Indonesian Journal of Electrical Engineering and Computer Science
Vol. 35, No. 3, September 2024, pp. 1933~1941
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v35.i3.pp1933-1941

Enhancing emotion detection with synergistic combination of word embeddings and convolutional neural networks

Anil Kumar Jadon, Suresh Kumar
Department of Computer Science and Engineering, Manav Rachna International Institute of Research and Studies (MRIIRS), Faridabad, India

Article Info

Article history:

Received Feb 27, 2024
Revised May 7, 2024
Accepted May 12, 2024

Keywords:

BERT
Convolution neural network
Deep learning
ELMo
Emotion detection
GloVe
RoBERTa
Transfer learning
Word2Vec

ABSTRACT

Recognizing emotions in textual data is crucial in a wide range of natural language processing (NLP) applications, from consumer sentiment research to mental health evaluation. The word embedding techniques play a pivotal role in text processing. In this paper, the performance of several well-known word embedding methods is evaluated in the context of emotion recognition. The classification of emotions is further enhanced using a convolutional neural network (CNN) model because of its propensity to capture local patterns and its recent triumphs in text-related tasks. The integration of CNN with word embedding techniques introduced an additional layer to the landscape of emotion detection from text. The synergy between word embedding techniques and CNN harnesses the strengths of both approaches. CNNs extract local patterns and features from sequential data, making them well-suited for capturing relevant information within the embeddings. The results obtained with various embeddings highlight the significance of choosing synergistic combinations for optimum performance. The combination of CNNs and word embeddings proved a versatile and effective approach.

Corresponding Author:

Anil Kumar Jadon
Department of Computer Science and Engineering
Manav Rachna International Institute of Research and Studies (MRIIRS)
Faridabad, India
Email: aniljadon.jadon@gmail.com

1. INTRODUCTION

There is a massive amount of textual data in the modern digital world. The digital world is overflowing with human emotions conveyed in words, from tweets to reviews and comments to blogs. The ability to algorithmically identify and categorize these emotions [1], [2] can have significant ramifications for many industries, from firms looking to gauge customer sentiment to healthcare practitioners trying to gain insights into mental health through textual exchanges. Natural language processing (NLP), which links unprocessed text and computational analysis, is essential for sifting through this massive sea of text and drawing significant insights. Identifying and categorizing the underlying emotions inherent in textual data is the subject of the NLP sub-domain of emotion detection [3]. Such tasks formerly extensively leaned on lexicon-based methods [4], where emotions were determined based on prepared lists of words associated with emotions. However, with the development of deep learning, the paradigm has moved in favor of distributed word representations, also known as word embeddings. These embeddings capture language's semantic nuance and nuances, which offer dense vector representations of words. Unlike lexicon-based methods, which are often rigid and lack nuance, word embeddings offer a more fluid and contextual understanding of text. In recent years, the popularity and growth of Word embedding approaches have increased tremendously [5]. The foundation was created by established techniques like Word2Vec [6] and global vectors (GloVe) [7], but soon more complex models like embeddings from language models (ELMo) [8], bidirectional encoder representations from transformers (BERT) [9], and robustly optimized BERT pre-training approach (RoBERTa) [10] gained popularity and promised context-aware embeddings. Each method has unique advantages, disadvantages, and underlying principles that require a thorough analysis to determine their effectiveness.

The convolutional neural networks (CNNs) [11] have also made tremendous strides in the NLP field. CNNs are surprisingly good at handling text, especially when identifying local patterns inside sentences. CNNs were initially developed for image processing. Therefore, combining sophisticated word embeddings with CNN models appears to be a viable way to progress in emotion recognition. With this background, the current research intends to compare synergic integration of CNN with several word embedding approaches when applied to emotion detection using the international survey on emotion antecedents and reactions (ISEAR) dataset.

2. RELATED WORK

The field of emotion identification from text has seen a lot of research and development over the past few decades. With a focus on word embeddings [12] and the application of CNNs in emotion detection, this section summarizes the most significant contributions and findings. Initially, emotion detection relied mostly on lexicon-based methods, which mapped words to predefined emotion categories. However, with the evolution of neural network, more sophisticated techniques emerged, offering better contextual understanding in the field of emotion detection.

