2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) | 979-8-3503-3086-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAC3N60023.2023.10541692

# Comparative Analysis of Text based Emotion Detection on GoEmotions Dataset

Vinod Kumar
*Dept of Computer Science and Eng.*
*Delhi Technological University*
Delhi 110042, India
vinod_k@dtu.ac.in

Akshit Bansal
*Dept. of Computer Science and Eng.*
*Delhi Technological University*
Delhi 110042, India
bansalakshitwork@gmail.com

Umang Gupta
*Dept. of Computer Science and Eng.*
*Delhi Technological University*
Delhi 110042, India
umanggupta1975@gmail.com

***Abstract*—Today's environment relies heavily on social media, blogs with customer reviews, tweets, and comments for important communication. Understanding the feelings conveyed through these tweets and comments is crucial for improving a variety of fields, such as the detection of harmful online behavior (hate speech and the use of abusive language), a better understanding of customer review content, the development of chatbots with empathy, and many others. As a result, we introduce Go Emotions, the biggest humanly annotated dataset of 58k English Reddit comments that have been classified as either neutral or one of 27 emotion categories. The GoEmotions dataset's distinguishing feature is its emphasis on capturing a broad spectrum of emotions. GoEmotions includes a more varied and nuanced set of 27 emotion categories, in contrast to many other emotion datasets that often concentrate on few basic emotions (such as happy, sadness, rage, etc.). Not only do these categories cover fundamental emotions, but also more subtle ones like appreciation, amusement, gratitude, love, and others. Due to a wide range of categories, on the GoEmotions dataset for emotion classification, basic models like SVM (Support Vector Machines) and Naive Bayes might do worse than advanced models like CNNs, RNNs, and Transformers. Thus, applying the methodology from the base paper (BERT) and other advanced models, we present a comparative analysis of the same.**

***Index Terms*—Bert, CNN, emotion detection, GoEmotions dataset, RNN**

## I. Introduction

A crucial aspect of the human experience and social interaction is the capacity to recognize the emotions that are associated with every manifestation. Strong yet complex emotions can be expressed in a few words. Since 1997 (by Picard), the intention has been to make machines capable of understanding these nuanced emotions. [1]

There have been numerous attempts by various NLP academics to classify language-based emotions for a variety of domains, including the categorization of emotions for news headlines, tweets, and narrative sequences, among other things. However, each of these studies had a problem with their short datasets, which led to their coverage of a small set of emotions and their crude classification into Ekman and Plutchik emotions. [2]

In 2018, Bostan and Klinger produced an extraordinary attempt that illustrated the requirement for a sizable dataset. They identified 14 different emotions from a big dataset of 40K tweets. The largest number up until that point was 14, or 14 emotions. Their work highlights the need for a large-scale, consistently tagged emotion collection with high-quality annotations as opposed to a fine-grained taxonomy. [3]

Plenty of different categories of positive, negative, and ambiguous emotions. are included in our taxonomy, in contrast to other research and papers where a single emotion was used to classify the entire genre of that emotion. This qualifies it for later conversation interpretation tasks that call for a deft comprehension of emotion expressions, such as the analysis of customer feedback or the creation of chatbots.

A popular dataset for classifying emotions in text is GoEmotions. It was developed to record subtle feelings in user-generated content on social networking sites. The dataset includes a significant number of Reddit comments that users were asked to rate on a scale of one to five different emotions. [4]

The GoEmotions dataset excels at capturing a variety of emotions due to its emphasis on doing so. GoEmotions includes a more varied and nuanced set of 27 emotion categories, in contrast to many other emotion datasets that often concentrate on a small number of basic emotions (such as happy, sadness, rage, etc.). Not only do these categories cover fundamental emotions, but also more subtle ones like appreciation, amusement, gratitude, love, and others.

GoEmotions is an excellent fit for applications that call for fine-grained emotion analysis in text, like sentiment analysis, emotion detection, and emotion-aware recommendation systems,due to the availability of numerous emotion categories and the multi-label structure of the dataset. The GoEmotions dataset can be used by researchers and professionals to develop machine learning models for precisely identifying and comprehending emotions in text data.

### A. Objective

This study's objective is to compare the performance of various deep learning models for text emotion classification using the GoEmotions dataset, specifically RNN, CNN and BERT. The goal is to identify the model or models that work well together for accurately and efficiently categorizing emotions in textual input.

