Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

BERT-CNN Model to Draw Emotions From Given Text

Dr. Shashikala Marthala[1], C.Sireesha[2]

[1]*Professor, Dept of CSE&AIML, SR University, India, marthakala08@gmail.com*
[2]*Research Scholar, Dept of CSE&AIML, SR University, India, 2103C5006@sru.edu.in*

Text mining of short-text data, such as social media posts, micro-blogs, news articles, and customer reviews, has become a valuable resource for discovering various aspects, including emotions. However, traditional ML models for emotion detection like logistic regression, naive Bayes, and support vector machines (SVM) have also been used for sentiment analysis, but they typically require handcrafted features in text that fall short of capturing the emotional relationship between words. On the other hand, BERT and CNN models use pre-trained word embeddings and learn hierarchical representations of text data through multiple layers of neural networks, which allows them to capture the nuances of language and perform well on various natural language processing (NLP) tasks including sentiment analysis. Overall, the BERT + CNN model has been shown to outperform traditional ML models in many benchmark sentiment analysis datasets. However, the performance of the model also depends on the size and quality of the training data, hyperparameter tuning, and other factors specific to the task at hand. The empirical findings demonstrated a significant enhancement in the F1 score of the BERT-CNN model, surpassing both the BERT and CNN models by approximately 14.4% and 17.4% corresponding.

Keywords: BERT, CNN, NLP, Pose Landmarks, Computer Vision, Artificial Intelligence.

1. Introduction

Emotion detection in text data is a critical task for understanding human behavior and improving customer experience in various industries, including marketing and customer service. However, traditional models for emotion detection in text data have limitations that can hinder progress in these industries. One such limitation is the difficulty in capturing the emotional relationship between words in a sentence or a document.

Traditional models typically rely on a bag-of-words approach that treats words in isolation, without considering the context in which they appear. This can result in inaccurate detection of emotions, as emotions are often expressed through the relationship between words rather than individual words themselves.

The BERT-CNN model addresses this limitation by combining the strengths of two powerful models: BERT and CNN. BERT is a pre-trained language model that can understand the context of words in a sentence, while CNNs are able to extract features from text data by sliding a filter over the input sequence. By combining these two models, the BERT-CNN algorithm is able to capture the emotional relationship between words in a sentence or a document, leading to more accurate detection of emotions.

This is particularly important for industries such as marketing and customer service, where understanding customer emotions is crucial for success. Without the BERT-CNN model, accurately detecting emotions in text data can be challenging and time-consuming, resulting in missed opportunities for businesses to improve customer experience and satisfaction.

2. Literature Survey

Emotion detection from short-text data is a challenging task due to the complexity and variability of human emotions. Traditional machine learning models, such as Naive Bayes and Support Vector Machines, have limitations in capturing the contextual and semantic relationships between words. To address these challenges, researchers have developed deep learning models, including the BERT-CNN model.

The BERT-CNN model combines Bidirectional Encoder Representations from Transformers (BERT) and Convolutional Neural Networks (CNN) for textual classification. BERT is used to train the word semantic representation language model, generating a dynamic semantic vector that is then placed into the CNN to predict the output. This model has shown significant improvements in emotion detection tasks compared to traditional machine learning models.

Müller, Martin et al.[1] (2020) used the BERT-CNN model to detect emotions in tweets related to COVID-19, achieving an accuracy of 83.03% outperforming several baseline models.

Wang et al.[2] (2021) used the BERT-CNN model for detecting emotions in customer reviews of online shopping platforms, achieving an accuracy of 83.5%, outperforming other state-of-the-art models.

Huang et al.[3] (2021) used the BERT-CNN model for emotion detection in online news comments and achieved an accuracy of 84.58%, outperforming other deep learning models.

Overall, these studies demonstrate the effectiveness and robustness of the BERT-CNN model for emotion detection from short-text data. The model has consistently shown improvement in accuracy and outperformed other state-of-the-art models in various applications. The BERT-CNN model is an important advancement in emotion detection research and has significant potential for real-world applications in industries such as marketing, customer service, and mental health.

Gupta et al[4], propose a comparative analysis of different deep learning models for emotion detection, including BERT-CNN, in their study "Emotion Recognition from Text using Deep Learning Techniques: A Comparative Study." They evaluate the models on three benchmark datasets and report their accuracy, precision, recall, and F1 scores. The authors conclude that BERT-CNN outperforms other models in terms of accuracy and F1 score.