2.1. Evolution of word embeddings

Traditional methods for representing text, including term frequency-inverse document frequency (TF-IDF) [13] and bag of words (BoW), cannot capture the semantic connections between words, whereas. The breakthrough of Word2Vec [14] shifted the paradigm, which offered semantically rich embeddings using shallow neural architectures. Its versions, skip-gram and continuous bag-of-words (CBOW), were primarily focused on utilizing local context to produce useful word vectors. Akuma *et al.* [13] describes the difficulty in controlling hate speech on social media sites like Twitter and its adverse effects on users. It examines efforts to improve detection techniques by combining machine learning models with algorithms like TF-IDF and BoW. The objective is to identify the optimal model for instantly identifying this content using an API-integrated online system. The decision tree model utilizing TF-IDF performed better than others, reaching 92.43% accuracy. With the release of GloVe [15] the field of word embedding achieved a new milestone. This method combined local context with global co-occurrence statistics to produce embeddings representing a broader semantic landscape. By focusing on sub-word units, FastText added to the richness of this domain by accommodating the nuances of several languages with complex morphologies. With ELMo [16] which focused on dynamic embedding generation, the transformative character of deep contextual embeddings came to the fore. The success of Transformer-based models like BERT and RoBERTa further demonstrated the importance of context in embeddings.

2.2. Emotion detection: from lexicons to neural networks

Historically, lexicon-based techniques have been the mainstay of emotion recognition. EmoLex [17] and similar works offered insights into emotion tagging based on predefined word lists, which could not capture complex emotional aspects in texts. This constraint led to a change in the field of emotion detection toward machine learning and deep learning. The ISEAR dataset has continued to play a vital role in these efforts, with numerous studies using it to differentiate between distinct emotion categories.

2.3. CNNs in textual tasks

Initially known for image processing, CNNs [18] showcased their robust performance in NLP. They were exceptionally skilled at text interpretation because of their innate capacity to recognize local patterns. Kim's work in 2014 [19] was essential in demonstrating the effectiveness of CNNs for classification at the sentence level. Later, academicians began to go deeper and started building models using CNNs. The demonstrations by Zhang and Wallace [20] on how these architectures could successfully capture n-gram subtleties provided more evidence of CNNs' adaptability in text processing, particularly when used with multi-sized kernels. CNNs' usefulness in classification tasks such as emotion detection and other classification tasks has grown, especially when combined with enhanced word embeddings [21]. The CNN architectures with Transformer-based embeddings [22] have been explored in recent research to improve text categorization performance.

3. METHOD

The international survey on emotion antecedents and reactions (ISEAR) dataset is used and fundamental text preprocessing techniques are applied for conducting all experiments related to word embeddings. This dataset was also divided into a train and validation Using the 80/20 random sampling method. Various word embedding techniques, such as Word2Vec, GloVe, FastText, ELMO, BERT, and RoBERTa are then evaluated on this dataset to analyze their effectiveness in emotion detection tasks.

3.1. Word embedding techniquesc

The study of NLP has been completely changed by word embeddings, which are essentially dense vector representations of words. The word embeddings encode semantic meanings based on word co-occurrence and distribution patterns in corpora. For our study, we thoroughly investigated several well-established embedding techniques:

3.1.1. TF-IDF

The TF-IDF technique measures a term's importance within a document with-respect-to its frequency across a corpus [13]. The term "term Frequency" (TF) refers to how often a term appears in a document. The inverse document frequency (IDF), is calculated as the logarithm of the total number of documents over documents containing the phrase, assesses the term's informational richness. The TF-IDF weight is produced by multiplying TF by IDF, providing a representation that emphasizes word uniqueness across documents.

