Future Computing and Informatics Journal

Volume 7
Issue 1 (2022) *Issue 1*

Article 3

2022

TEXTUAL EMOTION DETECTION APPROACHES: A SURVEY

Mahinda Mahmoud Samy Zidan
Mahinda Mahmoud Samy Ahmed Zaki Zedan, mahinda.zidan@fue.edu.eg

Ibrahim Elhenawy
Faculty of Computer and Informatics, Zagazig University, Egypt, ielhenawy@zu.edu.eg

Ahmed R. Abas
Faculty of Computer and Informatics, Zagazig University, Egypt, arabas@zu.edu.eg

Mahmoud Othman
Assistant Professor, msamy@fue.edu.eg

Future Computing and Informatics Journal

Homepage: https://digitalcommons.aaru.edu.jo/fcij/

doi: http://Doi.org/10.54623/fue.fcij.7.1.3

TEXTUAL EMOTION DETECTION APPROACHES: A SURVEY

Mahinda Zidan[1, a], Ibrahim Elhenawy[2, b], Ahmed R. Abas[2, c] and Mahmoud Othman[1, d]

[1]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt

[2] Department of Computer Science, Faculty of Computer and Informatics, Zagazig University, Egypt.

[a] mahinda.zidan@fue.edu.eg, [b] ielhenawy@zu.edu.eg,[c] arabas@zu.edu.eg,

[d] msamy@fue.edu.eg

ABSTRACT

Over the past decades, social media attracted individuals to express their feelings on any topic or item, resulting in an incremental growth in the size of created data. These feelings and unstructured data paved the path for business organizations to gather information and build statistical analysis. Various machine learning and natural language processing-based approaches are used for sentiment and emotion analysis. Moreover, deep learning-based approaches recently gained popularity due to their remarkable performance in text analysis. This paper provides a comprehensive overview of the prominent machine learning models applied in emotion analysis. It explores various emotion analysis taxonomies, in addition to the constraints of prevalent deep learning architectures. The paper also reviews some of the previously presented contributions in emotion analysis with a focus on deep learning methodologies as well as the most common datasets. It presents a comprehensive comparison between several emotion analysis models. This paper demonstrates the effectiveness of learning-based techniques in tackling emotion analysis challenges.

Keywords: Deep Learning; Text classification; Emotion models; Emotion detection; Natural Language processing.

Published by Arab Journals Platform, 2022

1. INTRODUCTION

Sentiment analysis, also known as opinion mining, is the computer analysis of people's perceptions, feelings, assessments, and attitudes about entities such as organizations, services, goods, events, topics, issues, and personalities, as well as their features [1]. The field's emergence and rapid growth are similar to that of social media on the Internet, including reviews, Twitter, blogs, microblogs, forum debates, and social networks. The goal of sentiment analysis is to distinguish between good, negative, and neutral emotions.

Emotion is a multidisciplinary field that includes computer science, psychology, and other disciplines. Emotions are defined in psychology as a psychological state that is accompanied by feelings, responses, attitudes and a level of pleasure or unhappiness [1] [2] [3]. Audio recordings, video recordings, and written documents can all be used to identify emotions. Emotion analysis from text documents appears to be difficult due to the fact that emotional words are not often used directly in textual statements, but rather result from a comprehension of the meaning of concepts and interactions of concepts expressed in the text document. Sadness, surprise, happiness, depression, fear, disgust, frustration, joy, rage, and other emotions are all examples of emotions. Other emotional models were also represented by the researchers.

The rest of this work is divided into the following sections. Section 2 discusses different forms of emotional models. Section 3 summarizes the existing models related to emotion detection tasks. Section 4 introduces the most commonly used datasets with a summary table and evaluation matrix. Section 5 presents the related work on emotion extraction with a summary table. Finally, Section 6 brings the work described in this survey to a conclusion.

2. EMOTION MODELS

Human emotions may be detected and categorized psychologically based on emotion type, intensity, and a variety of other factors, all of which can be combined and developed into emotional models. Emotional models are made up of scores, ranks, and dimensions that are used to categorize and define diverse human emotions.

The most widely utilized methods for detecting emotions are categorical and dimensional models. According to the categorical model, there are just a few basic and discrete emotions, each with its own function. The dimensional model, on the other hand, takes a different approach and describes emotions in a three-dimensional manner. In this technique, the dimensional model suggests that an emotional space is created and that each emotion is confined within this space.