ISBN: 979-8-3503-3086-1/23/$31.00©2023 IEEE

### B. Workflow

- Data Visualization: To get a bigger picture of the dataset for better overall understanding through various data visualization techniques like n-gram exploration, charts and graphs etc.
- Dataset Preparation: Acquire the GoEmotions dataset, which contains a large collection of Reddit comments annotated with multiple emotion labels. Preprocess the dataset by tokenizing the text, handling any noise or inconsistencies, and splitting it into training, validation, and test sets.
- Model Implementation: Implement CNN, RNN, and BERT models for emotion classification. For CNN, design a convolutional architecture with suitable layers and filters. For RNN, choose an appropriate architecture such as LSTM or GRU. For BERT, on the GoEmotions dataset, fine-tune the pre-trained model.
- Training and Evaluation: Train each model on the training set and tune the hyperparameters for optimal performance. Utilize the proper assessment criteria, such as accuracy, precision, recall, and F1-score on the validation set, to assess the models.
- Comparative Analysis: Compare the performance of CNN, RNN, and BERT models in terms of accuracy and other relevant metrics. Analyze their strengths and weaknesses for emotion classification in the GoEmotions dataset. Identify scenarios where each model excels or struggles.
- Discussion and Conclusion: Explain the strengths and weaknesses of each model, summarize the results of the comparative study, and offer suggestions for choosing the best model for text emotion categorization using the GoEmotions dataset.

## II. Related Work

Due to varied emotion taxonomies, a dearth of trustworthy labeled data in many fields, and a highly subjective annotation standard, emotion identification from text is a difficult undertaking. Sentiment analysis is widely utilized in a variety of commercial fields to assess customer perceptions of products and services in order to improve them. Sentiment analysis gave rise to the crucial field of research known as emotion detection from text, which examines the entirety of a sentence in order to identify a more nuanced sentiment that is characterized on an emotional level [5].

The eight core emotions in Plutchik's wheel of emotions (1980) and the six fundamental emotions in Ekman (1984) are two often used categories for emotions. Most emotion datasets are created manually, hence they are typically lower in size. With 39k manually classified samples, CrowdFlower [6] which was created when Bostan and Klinger (2018) were conducting their research, was one of the largest manually labeled datasets in history.

Wang et al. [7] employed two machine learning algorithms to construct a huge dataset (about 2.5 million tweets) utilizing emotion-related hashtags in order to address the issue of the lack of annotated emotional content. They extended the 131 hashtag words for the seven fundamental emotions using Shaver et al.'s [8] mapping of hashtags to emotions. The study [4] introduced GoEmotions, the biggest manually annotated dataset of 58 thousand english Reddit comments classified as either neutral or 27 different emotions. The six main emotion categories—joy, anger, fear, sorrow, disgust, and surprise—as described by Ekman made up the majority of datasets in the sentiment analysis field prior to the development of GoEmotions and CrowdFlower. This is the most important feature that sets Go Emotions apart from all previous research on emotion datasets. The sole publication, Go Emotions, labels the data using 27 different emotions while remaining impartial. Recent research has revealed that deep neural networks are the best technique for classifying content in natural language. In a range of challenging subjects, deep learning generates cutting-edge insights. Numerous research have demonstrated deep learning-based sentiment categorization models that outperform conventional machine learning models as a result of deep learning's performance. However, the features of the dataset on which a deep learning-based sentiment classification model is trained determine the optimum model structure. In addition, this model structure is manually selected based on an expert's domain knowledge or is chosen from a grid search of viable candidates. Natural language processing tasks have been shown to greatly benefit from the use of recurrent neural networks (RNNs) [9]. In particular, their variations are Gated Recurrent Neural Nets (GRUs) [10] and Long-Short Term Memory (LSTM) [11]. Convolutional networks (CNN) [12] are an additional deep technique that is frequently used for picture classification and has a wide range of uses in the NLP field, including sentiment analysis of text.

Recent research has shown that the CNN is capable of learning a language's hierarchical structure and handling changing lengths effectively [13]–[15].