3.1.2. Word2Vec

Word2Vec, developed by Mikolov *et al.* [6], [23] in 2013, is a well-known word embedding method that completely changed the field of NLP. Word2Vec represents words as high-dimensional vectors to capture the semantic relationships between them. This context-based semantic data is stored in dense vectors, which typically have a dozen to a hundred dimensions. Word2Vec uses shallow neural networks, intended particularly for processing textual data. Under Word2Vec, there are two main architectures:

Skip-Gram: this model uses a word ω_t (the target word) to forecast the words $\omega_{t-j,} \ldots, \omega_{t-1,}\omega_{t+1}, \ldots\ldots, \omega_{t+1}$ (the context words) that will be found immediately around it. Using the term "ocean" as an example, the Skip-Gram model may attempt to predict words like "blue," "vast," and "waves" that are next to it. Mathematically, it optimizes the following objective presented in (1).

$$max \sum_{t=1}^{T} \sum_{-j \le k \le j, k \ne 0} Log\ P(w_{t+k}|w_t) \quad (1)$$

Where $P(w_{t+k}|w_t)$ is the probability of observing a context word given the target word, which is parameterized by the embeddings. Skip-Gram excels at handling uncommon words, which is one of its main strengths.

CBOW: unlike skip-gram, the CBOW model works in the opposite manner. It attempts to guess the target word using a variety of context terms as input. In the same example, CBOW would try to predict the word "ocean" given the terms "blue," "vast," and "waves." generally works faster and is more accurate for frequent words. The model optimizes the following objective in (2):

$$max \sum_{t=1}^{T} Log\ P\left(w_t | w_{t-j,} \ldots, w_{t-1,} w_{t+1,} \ldots w_{t+j}\right) \quad (2)$$

Where $P\left(w_t | w_{t-j,} \ldots, w_{t-1,} w_{t+1,} \ldots w_{t+j}\right)$ is the probability of observing the target word given the context words, which is also parameterized by the embeddings? Word2Vec not only groups similar words together but can also capture more complex relationships.

3.1.3. Global vectors for word representation

Pennington *et al.* [7] presented GloVe in 2014. It is a word embedding technique that considers the overall statistics of a given corpus to capture semantic and syntactic word associations. GloVe focuses on the entire statistical data of a text dataset as opposed to Word2Vec, which strongly relies on local context. GloVe is based on the notion that word co-occurrence probability ratios can convey significant semantic information. The method's foundation is the factorization of the word co-occurrence matrix. Under the GloVe, there are two main architectures is co-occurrence matrix construction and matrix factorization.

Co-occurrence matrix construction: start constructing a word co-occurrence matrix, X, for a given corpus. Each entry X_{ij} in this matrix represents how often word i appears in the context of word j. Matrix factorization: GloVe aims to factorize this matrix such that the dot product of two-word vectors nearly matches the logarithm of the co-occurrence count of those two words. This is presented in (3). For every word i, it has two vectors: ω_i (the word vector) and $\widetilde{\omega}_j$ (the context word vector).

$$w_i^T \widetilde{w}_j \approx \log(Xij) \tag{3}$$

The GloVe objective function is presented in (4) as:

$$J = \sum_{i,j=1}^{V} f(Xij)\left(w_i^T \widetilde{w}_j + b_i + \tilde{b}_j - \log(Xij)\right)^2 \tag{4}$$

where:
V is the vocabulary size. b_i and $\tilde{b}_j$ are bias terms for words i and j, respectively.
f is a weighting function that ensures words with no co-occurrences are assigned appropriate weights.
The factorization procedure aims to ensure that the generated vectors accurately reflect the semantic connections between words.

3.1.4. FastText

Joulin *et al.* [24] at Facebook's artificial intelligent research (FAIR) lab, introduced the FastText word representation and categorization model in 2016. FastText can produce improved representations for out-of-the-vocabulary (OOV) words and morphologically rich languages by functioning at the sub-word level. Each word in FastText is represented as a collection of character n-grams, and the vector for the word is the sum of these n-gram vectors. For instance, for the word "apple" and using tri-grams as an example, the n-grams would be as presented in (5).

$$G_{apple} = \{ap_, app, ppl, ple, le\} \tag{5}$$

In FastText, every n-gram ‘g’ is associated with a vector $'Zg'$ in the embedding space. The vector representation of a word ‘ω’ is then the sum of the vectors of its n-grams as presented in (6).

$$v_w = \sum_{g \in G_w} Zg \tag{6}$$

This lets the model make a word vector for a word that was not seen during training. If the word is not in the dictionary, but its character n-grams were seen in other words during training, FastText can figure out its vector by adding the n-gram vectors.