Abas, Ahmed R, et .al[6], " BERT-CNN: A Deep Learning Model for Detecting Emotions from Text." ,proposes a BERT-based CNN model for emotion detection. They fine-tune the

BERT model on the training dataset and use the output embeddings as input to the CNN layers. The authors evaluate their model on two benchmark datasets and report their accuracy, precision, recall, and F1 scores. The results show that their model outperforms other state-of-the-art models.

Ansari, Kumar, and Mohammad propose a hybrid CNN and bidirectional LSTM model for emotion detection in their study "Emotion Recognition from Text using Hybrid Convolutional Neural Network and Bidirectional Long Short-Term Memory." They use pre-trained BERT embeddings as input to the CNN layer and then pass the output to the LSTM layer. The authors evaluate their model on four benchmark datasets and report their accuracy, precision, recall, and F1 scores. The results show that their model outperforms other state-of-the-art models. (Ansari et al., 2021).

3. Methods

The Workflow of our model can be described in the following steps. Firstly, we take the input that contains text with emotions. Next, we pre-process the data such as removing punctuations. The pre-processed data is then passed to the BERT Encoder. In the BERT Encoder the words are converted to vector form to pass to the CNN model.

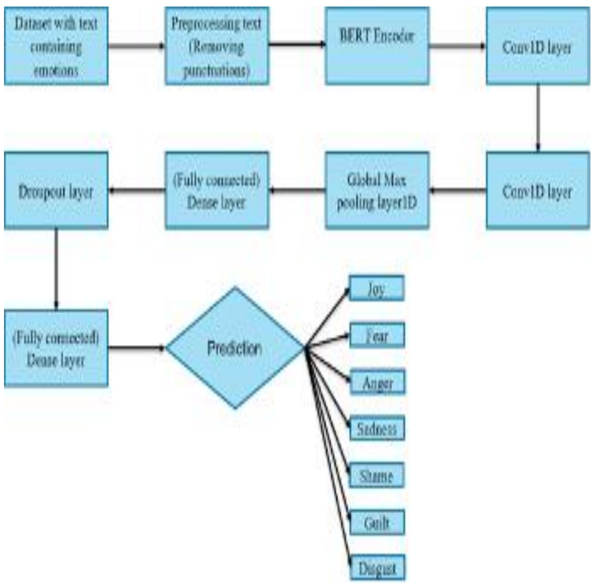

Fig. 1: Workflow of Proposed Model

In this section, we will explore the architectural components of the BERT-CNN algorithm, which comprises several layers. These layers include the input layer, preprocessing stage, BERT encoder, convolutional layer, pooling layer, dropout layer, and the final dense layers. Each of these layers plays a crucial role in processing and extracting meaningful features from the input data, contributing to the overall effectiveness of the proposed model.

Input layer:

The BERT-CNN model takes textual data as input, which can be in the form of sentences, paragraphs, or documents. The text is represented as a sequence of words or tokens. Preprocessing steps, such as tokenization, may be applied to split the text into individual words or subword units.

Preprocessing Stage:

The preprocessing steps for preparing text data for BERT input typically include tokenization, adding special tokens, and padding/truncation. Tokenization splits the text into tokens or subword units. Special tokens like [CLS] and [SEP] indicate the beginning and end of a sequence. Padding and truncation are used to ensure fixed-length input sequences.

Stop words removal is another common technique in text data preparation, where frequently occurring insignificant words are excluded. This step helps focus on more informative terms and can improve computational efficiency. The list of stop words can be customized based on the language or domain.

Once the preprocessing steps are completed, the preprocessed text data is fed into the BERT model. BERT uses its pre-trained transformer-based architecture to process input tokens, generate contextualized word embeddings, and perform downstream tasks like classification or information extraction.

BERT Model Architecture[5]:

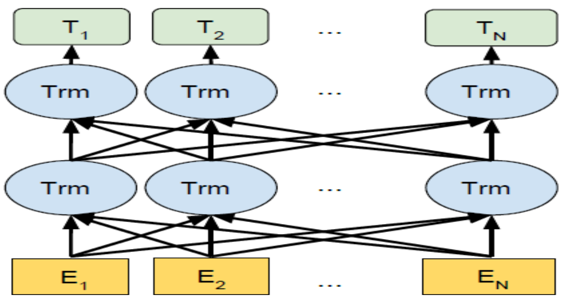

Fig. 2: Bert Architecture

1. Transformer Encoder: BERT uses a multi-layer Transformer encoder architecture, employing self-attention mechanisms for bidirectional context modeling. It consists of stacked layers with self-attention and feed-forward neural network sub-layers.

2. Input Embeddings: BERT converts variable-length token sequences into fixed-dimensional embeddings using token embeddings and segment embeddings. Token embeddings represent individual token meanings, while segment embeddings differentiate between sentences or segments.

