used in different tasks of natural language processing to provide the best word representation. There are many types of word embedding algorithms, such as ELMo [27], GloVe [28], word2vec [29], and FastText [30]. In this study, we work with the pre-trained models of these four techniques.

2.4. GloVe

Global vector word representation (GloVe) is based on the co-occurrence and factorization of a matrix to generate their vectors. The idea is to find the relationship between words from a statistical point of view. GloVe starts by constructing a large matrix of words x context and stores the co-occurrence information as shown in Table 1. In this study, we used a pre-trained GloVe word embedding generated by Standford University that was trained on 840 billion words, with 300 dimensions.

Table 1. Unlocking the power of word representation through matrix construction for GloVe

	The	Dog	Lay	On	Carpet
The	0	1	0	1	1
Dog	1	0	1	0	0
Lay	0	1	0	1	0
On	1	0	1	0	0
Carpet	1	0	0	0	0

2.5. Word2vec

Word2vec is a word representation technique that utilizes the presence of words within written content to establish connections between words. For instance, word2vec might relate the words "females" and "males" since they often appear in similar settings. word2vec has two forms of architecture: context-prediction, which predicts the surroundings of a given word, and context-based-prediction (Bag-of-words), which predicts a word from a given surroundings. In essence, word2vec takes as input a written corpus and produces as output a word vector. In this research, we employed a pre-trained word-vectorization model that was trained on the Google News corpus, which comprises about 100 billion words and has 300-dimensional vectors.

2.6. FastText

FastText is a tool created by Facebook that is used for text classification and word representation. One of the key advantages of FastText is its ability to generate better word embeddings for rare words using n-gram character vectors. In this study, we used FastText to obtain the weights for our embedding layer based on a pre-trained model that was trained on a 2-million-word vector on common crawl and has 300-dimensional vectors.

2.7. ELMo

ELMo characterizes a sequence of words as a sequence of vectors. It employs a bi-directional LSTM model to construct its word representations. Additionally, the benefit of ELMo is that a word can have various vector representations based on the context. For instance, the word "pail" in the following two sentences: "He let go of the pail," and "I have a list of things to do before I die, a pail list. The word "pail" has different meanings in both sentences. In the ELMo method, different vectors will represent the word «pail» because it is surrounded by different words, which means different contexts. This is in contrast with other methods, which will give the same vector for both situations. In this study, we used a pre-trained model of ELMo provided by Google. The parameters we used in our research are the default signature and as_dict set to true.

2.8. Building our classifier model

In this study, for GloVe, word2vec and FastText, we built a neural network that contains the following layers: the embedding layer, flatten layer, and dense layer. The activation function used is softmax. Figure 3 describes the parameters.

For the loss, we used "categorical_crossentropy" and Adam as the optimizer method, with 'accuracy' as the metric. For ELMo, we built the following layers: an embedding layer that takes text as input, a first dense layer that takes the embedding layer as input with 'relu' as the activation function, and a second dense layer with a sigmoid as the activation function. We also used 'binary_crossentropy' as the loss function, 'rmsprop' as the optimizer, and 'accuracy' as the metric.

```
Sequential ()
Embedding (contain the weight of the methods describe above)
Flatten ()
Dense (2, activation ='softmax')
```

Figure 3. Constructing a neural network architecture

3. RESULTS AND DISCUSSION

Brexit refers to the United Kingdom's exit from the European Union (EU). In 2016, the UK voted in a referendum to leave the EU. The decision to leave the EU has sparked a great deal of political debate and controversy within the UK, as well as with other countries in the EU. Some of the key issues surrounding Brexit include immigration, trade, and sovereignty. The process of leaving the EU has been complex and has involved negotiations between the UK and the EU to determine the terms of the UK's withdrawal, as well as the future relationship between the UK and the EU.

In this study, we performed sentiment analysis and word embedding on a dataset of tweets from kaggle. For the sentiment analysis, we employed an LSTM model to categorize the sentiment of each text as positive, negative, or neutral. For the word embedding, we used a pre-trained model to map each word and phrase in the dataset to a high-dimensional vector, allowing us to analyze the relationships between different words and phrases in the context of the dataset.

3.1. Accuracy of our models

Table 2 presents a summary of the accuracy of our models. The table shows the results of four different models: Glove, word2vec, ELMo, and FastText. The accuracy of each model is measured on a scale of 0 to 1, with 1 being a perfect score. The results indicate that all models performed well, with Glove and FastText achieving an accuracy of 0.88, word2vec achieving an accuracy of 0.87, and ELMo achieving an accuracy of 0.86. Overall, the table shows that all models performed similarly, and achieved high accuracy scores, which suggests that all of the models are suitable for use in sentiment analysis tasks.