3.1.5. ELMo

ELMo [8], created by researchers at the Allen Institute for AI, transformed the field of NLP by offering dynamic embeddings that capture polysemy by taking context into account. ELMo considers the complete sentence to provide word vectors, unlike conventional embeddings like Word2Vec and GloVe, which only provide a single, static vector for each word. ELMo employs a bidirectional long short-term memory (LSTM) structure that has been trained on a sizable corpus as a language model. Both the left and right contexts of each word are captured. These bidirectional LSTMs are used in several layers throughout the system. As a result, each word has multiple sets of embeddings, one from each LSTM layer. ELMo combines the embeddings from all LSTM layers to generate the embeddings for a particular task. The word is represented as a weighted sum of LSTM outputs that ensures that the word's representation is influenced by its surrounding context. Given an input sentence $s = w_1, w_{2,} \dots, w_T$ with T tokens, each word w_t is passed through an embedding layer followed by L layers of bidirectional LSTM (BiLSTMs). Let $h_{t,l}^{Forward}$ and $h_{t,l}^{backward}$ denotes the hidden states of the forward and backward LSTMs at layer l and time t, respectively. This multi-layer BiLSTM network of ELMo, was developed with the intention of language modeling, and it includes:

- Depth: ELMo has a deep architecture that typically consists of many layers of BiLSTMs. Each layer of the network captures information in a slightly different way, with lower layers being better at catching syntactic details and higher layers more skilled at comprehending semantic contexts. The hidden state at each layer l for a token w_t can be composed as presented in (7).

$$h_{t,l} = B_i LSTM\left(h_{t,l-1}^{Forward}, h_{t,l-1}^{backward}, l\right) \tag{7}$$

$for\ l = 1, 2, \dots\dots, L$

– Contextualization: the ELMo model processes an input sequence of tokens through these BiLSTM layers, resulting in a set of high-dimensional vectors for each token at each layer. The contextual embeddings in these vectors represent the word in the context of the provided sequence. These BiLSTM layers create a set of $2L + 1$ (including the word embeddings) sets of embeddings h_t^l for each token at each layer l presented in (8).

$$h_t^l = \left[h_{t,l}^{Forward}\,;\, h_{t,l}^{backward}\right] \tag{8}$$

– Representation: to create representations for tasks requiring word embeddings, ELMo weighs and concatenates the embeddings from several layers. Because these weights are task-specific, the embeddings automatically highlight the syntactic or semantic components that are most advantageous for a particular NLP job. To get the final ELMo embedding $ELMo_t$ for a token w_t , the embeddings from all the layers are combined in a task-specific weighted sum as presented in (9).

$$\mathrm{ELMo}_t = \gamma \sum_{l=0}^{L} s_l h_t^l \tag{9}$$

Here, s_l are the SoftMax-normalized weights and γ is a task-specific scaling factor. The weights s_l , is presented in (10), are learned during the task-specific fine-tuning process, allowing the model to emphasize the embeddings that are most useful for each task.

$$s_l = \frac{exp(a_l)}{\sum_{j=0}^{L} exp(a_j)} \tag{10}$$

Where a_l are learnable parameters.
By this mechanism, ELMo captures complex characteristics like polysemy and simpler features like syntax and semantics, depending on what is most beneficial for the task.