Using Twitter sentiment datasets and the Stanford Sentiment Treebank (SST), Kalchbrenner et al.'s experiment involved a network with two convolution and pooling layers. The proposed CNN structure outperformed traditional feedforward networks and the SVM, according to experimental findings. Other research, including [13], enhanced their performance by using appropriate regularization techniques, like dropout [16], as well as more convolution and pooling layers. Hu et al.'s research [17] showed that for sentiment analysis, deep-learning-based models outperformed conventional techniques like dictionary-based algorithms, SVM or Naive Bayes. On an Amazon review dataset, Kati and Milievi's [18] evaluation of CNN and LSTM performance However, because their experimental setting was not clearly defined, it was challenging for readers to derive useful recommendations. There were no performance comparisons offered in the nine studies that Zhang et al. [19] presented that used deep learning structures like the CNN and RNN. Since this is the case, the article [20] presents a classification approach based on deep neural networks, Bi-LSTM, CNN, and self-attention and demonstrates its effectiveness on diverse datasets. However,

using a transformer-based model with language model pre-training (BERT) has shown to perform at the cutting edge in a number of NLP tasks. Due to the better performance of this neural model, all of the top-performing models in the EmotionX Challenge [21] utilised a pre-trained BERT model. Recently, a more successful method for automatically determining the representation of words in the context of semantics using neural networks has become widely used. The author introduces the idea of word embedding in [22], which may be summed up as a "learned distributed feature vector to represent similarity between words."

[23] describes a word embedding representation. GloVe creates a word vector space using statistics to create a final model that outperforms word2vec in various NLP tasks like the analogies. . Furthermore, word2vec regularly performs worse than GloVe when learning time efficiency is taken into account.

## III. IMPLEMENTATION OF MODELS

We conduct three different sets of emotion classification experiments, including BERT as described in the foundational study. We use the GoEmotions dataset to test the efficacy of various deep learning models for classifying emotions in text, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and BERT (Bidirectional Encoder Representations from Transformers).

### *A. CONVOLUTIONAL NEURAL NETWORK (CNN)*

A Convolutional Neural Network is a Deep Learning technique that can recognise differences in images by taking in an input image, giving various objects and features in the image importance (learnable weights and biases), and then learning how to process the image. Comparatively speaking, a CNN's pre-processing time is much less than that of other classification techniques. Unlike simpler algorithms, CNNs can learn these filters and their characteristics. Embedding is the computer representation of a sentence. It is used to make the machine understand the language of man.

The code begins by fetching the data from CSV files and splitting it into training and test sets. It then proceeds to tokenize the text data, converting words into numerical tokens, and ensuring that all input sequences have the same length using padding. The tokenization process creates a mapping of tokens to words, which is used as input for the neural network.

Next, the code incorporates pre-trained word embeddings, specifically the GloVe embeddings, to represent words as vectors in a high-dimensional space. The embedding matrix is created based on the GloVe embeddings, and an embedding layer is added to the neural network to accept this matrix.

The neural network architecture consists of an embedding layer, followed by a convolutional layer with multiple filters, a dropout layer for regularization, and dense layers with rectified linear units (ReLU) activation. The output layer uses sigmoid activation to perform multi-label classification. The model is trained using the Adam optimizer and a binary cross-entropy loss function. A learning rate of 0.0002 is chosen through trial and error.

To prevent overfitting, the code includes a dropout layer that randomly deactivates a random 20% of nodes in each batch, making sure that the model doesn't rely too much on a single set of data. Additionally, callbacks are implemented to monitor the model's performance and save the best weights. The code also includes a function to find the optimal sigmoid threshold for classification. Precision, recall, and F1 scores are used to assess the outcomes once the model has been trained. The F1 score is calculated for each emotion at different threshold values, and the optimal threshold is determined based on the highest F1 score.

### *B. RECURRENT NEURAL NETWORK (RNN)*

The methodology begins with importing the necessary libraries, such as TensorFlow, Keras, NumPy, Matplotlib, and Hugging Face's nlp package. These libraries include resources for data visualization, natural language processing, and deep learning.

The data is then separated into the training, test, and validation sets after being imported using the pandas package. Each set contains tweets along with their corresponding emotion labels. The comments are tokenized using TensorFlow's built-in Tokenizer library, which assigns a unique numeric token to each word in the corpus. The tokenizer is trained on the training set of tweets to map the words to their corresponding tokens.

To ensure a fixed input size for the RNN model, the token sequences are padded or truncated to a maximum length of 50 words. This step prepares the data for input into the model. The emotion labels are converted into numeric representations using dictionaries. The labels are encoded with numeric values for multi-class classification.