Table 2. The accuracy of our models: a summary

Model names	Accuracy
Glove	0.88
Word2vec	0.87
ELMo	0.86
FastText	0.88

3.2. Reports and metrics of our models

Tables 3 and 4 present the results of evaluating the performance of the pre-trained word embedding models, GloVe and word2vec respectively, using several different metrics. The tables show the results for precision, recall, and F1-score for each model. The precision metric measures the proportion of true positive results among all positive results, recall measures the proportion of true positive results among all actual positive observations, and F1-score is the harmonic mean of precision and recall. The tables also show the accuracy of each model, which is the proportion of correctly classified observations. The table also show macro avg and weighted avg.

The evaluation of the GloVe model reveals impressive performance metrics across multiple categories. With a precision of 0.87 for the negative class and 0.89 for the positive class, the model showcases its ability to accurately classify sentiment. Additionally, the model exhibits a recall of 0.88 for both classes, indicating its capacity to effectively capture instances of sentiment expression. Furthermore, with F1-scores of 0.88 for the negative class and 0.89 for the positive class, the GloVe model demonstrates a balanced performance in terms of precision and recall. Overall, the model achieves an accuracy of 0.88, highlighting its proficiency in sentiment analysis, as reflected in both the Macro avg and weighted avg scores, which also stand at 0.88.

Upon analyzing the performance of the word2vec model, noteworthy findings come to light. The precision of 0.87 achieved for both classes signifies the model's ability to accurately classify sentiment across the board. With a recall of 0.86 for both classes, the model demonstrates its proficiency in capturing sentiment expressions comprehensively. Furthermore, the F1-scores of 0.86 for the negative class and 0.88 for the positive class exemplify a balanced performance in terms of precision and recall. Overall, the word2vec model attains an accuracy of 0.87, as reflected in both the Macro avg and weighted avg scores, further solidifying its efficacy in sentiment analysis tasks.

Based on the tables, it is evident that both GloVe and word2vec models exhibit commendable performance, showcasing comparable results across all evaluation metrics. The closely aligned precision, recall, and f1-score values signify a well-balanced nature of these models, indicating their proficiency in accurately predicting sentiment for both positive and negative classes. These findings emphasize the reliability and effectiveness of both GloVe and word2vec in sentiment analysis tasks, underscoring their capability to provide valuable insights into the sentiment expressed within textual data.

Table 3. Measuring the performance of GloVe

	precision	recall	F1-score
0	0.87	0.88	0.88
1	0.89	0.88	0.89
accuracy			0.88
Macro avg	0.88	0.88	0.88
Wighted avg	0.88	0.88	0.88

Table 4. Measuring the performance of word2vec

	precision	recall	F1-score
0	0.87	0.86	0.86
1	0.87	0.86	0.88
accuracy			0.87
Macro avg	0.87	0.87	0.87
Wighted avg	0.87	0.87	0.87

3.3. Results

After completing our analysis, we are delighted to share the results. We put in a great deal of effort to carefully evaluate the data and arrive at these conclusions. We believe that the findings of our study will provide valuable insights and help advance our understanding of sentiment analysis and pre-trained word embeddings techniques. We hope that you find the results as interesting and informative as we do.

In the Table 5, we present the results of our research study that we explained before for comparing various pre-trained word embedding models. The results include a comparison to our previous work on the Brexit topic, as well as statistics from NatCen’s. We believe that these results provide valuable insights into the performance of different word embedding models and can help guide future research in this area.

Table 5 shows the results of our research study comparing various pre-trained word embedding models. The table compares the performance of Glove, word2vec, FastText, Elmo, LSTM and NatCen’s, in terms of the percentage of accurately classified samples of remain in EU and leave EU, regarding the Brexit topic. The results show that Glove and word2vec are the best performer with 73.56% and 75.26% respectively, followed by FastText, ELMo, LSTM, and NatCen’s with 65.48%, 61.21%, 54.88%, and 55.55% respectively.

Table 5. Analyzing the performance of word embedding models: a comparative study

	Glove	Word2vec	FastText	ELMo	LSTM	NatCen’s
Remain in EU	73.56%	75.26%	65.48%	61.21%	54.88%	55.55%
Leave EU	26.44%	24.74%	34.51%	38.79%	45.12%	44.45%

In this study, we aim to highlight the differences between using a simple word embedding layer and a pre-trained layer for sentiment analysis. The use of word embeddings in natural language processing (NLP) has shown significant improvement in various NLP tasks, including sentiment analysis. Word embeddings represent words in a low-dimensional vector space, where the distance between the vectors captures the semantic relationships between the words. To demonstrate these differences, we will provide an example of a tweet related to Brexit. Then, we will explain how a simple word embedding layer and a pre-trained layer works in a sementic perspective: "Brexit negotiations are going nowhere. It's like watching a game of chess where both sides are stuck in a stalemate."