3.1.6. BERT

Researchers at Google AI developed BERT [25], released in 2018 and is another significant step forward in the field of NLP. It combines the context-driven features of ELMo with the transformer design. It accomplishes a deep bidirectional understanding of text data. BERT architecture [26] includes:

– The transformer: the basis of BERT is the transformer architecture, which was first explained in "Attention is all you need" by Vaswani *et al.* in 2017 [27]. The transformer model primarily uses attention processes to figure out how words or parts of words in a text are related to each other in context.
– Encoder stacks: BERT utilizes only the encoder stacks from the transformer architecture. There are several layers of these encoders layered on top of one another, depending on the version (BERT-base or BERT-large). Each encoder layer E_l in the stack can be presented as in (11). Where $Q_, K\ and\ V$ are the query, key, and value vectors for the self-attention mechanism and E_{l-1} is the output from the previous layer.

$$E_l = MultiHead(Q, K, V) + FeedForward(E_{l-1}) \tag{11}$$

– Multi-head self-attention mechanism: each encoder in the stack contains a multi-head self-attention mechanism. It enables the model to concentrate on various words in the input text. The multi-head self-attention can be represented in (12).

$$MultiHead(Q, K, V) = Concat(H_1, H_2, \dots H_k)W^0 \tag{12}$$

Where $H_i = Attention\left(QW_i^Q, QW_i^K, QW_i^V\right)$, k is the number of heads and W^O, W_i^Q, W_i^K, W_i^V are the learnable weights.
– WordPiece tokenization: using the WordPiece tokenization approach, the text is divided into subwords before processing. This enables a more flexible and extensive vocabulary by dividing words into smaller components.
– Positional embeddings: positional embeddings are added to the token embeddings to provide information about the position of a word in a sequence. The positional embedding PE is summed with the token embedding T to produce the final input X presented in (13).

$$X = T + PE \tag{13}$$

Pre-training tasks: BERT is trained on two tasks simultaneously:

- Masked language model (MLM): the model tries to predict a portion of the input words x_i that are randomly masked [MASK] and are only known by their context. As a result, bidirectional training is encouraged as opposed to models like GPT, which can only predict words in a forward order. The objective function is to minimize the negative log-likelihood of predicting the original token. x_i, given the masked sequence presented in (14).

$$\mathcal{L}_{MLM} = -\log P(x_i | x_{masked}) \tag{14}$$

- Next sentence prediction: the model is fed two sentences, and it predicts whether the second statement in the pair comes after another sentence in the source text. Given two sentences S_A and S_B, BERT target to predict if S_B is the subsequent sentence to S_A in the original input text. This is a binary classification problem with objective function binary cross-entropy loss presented in (15).

$$\mathcal{L}_{NSP} = -y\, log(\hat{y}) - (1-y)\log(1-\hat{y}) \tag{15}$$

The final pre-training objective is a sum of these two losses presented in (16).

$$\mathcal{L}_{total} = \mathcal{L}_{MLM} + \mathcal{L}_{NSP(x)} \tag{16}$$

The BERT architecture is illustrated in Figure 1.

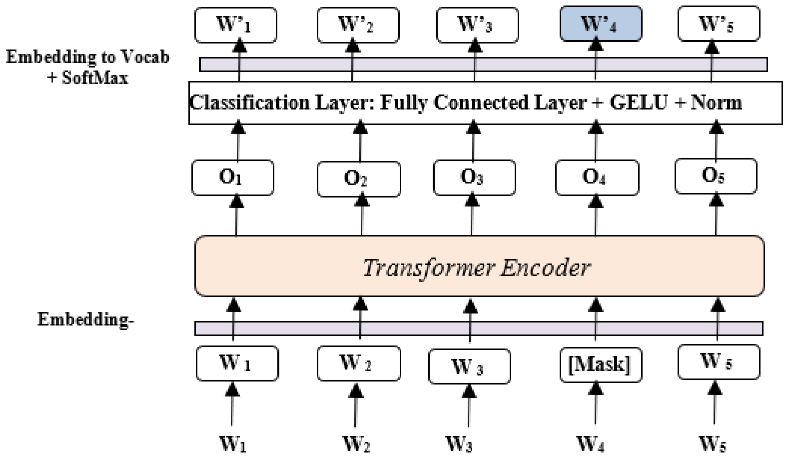

Figure 1. BERT architecture [28]