Next, a sequential model is created using Keras. A dense output layer, a bidirectional LSTM layer, and an embedding layer make up the model architecture. The embedding layer maps the tokenized words to fixed-size vectors, the LSTM layer captures sequential dependencies in the data, and the dense layer produces a probability distribution over the six emotion classes. Layers of the model:

- Embedding layer
- Bidirectional LSTM: This indicates that the LSTM layer's contents can move from right to left as well as left to right.
- Dense Layer with 28 units to represent the 28 classes that are present. The target classes' probability distributions are returned by the activation function, which is set to softmax.

The sparse categorical cross-entropy loss function is used to build the model, which is then trained on the training data. On the test set, the model's performance is assessed, and metrics like accuracy and loss are computed. The model is then trained for a total of 10 epochs, or full iterations of the training dataset. This hyperparameter controls the duration of training and allows the model to learn from the data. Additionally, an

early stop callback is set up to monitor the validation accuracy. The training procedure is terminated early if the model does not demonstrate an increase in validation accuracy for more than two epochs.

### *C. BIDIRECTIONAL ENCODER REPRESENTATION FROM TRANSFORMERS (BERT)*

In 2018, Google Research researchers made their proposal. The main objective of it was to illuminate the meaning of Google Search-related questions. The method used in the Google research article [4] uses the GoEmotions dataset and the BERT (Bidirectional Encoder Representations from Transformers) model to classify emotions in text. Here is a summarized overview of the approach:

- Dataset Preparation: The GoEmotions dataset is split into training, validation, and testing sets. Text data is preprocessed by tokenizing sentences and adding special tokens for BERT.
- BERT Model: The pre-trained BERT model is based on a transformer architecture and captures contextualized word representations. The BERT base model consists of 12 transformer layers.
- Transfer Learning: The pre-trained BERT model is fine-tuned on the GoEmotions dataset using transfer learning. The last layer is replaced with a specific emotion classification layer.
- Training: The fine-tuned BERT model is trained on the training set using an optimization algorithm such as Adam. The model learns to predict emotion labels based on text input.
- Evaluation: The model's performance is evaluated on the validation set using metrics like accuracy, precision, recall, and F1 Score.
- Hyperparameter Tuning: Hyperparameters like learning rate, batch size, and training epochs are tuned to optimize the model's performance.
- Testing: The trained and optimized model is evaluated on the testing set to assess its performance on unseen data. Evaluation metrics are reported.
- Transfer Learning with BERT: The approach leverages BERT's pre-training on a large corpus of data to improve performance.

## IV. RESULTS AND COMPARATIVE ANALYSIS

The algorithms CNN, RNN and BERT are trained and then they are executed on the test dataset. In CNN, F1 scores are evaluated at different values of thresholds and the maximum F1 Score is achieved at 0.25 threshold. The evaluation metrics achieved namely F1 Score(macro), Recall Score and Precision Score for them are shown in table I.

TABLE I
COMPARISON OF MODELS ON THE BASIS OF DIFFERENT EVALUATION METRICS

| Metric | CNN | RNN | BERT |
|---|---|---|---|
| F1 Score | 0.43 | 0.41 | 0.46 |
| Recall Score | 0.45 | 0.42 | 0.63 |
| Precision Score | 0.41 | 0.40 | 0.40 |

Some observations based on the above table:

- BERT achieves the highest F1 score among the three models, indicating better overall performance in terms of balancing precision and recall.
- BERT has the highest recall score, indicating that it has a better ability to identify positive instances.
- All three models have similar precision scores, indicating their ability to minimize false positives.

Overall, BERT outperforms both CNN and RNN in terms of F1 Score and Recall Score, while all models have similar Precision Scores. It suggests that BERT may be the most suitable model for the given task, as it achieves a better balance between precision and recall. However, it is important to consider other factors, such as computational requirements and data availability, when selecting the appropriate model.

Next, F1 Scores for each emotion are evaluated to judge on which emotions, which algorithm performed better. The results gathered are shown in table II.