The Ekman emotion model [4] is a well-known and commonly utilized categorization model, which defines six primary human emotions: surprise, happiness, disgust, fear, sadness and anger. These emotions are referred to as universal since they are realized in the same way throughout cultures and time periods. Ekman's emotion model has been employed in a variety of studies and systems that differentiate emotional states from textual input and facial expressions. The Ortony–Clore–Collins (OOC) emotional model [5] is another model that has been utilized in several researches on human emotion detection.

The OOC model describes 22 emotion types based on human emotional responses to a variety of scenarios, and it is primarily intended to simulate human emotions in general. It's also become the de facto standard for emotion synthesis, and it’s mostly employed in systems that observe emotions or create artificial characters with emotions.

Parrott's model [6] consists of a set of six basic emotions, including surprise, joy, anger, fear, love and sadness as well as a

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

three-level tree structure of emotions. The six basic emotions make up the initial level of this classification model, and each level after that refines the granularity of the previous one, making abstract sentiments more concrete. Parrot's method detects approximately 100 different emotions and it is the most difficult classification of emotions because it organises them into a tree-structured list. Plutchik's model of emotions [7]is a multi-dimensional model that specifies eight main bipolar emotions and proposes an integrated theory based on evolutionary concepts. Based on their bipolarity, these eight emotions are divided into four groups: surprise vs. anticipation, anger vs. fear, trust vs. disgust, and joy vs. sadness.

There are three levels to each emotion: The difference between serenity and ecstasy is that serenity is a milder form of joy, whereas ecstasy is a more intense form of joy.

Feelings can also be formed by combining the eight fundamental emotions. Joy and trust, for example, can be joined to make love.

The circumplex model of emotions was presented by Russell [8], where emotions are depicted in a two-dimensional circular space. The emotion's polarity is represented by one dimension of space, while the emotion's activation is represented by the other. The polarity dimension categorizes emotions as either positive or negative, while the activation dimension categorizes emotions as active or inactive.

3. EMOTION DETECTION TECHNIQUES

There are four different types of approaches for detecting emotions in text: keyword approach, Lexical / Corpus approach, learning based approach and hybrid approach. The following subsections define approaches for textual emotion identification. Fig.1 shows the different emotion detection techniques.