3.1.7. RoBERTa

RoBERTa [10] is an improved version of BERT that Facebook AI introduced. RoBERTa uses dynamic masking so that the model sees each token masked at different positions during pre-training. RoBERTa uses byte-pair encoding, which enables it to have a more adaptable and granular tokenization scheme than BERT, which uses the WordPiece tokenization. The transformer architecture used by RoBERTa is the same as that used by BERT. The model uses self-attention mechanisms to produce context-rich embeddings by weighing the importance of words in a phrase with-respect-to a specific word. The key functional components of RoBERTa are as follows:

- Layers: RoBERTa, like BERT, comprises several transformer layers. Multiple heads of self-attention and feed-forward neural networks are included in each layer. With 12 layers for the base version and 24 layers for the large version, RoBERTa's base architecture matches that of BERT.
- Attention heads: several attention heads operate continuously within each transformer layer, enabling the model to capture diverse relationships and characteristics of the data. Each layer in the base model has 12 attention heads, whereas 16 in the larger version.
- Hidden units: the base model's hidden size or depth is 768, while the large model's hidden size or depth is 1,024.

– Byte-pair encoding (BPE): BPE is a technique for subword tokenization that was initially developed for data compression purposes and subsequently used in various natural language processing applications, particularly in the domains of neural machine translation and pre-trained language models.

3.2. CNN architecture for classification

Although CNNs have traditionally been used to analyze image data, their capacity to spot specific patterns makes them a superb choice for text classification tasks like emotion detection. They have proven to be incredibly effective in text classification tasks. They can employ the convolution process to collect local features and comprehend hierarchical representations of input data. CNN architecture [18] includes the convolution, pooling, fully connected, and output layers. The proposed method for the emotion classification model is presented in Figure 2.

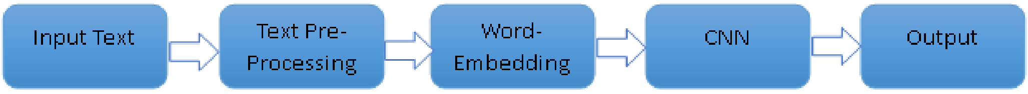

Figure 2. Proposed model architecture

4. EXPERIMENTS AND RESULTS

We used the cloud-based Google Colab platform for the empirical analysis of our research on the ISEAR dataset. This platform provided a robust computational environment with 32 GB of RAM and the high-performance NVIDIA Tesla T4 GPU. Table 1 provides a comprehensive analysis of the various word embedding methods integrated with CNNs to identify emotions. It compares the performance of each strategy on the ISEAR dataset using four key metrics: accuracy, precision, recall, and F1-score.

Table 1. Comparison of various embedding approaches

CNN approach with	F1-score	Precision	Recall	Accuracy
TF-IDF	.57	.57	.57	.57
Word2Vec	.65	.65	.65	.65
GloVe	.61	.61	.61	.61
FastText	.64	.65	.65	.65
ELMo	.57	.57	.57	.57
BERT	.69	.70	.70	.70
RoBERTa	**.73**	**.74**	**.73**	**.73**

Table 2 showcases the comparative analysis of the performance of the proposed synergistic integration of word embeddings with CNN with the techniques available in literature. We assessed these studies based on F1 score, precision, recall, and accuracy. It is evident from the results that the proposed technique outperforms most of the techniques available in literature. Through this comparison, we identified potential areas of enhancement and gained insights into the evolving landscape of emotion detection.

Table 2. Comparison of previous researches

Reference	Approach	F1-score	Precision	Recall	Accuracy
Atmaja *et al.* [3]	Recurrent neural network (RNN)-gated recurrent unit (GRU)	54%	56%	54%	56%
Setiawan *et al.* [9]	BERT-deep neural network (DNN)	-	-	-	66%
Razek and Frasson [29]	Machine learning (ML)	27%	23%	45%	-
Polignano *et al.* [12]	GoogleEmb	62%	63%	63%	-
Polignano *et al.* [12]	GloVeEmb	62%	62%	62%	-
Polignano *et al.* [12]	FastTextEmb	64%	64%	63%	-
Asghar *et.al.* [30]	ML	63%	64%	64%	64%
Alhuzali and Ananiadou [2]	BERT	69%	71%	69%	-
Acheampong *et al.* [1]	RoBERTa	-	-	-	74%
BERT (ours)	**BERT+CNN**	69%	70%	70%	70%
RoBERTa (ours)	**RoBERT+CNN**	**73%**	**74%**	**73%**	**73%**