If we use a general embedding layer, it will generate word embeddings for each word in the sentence without any prior knowledge or training on a specific task. These embeddings will be based on the distributional semantics of the words, which means that words that appear in similar contexts are likely to have similar embeddings. For example, the embedding for "Brexit" and "negotiations" may be similar since they appear in the same sentence and are related to the same topic. However, a general embedding layer may not be able to capture the full semantic meaning of the sentence or the sentiment behind it.

On the other hand, if we use a pre-trained word embedding like GloVe, it has been trained on a large corpus of text and has already captured the semantic relationships between words. Therefore, it will be better at capturing the meaning of the sentence and the sentiment behind it. For example, GloVe may be able to capture the negative sentiment in the sentence and the fact that Brexit negotiations are not progressing, which may be reflected in the embeddings for "going nowhere" and "stalemate." Overall, using a pre-trained word embedding like GloVe can be more effective than a general embedding layer in capturing the semantic relationships and sentiment in a sentence.

4. CONCLUSION

In conclusion, our study has demonstrated the effectiveness of pre-trained word embedding models for sentiment analysis. Through a series of experiments, we were able to show that these models can achieve high levels of accuracy when applied to a variety of text data. Furthermore, our analysis of the output of the models provided valuable insights into the sentiments expressed in the data. One limitation of using pre-trained word embeddings for sentiment analysis is that they are based on a fixed set of relationships between words, which may not always be relevant or appropriate for a specific task or dataset. For example, a pre-trained word embedding model trained on a general-purpose dataset may not capture domain-specific terminology or relationships that are important for a sentiment analysis task in a specific industry. Additionally, pre-trained word embeddings may be biased due to the biases present in the dataset used to train them. This can lead to incorrect or unfair sentiment classification, particularly for texts that deal with sensitive topics or marginalized groups. Finally, pre-trained word embeddings may not be able to accurately capture the sentiment of novel or rare words that were not present in the training dataset, leading to errors in classification. Looking to the future, we believe that continued research in this area will help to further improve the performance of sentiment analysis models. In particular, the development of new and more sophisticated pre-trained word embedding models will likely play a key role in this progress. Furthermore, advances in natural language processing and machine learning algorithms will help to enable sentiment analysis models to be applied in a wider range of contexts, including new domains and languages. Overall, we are optimistic about the potential of pre-trained word embedding models to advance the field of sentiment analysis. These models offer a powerful tool for extracting sentiment information from text data, and we believe that they will continue to play a crucial role in this area of research and development.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial 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. Contract number is 26UAE2020.

REFERENCES

[1] C. A. Iglesias and A. Moreno, "Sentiment Analysis for Social Media," *Appl. Sci.*, vol. 9, no. 23, p. 5037, Nov. 2019, doi: 10.3390/app9235037.

[2] E. O. Omuya, G. Okeyo, and M. Kimwele, "Sentiment analysis on social media tweets using dimensionality reduction and natural language processing," *Eng. Reports*, vol. 5, no. 3, Mar. 2023, doi: 10.1002/eng2.12579.

[3] B. Liu and L. Zhang, "A Survey of Opinion Mining and Sentiment Analysis," in *Mining Text Data*, Boston, MA: Springer US, 2012, pp. 415–463. doi: 10.1007/978-1-4614-3223-4_13.

[4] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[5] B. Liu, "Mining Opinions, Sentiments, and Emotions," in *Sentiment Analysis*, Cambridge University Press, 2015, p. 367. doi: 10.1017/CBO9781139084789.

[6] D. R. Kawade and D. K. S. Oza, "Sentiment Analysis: Machine Learning Approach," *Int. J. Eng. Technol.*, vol. 9, no. 3, pp. 2183–2186, Jun. 2017, doi: 10.21817/ijet/2017/v9i3/1709030151.

[7] Y. Yuan and W. Lam, "Sentiment Analysis of Fashion Related Posts in Social Media," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, New York, NY, USA: ACM, Feb. 2022, pp. 1310–1318. doi: 10.1145/3488560.3498423.

[8] A. Adak, B. Pradhan, and N. Shukla, "Sentiment Analysis of Customer Reviews of Food Delivery Services Using Deep Learning and Explainable Artificial Intelligence: Systematic Review," *Foods*, vol. 11, no. 10, p. 1500, May 2022, doi: 10.3390/foods11101500.

[9] M. S. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, IEEE, Jul. 2013, pp. 1–5. doi: 10.1109/ICCCNT.2013.6726818.

[10] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 4, p. e1253, Jul. 2018, doi: 10.1002/widm.1253.