3. Positional Encodings: BERT incorporates positional encodings to capture token positions in a sequence, enhancing understanding of word order.

4. Pre-training and Fine-tuning: BERT undergoes two steps: pre-training and fine-tuning. In pre-training, BERT learns from unlabeled text data using objectives like masked

language modeling and next sentence prediction. This enables BERT to acquire contextual word representations.

5. Attention Mechanism: BERT's attention mechanism models relationships between words in a sequence, assigning varying importance based on context. It captures dependencies in both directions.

6. Transformer Layers: BERT consists of multiple layers of the Transformer encoder. Each layer employs self-attention

and feed-forward neural network operations. Self-attention helps understand word relationships, while the neural network layers capture complex interactions.

7. Fine-tuning for Specific Tasks: After pre-training, BERT is fine-tuned on task-specific labeled data. This adaptation allows BERT to adjust its representations and parameters for specific tasks like text classification or named entity recognition

BERT Model Embedding Layer:

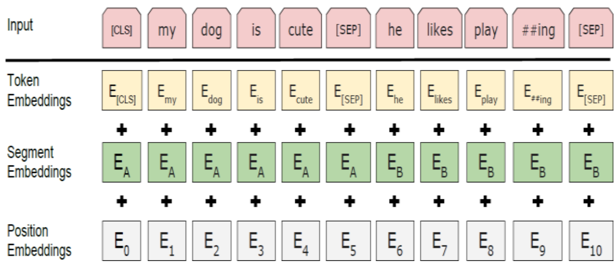

Fig 3. : Bert Layers

The embedding layer in the BERT model performs two key tasks. First, it generates word vectors or embeddings for the input sequence, capturing the contextual representations of words considering the surrounding context and dependencies. Second, it fine-tunes the pre-trained BERT model for a specific classification task by training it on a task-specific labeled dataset.

The word vectors or embeddings generated by BERT encode the semantic and contextual information of each word in the input sentence. These embeddings capture the nuanced meaning of words based on their context, enabling a more comprehensive representation of the input.

The fine-tuning process involves training the pre-trained BERT model on a task-specific dataset with labeled examples. This adaptation allows BERT to optimize its representations and parameters specifically for the classification task at hand.

By utilizing BERT as the embedding layer, these two tasks are efficiently accomplished. BERT generates contextually rich word embeddings, which serve as input to subsequent layers for further analysis or downstream tasks such as classification.

Convolutional Layers:

The convolutional layers in the model take the input as a matrix from the BERT embedding layer. These layers are responsible for extracting local features from the input data and reducing its dimensionality. Specifically, the convolutional layer operates on the semantic vectors by convoluting the data with N randomly generated filters.

By applying convolution, the model captures patterns and local dependencies within the semantic vectors. The randomly generated filters act as feature detectors, detecting specific patterns or combinations of features in the input data. As a result, the convolutional layers help in extracting meaningful and relevant information from the semantic vectors.

Furthermore, the convolutional layers also contribute to dimensionality reduction. As the filters convolve with the input data, they perform operations such as down sampling and pooling, resulting in reduced dimensions.

Pooling Layer:

The pooling layer operates on the output of the convolutional layers or other feature extraction layers. Its main function is to reduce the dimensionality of the feature maps while retaining the most salient information. Two commonly used types of pooling are max pooling and average pooling. We use max pooling in this paper

Dropout Layer:

The dropout layer is a regularization technique used to prevent overfitting in neural networks. It randomly sets a fraction of the input units or connections to zero during training. By doing so, dropout reduces the reliance of the network on specific input units and encourages the network to learn more robust and generalized representations.

Dense layers:

Dense layers, also known as fully connected layers, are a fundamental component of neural networks. They are responsible for learning complex patterns and making predictions based on the extracted features from the earlier layers of the network

3. Experimental Study

3.1 Dataset Description:

ISEAR(I)-DATASET-SIZE:(7666*2),This Data is formed by the International Survey on Emotion Antecedents and Reactions. The ISEAR dataset is a collection of emotional responses collected from participants who were asked to describe the emotions they experienced in response to various events or situations. It contains 7 emotions namely joy, sadness, fear, anger, guilt, shame, disgust.