5. CONCLUSION

The integration of CNN with word embedding techniques introduced an additional layer to the landscape of emotion detection from text. When fused with CNN, the contextualized embeddings from ELMo, bidirectional contextual understanding from BERT and RoBERTa, traditional TF-IDF weighting, and global semantic relationship captured by GloVe yield promising results. The synergy between word embedding techniques and CNN harnesses the strengths of both approaches. CNNs extract local patterns and features from sequential data, making them well-suited for capturing relevant information within the embeddings. The outcomes also support the capability of CNNs to identify textual emotions. The distinct results obtained with various embeddings highlight the significance of choosing synergistic combinations for optimum performance. Furthermore, the interaction of accuracy, precision, recall, and the F1-score provides a comprehensive lens to evaluate model capabilities, highlighting the need for a balanced strategy. Combined CNNs and word embeddings proved a versatile and effective approach. The optimal choice depends on the dataset's intricacies and the application's requirements.

REFERENCES

[1] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Recognizing emotions from texts using an ensemble of transformer-based language models," in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Dec. 2021, pp. 161–164, doi: 10.1109/ICCWAMTIP53232.2021.9674102.

[2] H. Alhuzali and S. Ananiadou, "SpanEmo: casting multi-label emotion classification as span-prediction," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 1573–1584, doi: 10.18653/v1/2021.eacl-main.135.

[3] B. T. Atmaja, K. Shirai, and M. Akagi, "Deep learning-based categorical and dimensional emotion recognition for written and spoken text," in *IPTEK Journal of Proceedings Series*, 2019, pp. 1–10.

[4] L. Shang, H. Xi, J. Hua, H. Tang, and J. Zhou, "A lexicon enhanced collaborative network for targeted financial sentiment analysis," *Information Processing & Management*, vol. 60, no. 2, p. 103187, Mar. 2023, doi: 10.1016/j.ipm.2022.103187.

[5] L. P. Hung and S. Alias, "Beyond sentiment analysis: a review of recent trends in text based sentiment analysis and emotion detection," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 27, no. 1, pp. 84–95, Jan. 2023, doi: 10.20965/jaciii.2023.p0084.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "[Word2Vec] distributed representations of words and phrases and their compositionality (Google 2013)," *CrossRef Listing of Deleted DOIs*, vol. 1, pp. 1–9, 2013.

[7] J. Pennington, R. Socher, and C. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[8] Z. Huang and W. Zhao, "Combination of ELMo representation and CNN approaches to enhance service discovery," *IEEE Access*, vol. 8, pp. 130782–130796, 2020, doi: 10.1109/ACCESS.2020.3009393.

[9] E. I. Setiawan, L. Kristianto, A. T. Hermawan, J. Santoso, K. Fujisawa, and M. H. Purnomo, "Social Media emotion analysis in indonesian using fine-tuning BERT model," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Apr. 2021, pp. 334–337, doi: 10.1109/EIConCIT50028.2021.9431885.

[10] K. L. Tan, C. P. Lee, and K. M. Lim, "RoBERTa-GRU: a hybrid deep learning model for enhanced sentiment analysis," *Applied Sciences*, vol. 13, no. 6, p. 3915, Mar. 2023, doi: 10.3390/app13063915.

[11] F. Sargisson, X. Gao, and B. Xue, "Learning CNN architecture for multi-view text classification using genetic algorithms," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2022, pp. 1507–1514, doi: 10.1109/SSCI51031.2022.10022150.

[12] M. Polignano, P. Basile, M. de Gemmis, and G. Semeraro, "A comparison of word-embeddings in emotion detection from text using BiLSTM, CNN and self-attention," in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, Jun. 2019, pp. 63–68, doi: 10.1145/3314183.3324983.