TABLE II
COMPARING THE F1-SCORES OF EACH EMOTIONS

| Emotion | CNN | RNN | BERT |
|---|---|---|---|
| Admiration | 0.53 | 0.51 | 0.65 |
| Amusement | 0.62 | 0.70 | 0.80 |
| Anger | 0.36 | 0.34 | 0.47 |
| Annoyance | 0.26 | 0.20 | 0.34 |
| Approval | 0.28 | 0.21 | 0.36 |
| Caring | 0.29 | 0.20 | 0.39 |
| Confusion | 0.28 | 0.23 | 0.37 |
| Curiosity | 0.45 | 0.43 | 0.54 |
| Desire | 0.32 | 0.30 | 0.49 |
| Disappointment | 0.21 | 0.23 | 0.28 |
| Disapproval | 0.34 | 0.36 | 0.39 |
| Disgust | 0.41 | 0.47 | 0.45 |
| Embarrassment | 0.36 | 0.39 | 0.43 |
| Excitement | 0.26 | 0.15 | 0.34 |
| Fear | 0.47 | 0.52 | 0.60 |
| Gratitude | 0.78 | 0.71 | 0.86 |
| Grief | 0.18 | 0.00 | 0.00 |
| Joy | 0.40 | 0.34 | 0.51 |
| Love | 0.69 | 0.61 | 0.78 |
| Nervousness | 0.30 | 0.29 | 0.35 |
| Neutral | 0.51 | 0.53 | 0.68 |
| Optimism | 0.40 | 0.30 | 0.51 |
| Pride | 0.28 | 0.15 | 0.36 |
| Realization | 0.13 | 0.06 | 0.21 |
| Relief | 0.21 | 0.02 | 0.15 |
| Remorse | 0.48 | 0.52 | 0.66 |
| Sadness | 0.38 | 0.34 | 0.49 |
| Surprise | 0.42 | 0.34 | 0.50 |

From the above table, it can be seen that in most instances,

RNN performed very poorly compared to Bert, and as far as CNN is concerned, it showed comparable results to Bert in some emotions. But overvall, Bert outperformed both CNN and RNN. For example:

- CNN was able to detect the emotion Grief with an F1 Score of 0.18 whereas both RNN and BERT had 0 F1 Score for that emotion.
- CNN was able to provide a better F1 Score of 0.21 for the emotion Relief as compared to an F1 Score of 0.15 of BERT.
- All three models showed the highest and comparable F1 Scores for the emotion Gratitude.
- F1 Scores of admiration and amusement were also among the highest and comparable between all three models.

Also these results show that certain emotions like gratitude, amusement, and love can be much more easily predicted in comparison to subtle emotions like relief, disappointment, realization, care and grief.

Next, the algorithms were trained and evaluated on sentiment grouped data on GoEmotions. The F1 scores for the sentiments for the 3 algorithms are shown in table III.

TABLE III
COMPARING F1-SCORES FOR SENTIMENT GROUPED DATA ON GOEMOTIONS

| Sentiment | CNN | RNN | BERT |
|---|---|---|---|
| Neutral | 0.63 | 0.46 | 0.67 |
| Positive | 0.72 | 0.64 | 0.82 |
| Negative | 0.53 | 0.69 | 0.70 |
| Ambiguous | 0.82 | 0.62 | 0.60 |

Some observations based on the above table:

- CNN outperformed both RNN and BERT in recognizing ambiguous emotions and showed comparable or better performance than RNN and BERT in recognizing neutral emotions.
- RNN performed relatively well for negative emotions than CNN and showed comparable results to Bert.
- CNN performed better than RNN for positive emotions.
- BERT generally achieved the highest F1 scores across most emotions, outperforming CNN and RNN in recognizing positive emotions and performing competitively in recognizing negative emotions with RNN.

Lastly, the algorithms were trained and evaluated on the Ekman Taxonomy which consists of emotions: anger, disgust, fear, joy, neutral, sadness and surprise. The F1 Scores for each of these emotions for the 3 algorithms are shown in table IV.

TABLE IV
COMPARING F1-SCORES OF EACH MODEL FOR THE EKMAN TAXONOMY

| Emotion | CNN | RNN | BERT |
|---|---|---|---|
| Anger | 0.50 | 0.55 | 0.57 |
| Disgust | 0.51 | 0.54 | 0.53 |
| Fear | 0.48 | 0.59 | 0.68 |
| Joy | 0.78 | 0.75 | 0.82 |
| Neutral | 0.64 | 0.62 | 0.66 |
| Sadness | 0.52 | 0.57 | 0.59 |
| Surprise | 0.63 | 0.36 | 0.61 |

From the above table, let's examine the emotions where CNN or RNN achieved higher F1 scores than BERT:

- CNN outperformed both RNN and Bert will recognizing the emotion surprise.
- RNN once again performed well for negative emotions compared to CNN whereas CNN outperformed RNN for positive emotions.