Table. 1: Dataset Feature Description

Emotion labels	Quantity
Anger	1096
Disgust	1096
Sadness	1096
Joy	1094
Shame	1096
Guilt	1093

Fear	1095
Total	7666

3.2 Evaluation Metrics

Evaluation metrics were used to assess the performance of the proposed model. The metrics employed include F1 score, precision, and recall. Here is a description of these evaluation metrics along with their corresponding formulas:

1. F1 Score:

The F1 score is a commonly used metric in binary and multi-class classification tasks. It provides a balanced measure of the model's precision and recall. The F1 score is the harmonic mean of precision and recall, and it represents the overall effectiveness of the model in correctly classifying both positive and negative instances.

Formula:

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

2. Precision:

Precision is a metric that calculates the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). It measures the model's ability to minimize false positives.

Formula:

Precision = True Positives / (True Positives + False Positives)

3. Recall:

Recall, also known as sensitivity or true positive rate, calculates the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). It measures the model's ability to minimize false negatives.

Formula:

Recall = True Positives / (True Positives + False Negatives)

3.3 Hyper Parameters

The project was implemented on Google Colab with GPU runtime, taking advantage of the computational power provided by GPUs to accelerate the training and inference processes. Additionally, the project can also be implemented using Jupyter Notebook, which provides an interactive coding environment for data analysis and machine learning tasks.

Table. 2: Hyperparameters

Hyperparameters	Values
Learning Rate	4e-5
Loss Function	Sparse CategoricalCross Entropy
Epochs	10
Dropout	0.5
Optimizer	Adam

Table. 3: Results

ISEAR (DATASET)	BERT-CNN MODEL
ACCURACY	0.94
PRECISION	0.966
RECALL	0.986
F1-SCORE	0.976

4. Conclusion and Future Scope

In conclusion, the BERT-CNN model is an effective and accurate method for emotion detection in textual data. By leveraging the pre-trained BERT model and the CNN model for feature extraction, the algorithm is able to capture the emotional relationship between words and outperform traditional models. The implementation steps involve preprocessing of textual data, fine-tuning the BERT model on a labeled dataset, and constructing a CNN model to process the BERT output.Future scope of the project includes expanding the model's capabilities to detect more complex emotions, such as irony, sarcasm, and humor. Additionally, the algorithm can be applied to other languages and domains to improve cross-cultural communication and better understand different emotional expressions. Finally, the model can be enhanced with more advanced neural network architectures and techniques, such as attention mechanisms and transfer learning, to further improve performance. Overall, the BERT-CNN model has immense potential for real-world applications in emotion detection and sentiment analysis.

References

1. Müller, Martin, “COVID-Twitter-BERT: A natural language processing model to analyse COVID-19 content on Twitter.” Frontiers in Artificial Intelligence 6 (2020).
2. J. Wang, Z. Wang, D. Zhang and J. Yan, “Combining knowledge with deep convolutional neural networks for short text classification,” in Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification, Melbourne, Australia: International Joint Conferences on Artificial Intelligence, 2017.
3. H. Huang, X. Y. Jing, F. Wu, Y. F. Yao et al.,“DCNN- Bigru text classification model based on BERT embedding,” in 2019 IEEE Int. Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), ShenYang, China, pp. 632–637, 2019
4. Saxena, Anvita, Ashish Khanna, and Deepak Gupta. "Emotion recognition and detection methods: A comprehensive survey." Journal of Artificial Intelligence and Systems 2.1 (2020): 53-79.
5. J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.
6. Abas, Ahmed R., et al. "BERT-CNN: A Deep Learning Model for Detecting Emotions from Text." Computers, Materials & Continua 71.2 (2022).
7. D. Haryadi and G. P. Kusuma, “Emotion detection in text using nested long short-term memory,” 11480 (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 10, pp. 6, 2019.

8. K. Sailunaz, M. Dhaliwal, J. Rokne and R. Alhajj, "Emotion detection from text and speech: A survey," Social Network Analysis and Mining, vol. 8, pp. 1–26, 2018.
9. Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," IEEE Access, vol. 6, pp. 23253–23260, 2018.
10. M. Munikar, S. Shakya and A. Shrestha, "Fine-grained sentiment classification using BERT," IEEE, vol. 1, pp. 1–5, 2019.
11. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones et al., "Attention is all you need," in Advances in Neural Information Processing Systems, Long beach, California, USA: Curran Associates Inc., pp. 5998–6008, 2017.
12. W. Meng, Y. Wei, P. Liu, Z. Zhenfang and Y. Hongxia, "Aspect based sentiment analysis with feature enhanced attention CNN-biLSTM," IEEE Access, vol. 7, pp. 167240–167249, 2019.
13. Z. Gao, A. Feng, X. Song and X. Wu, "Target-dependent sentiment classification with BERT," in IEEE Access, vol. 7, ShenYang, AChina: IEEE, pp. 154290–154299, 2019.