[13] S. Akuma, T. Lubem, and I. T. Adom, "Comparing bag of words and TF-IDF with different models for hate speech detection from live tweets," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3629–3635, Dec. 2022, doi: 10.1007/s41870-022-01096-4.

[14] A. Cattle and X. Ma, "Predicting word association strengths," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1283–1288, doi: 10.18653/v1/D17-1132.

[15] Y. Zhang, "Named entity recognition for social media text," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2019, p. Master's thesis, Uppsala University.

[16] S. Bell, H. Yannakoudakis, and M. Rei, "Context is key: grammatical error detection with contextual word representations," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 103–115, doi: 10.18653/v1/W19-4410.

[17] S. Mohammad and P. D. Turney, "NRC emotion lexicon," *National Research Council, Canada*, vol. 2, pp. 1–234, 2013.

[18] S. Karkra, P. Singh, and K. Kaur, "Convolution neural network: a shallow dive in to deep neural net technology," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2 Special Issue 7, pp. 487–495, Sep. 2019, doi: 10.35940/ijrte.B1092.0782S719.

[19] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.

[20] Y. Zhang and B Wallace, "IJCNLP(1) - a sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *Na*, vol. 1, no. NA, pp. 253–263, 2017.

[21] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *COLING 2014 - 25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers*, 2014, pp. 69–78.

[22] S. F. Sabbeh and H. A. Fasihuddin, "A comparative analysis of word embedding and deep learning for arabic sentiment classification," *Electronics*, vol. 12, no. 6, p. 1425, Mar. 2023, doi: 10.3390/electronics12061425.

[23] A. K. Jadon and S. Kumar, "Emotion detection using Word2Vec and convolution neural networks," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 33, no. 3, pp. 1812–1819, Mar. 2024, doi: 10.11591/ijeecs.v33.i3.pp1812-1819.

[24] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 427–431, doi: 10.18653/v1/E17-2068.
[25] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: pre-training of deep bidirectional transformers for language understanding," *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologie (NAACL-HLT 2019)*, no. Mlm, pp. 4171–4186, 2018.
[26] A. R. Abas, I. Elhenawy, M. Zidan, and M. Othman, "BERT-CNN: a deep learning model for detecting emotions from text," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 2943–2961, 2022, doi: 10.32604/cmc.2022.021671.
[27] Vasvani A. *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
[28] F. K. Khattak, S. Jeblee, C. Pou-Prom, M. Abdalla, C. Meaney, and F. Rudzicz, "A survey of word embeddings for clinical text," *Journal of Biomedical Informatics*, vol. 100, p. 100057, 2019, doi: 10.1016/j.yjbinx.2019.100057.
[29] M. A. Razek and C. Frasson, "Text-based intelligent learning emotion system," *Journal of Intelligent Learning Systems and Applications*, vol. 09, no. 01, pp. 17–20, 2017, doi: 10.4236/jilsa.2017.91002.
[30] M. Z. Asghar *et al.*, "Performance evaluation of supervised machine learning techniques for efficient detection of emotions from online content," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1093–1118, 2020, doi: 10.32604/cmc.2020.07709.

BIOGRAPHIES OF AUTHORS

Anil Kumar Jadon is a Ph.D. research scholar and professional with significant expertise in data science and artificial intelligence. He is an active contributor to the field of AI research, with a focus on machine learning, artificial intelligence, and natural language processing. He can be contacted at email: aniljadon.jadon@gmail.com.

Prof. Dr. Suresh Kumar is a distinguished academician and professional associated with the Manav Rachna International Institute of Research and Studies (MRIIRS). He is a life member of the Indian Society of Technical Education (ISTE) and Computer Society India (CSI). He is also a Senior member of IEEE, USA, and IACSIT, Singapore. He has published more than eighty research papers in international journals and conferences. The HelixSmartLabs a startup incubated at Manav Rachna under his supervision and is registered as a Pvt. Ltd company. He has published two patents and a Technology Transfer in arules package of R Data Analytics Software (included in version 1.5.5 onwards). The package is available at https://cran.r-project.org/package=arules. He can be contacted at email: enthuvs@gmail.com.