In summary, while BERT generally performed well across most emotions, there were specific instances where CNN or RNN outperformed BERT. CNN exhibited better performance in recognizing disgust and sadness. It's essential to consider these variations when choosing the appropriate model for specific emotions within the Ekman taxonomy.

## V. CONCLUSION

Several conclusions can be reached after applying deep learning algorithms for emotion categorization through text to the Ekman dataset and the GoEmotions dataset using CNN, RNN, and BERT.

Performance: On both datasets, BERT continually surpasses CNN and RNN models. BERT has an edge in accurately classifying emotions since it can take in contextual information and comprehend the semantic meaning of words. CNN and RNN also give comparatively better results when trained on the Ekman dataset than the GoEmotions dataset.

Complexity and Size of the datasets: The GoEmotions dataset has more training examples for the models because it is substantially bigger than the Ekman dataset. BERT performs better as a result of being able to use its pre-training on massive amounts of text data.

Flexibility: BERT generalises effectively to multiple emotion classification tasks thanks to its pre-training on a vast corpus of text data. It can efficiently capture subtle emotions and adapt to various datasets.

Differences between the two datasets: The GoEmotions dataset includes a wider spectrum of emotions, whereas the Ekman dataset concentrates on the six fundamental emotions (anger, disgust, fear, happiness, sorrow, and surprise) discovered by Paul Ekman. BERT can efficiently handle the range of emotions in the GoEmotions dataset thanks to its contextual knowledge.

## REFERENCES

[1] R. W. Picard, "Affective computing," Tech. Rep. Technical Report No. 321, M.I.T Media Laboratory Perceptual Computing Section, 1995.

[2] D. Matsumoto and P. Ekman, "Basic emotions," in *The Oxford companion to emotion and the affective sciences*, pp. 69–72, 2009.
[3] L. A. M. Bostan and R. Klinger, "An analysis of annotated corpora for emotion classification in text," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.
[4] D. Demszky, J. S. Cardoso, C. Potts, and D. Jurafsky, "Goemotions: A dataset of fine-grained emotions," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5875–5891, 2020.
[5] A. Boucouvalas and X. Zhe, "Text-to-emotion engine for real-time internet communication," in *Proceedings of International Symposium on Communication Systems, Networks and DSPs*, pp. 164–168, University of Peloponnese, 2002.
[6] CrowdFlower, "Sentiment analysis in text [dataset]." https://data.world/crowdflower/sentiment-analysis-in-text, n.d.
[7] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Harnessing twitter 'big data' for automatic emotion identification," in *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, pp. 587–592, IEEE, 2012.
[8] P. Shaver, J. Schwartz, D. Kirson, and C. O'Connor, "Emotion knowledge: Further exploration of a prototype approach," *Journal of Personality and Social Psychology*, vol. 52, no. 6, p. 1061, 1987.
[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
[10] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[12] Y. LeCun and et al., "Generalization and network design strategies," in *Connectionism in Perspective*, pp. 143–155, 1989.
[13] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, pp. 2359–2364, October 2015.
[14] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modeling sentences," 2014.
[15] Y. Kim, "Convolutional neural networks for sentence classification," 2014.
[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, June 2014.
[17] R. Hu, L. Rui, P. Zeng, L. Chen, and X. Fan, "Text sentiment analysis: A review," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, pp. 2283–2288, December 2018.
[18] T. Katić and N. Milićević, "Comparing sentiment analysis and document representation methods of amazon reviews," in *Proc. IEEE 16th Int. Symp. Intell. Syst. Inform. (SISY)*, pp. 283–286, September 2018.
[19] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.
[20] 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 (UMAP'19 Adjunct)*, (New York, NY, USA), pp. 63–68, Association for Computing Machinery, ACM, 2019.
[21] C.-C. Hsu and L.-W. Ku, "Socialnlp 2018 emotionx challenge overview: Recognizing emotions in dialogues," in *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, (Melbourne, Australia), pp. 27–31, Association for Computational Linguistics, ACM, 2018.
[22] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, February 2003.
[23] 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)*, pp. 1532–1543, 2014.