[11] M. Ahmad, S. Aftab2, S. S. Muhammad, and S. Ahmad, "Machine Learning Techniques for Sentiment Analysis: A Review," *Int. J. Multidiscip. Sci. Eng.*, vol. 8, no. 3, 2017.

[12] Dr. G. S. N. Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, and M. B. Mounika Bagadi, "Text based Sentiment Analysis using LSTM," *Int. J. Eng. Res.*, vol. V9, no. 05, May 2020, doi: 10.17577/IJERTV9IS050290.

[13] A. Matsui and E. Ferrara, "Word Embedding for Social Sciences: An Interdisciplinary Survey," *Comput. Sci. Artif. Intell.*, vol. 1, 2022, doi: 10.48550/arXiv.2207.03086.

[14] B. Oscar Deho, A. William Agangiba, L. Felix Aryeh, and A. Jeffery Ansah, "Sentiment Analysis with Word Embedding," in *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, IEEE, Aug. 2018, pp. 1–4. doi: 10.1109/ICASTECH.2018.8506717.

[15] Y. Qi, D. Sachan, M. Felix, S. Padmanabhan, and G. Neubig, "When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation?," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 529–535. doi: 10.18653/v1/N18-2084.

[16] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews," *Procedia Comput. Sci.*, vol. 179, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.

[17] L. Xiaoyan, R. C. Raga, and S. Xuemei, "GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews," *J. Sensors*, vol. 2022, pp. 1–12, Oct. 2022, doi: 10.1155/2022/7212366.

[18] T. Yazdizadeh, "Comparative Evaluation on Efect of ELMo in Combination with Machine Learning, and Ensemble Models in Cyberbullying Detection," Carleton University, 2022.

[19] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia Comput. Sci.*, vol. 189, pp. 343–350, 2021, doi: 10.1016/j.procs.2021.05.103.

[20] M. Ihab, L. Soumaya, B. Mohamed, H. Haytam, and F. Abdelhadi, "Ontology-based sentiment analysis and community detection on social media: application to Brexit," in *Proceedings of the 4th International Conference on Smart City Applications*, New York, NY, USA: ACM, Oct. 2019, pp. 1–7. doi: 10.1145/3368756.3369090.

[21] S. Selva Birunda and R. Kanniga Devi, "A Review on Word Embedding Techniques for Text Classification," 2021, pp. 267–281. doi: 10.1007/978-981-15-9651-3_23.

[22] J. S. Santos, A. Paes, and F. Bernardini, "Combining Labeled Datasets for Sentiment Analysis from Different Domains Based on Dataset Similarity to Predict Electors Sentiment," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, IEEE, Oct. 2019, pp. 455–460. doi: 10.1109/BRACIS.2019.00086.

[23] M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Appl. Sci.*, vol. 12, no. 17, p. 8765, Aug. 2022, doi: 10.3390/app12178765.

[24] J. Chen, Y. Chen, Y. He, Y. Xu, S. Zhao, and Y. Zhang, "A classified feature representation three-way decision model for sentiment analysis," *Appl. Intell.*, vol. 52, no. 7, pp. 7995–8007, May 2022, doi: 10.1007/s10489-021-02809-1.

[25] U. D. Gandhi, P. Malarvizhi Kumar, G. Chandra Babu, and G. Karthick, "Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)," *Wirel. Pers. Commun.*, May 2021, doi: 10.1007/s11277-021-08580-3.

[26] R. Sann and P.-C. Lai, "Understanding homophily of service failure within the hotel guest cycle: Applying NLP-aspect-based sentiment analysis to the hospitality industry," *Int. J. Hosp. Manag.*, vol. 91, p. 102678, Oct. 2020, doi: 10.1016/j.ijhm.2020.102678.

[27] M. Yang, J. Xu, K. Luo, and Y. Zhang, "Sentiment analysis of Chinese text based on Elmo-RNN model," *J. Phys. Conf. Ser.*, vol. 1748, no. 2, p. 022033, Jan. 2021, doi: 10.1088/1742-6596/1748/2/022033.

[28] A. Zouzou and I. El Azami, "Text sentiment analysis with CNN & GRU model using GloVe," in *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*, IEEE, Oct. 2021, pp. 1–5. doi: 10.1109/ICDS53782.2021.9626715.

[29] L. Mostafa, "Egyptian Student Sentiment Analysis Using Word2vec During the Coronavirus (Covid-19) Pandemic," 2021, pp. 195–203. doi: 10.1007/978-3-030-58669-0_18.

[30] I. Santos, N. Nedjah, and L. de M. Mourelle, "Sentiment analysis using convolutional neural network with fastText embeddings," in *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, IEEE, Nov. 2017, pp. 1–5. doi: 10.1109/LA-CCI.2017.8285683.

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.