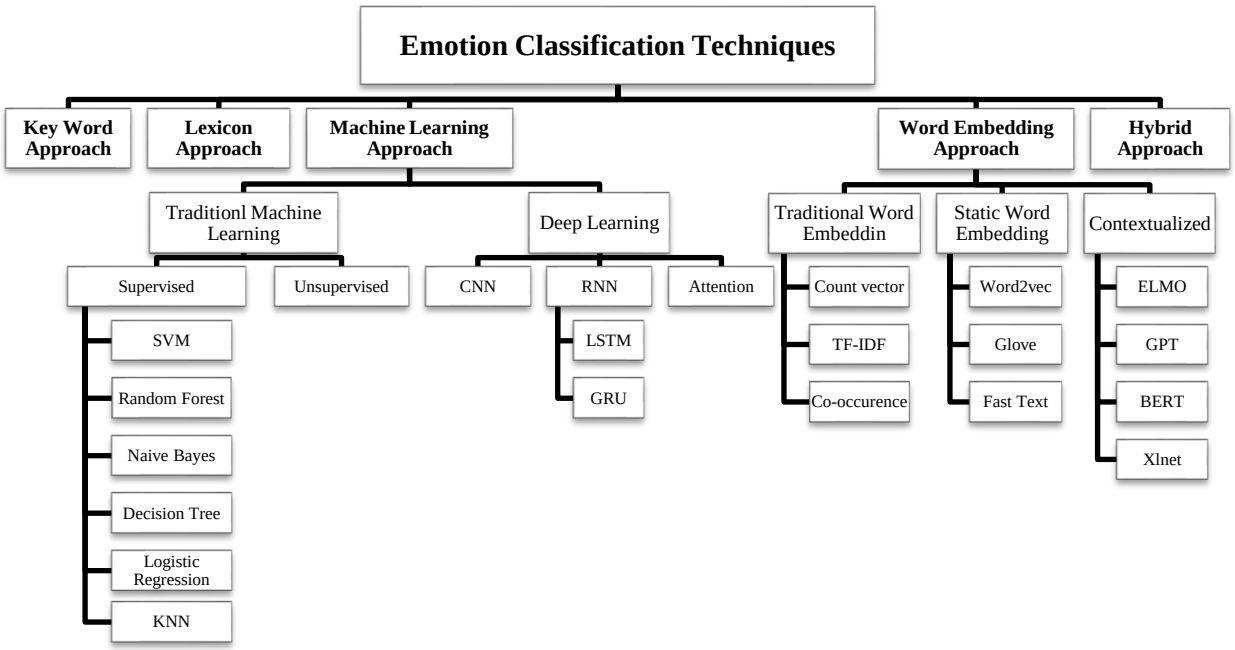

Figure 1: Emotion detection Techniques

Published by Arab Journals Platform, 2022

3.1 Key word Approach

The keyword-based strategy is the most traditional and straightforward in textual emotional analysis. It first looks for emotional terms in a text and then applies pre-defined rules and language to determine the emotion of a phrase. This technique makes use of rule-based dictionaries with many terms as well as emotional information. The keyword-based technique recognises emotions in phrases by using emotional ratings linked with each word. The vocabulary dictionary, Affin [9], Sentiwordnet [10], and the NLTK VADER Sentiment analyzer [11] all employ this method. The key word approach's limitations:

1) Ambiguity in Keyword Definitions

Emotion keywords are a basic strategy for detecting linked emotions; the meanings of the keywords might vary and be ambiguous. Except for words that stand for emotion labels, most words can have several meanings depending on context and usage, and it's simply impossible to include all such combinations in the EL set. Furthermore, in some extreme circumstances, such as sarcastic or cynical statements, even the most basic set of emotion categories (without all of their synonyms) could elicit multiple emotions.

2) Incapability of Recognizing Sentences without Keywords

The keyword-based strategy revolves around emotion keywords. As a result, phrases with no keywords infer that they lack emotions, which is obviously incorrect. For instance, both "I passed my qualified exam today" and "Hooray! I passed my qualify exam today" should imply the same feeling (joy), however the former without "hooray" may go unnoticed if "hooray" is not included "is the only word used to describe this emotion.

3) Lack of Linguistic Information Syntax structures and semantics

It also has an effect on how people express their emotions. For example, from the first person's perspective, "I laughed at him" and "He laughed at me" might imply opposite feelings. As a result, keyword-based techniques face a challenge when linguistic information is ignored. In summary, to detect emotions more precisely, keyword-based systems should detect not only the presence of terms, but also their linguistic content.

3.2 Lexicon Approach

The lexicon technique classifies text by applying an appropriate lexicon (a knowledge base containing text categorized according to emotions) to the input dataset. Emotion detection is similar to keyword detection, except that an emotion lexicon is utilized instead of a word list. The National Research Council of Canada (NRC), DepecheMood (DPM), Topic-based DepecheMood (TDPM), and EmoSenticNet (ESN) are some of the most widely used emotion and sentiment lexicons.

Joshi et al. [12] presented our emotion tracker called EmoGram. It consists of four stages:1- Tweet Downloader (to download tweets), (2) Tweet emotion scorer (to predict emotion in a tweet), (3) Emotion Scorer (to combine emotions in tweets of a given time period), and finally, a (4) Visualizer that places these emotions on a time axis, to generate emotion time sequence graphs. They evaluated the emotion tracker using a tweet dataset. They conducted the experiment with two lexicons: LIWC and Emo-Lex. The results they obtained after comparing the scores of each lexicon showed LIWC outperformed well when compared with Emo-Lex. Then, described three applications, each differing in (a) the text form, and (b) the way a time sequence is defined. The first application considered the time 515 sequence as a set of deliveries in a cricket match. For the second application, a play was considered to be a sequence of dialogues. In the third

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

application, we considered the recent Maggi controversy and validated how change in realworld events correlated with emotions in tweets. The limitations with their work was the Lack of detection of emotion interactions.

Tabak et al. [13] compared the performances of four lexicons, which are EmoSenticNet, NRC, topic-based DepecheMood and DepecheMood emotion lexicons, in terms of size, frequent words, variances and influence on the success of document classification into six emotions (Surprise, Joy, Sad, Fear, Disgust, Anger). They discovered that terms in the DepecheMood and NRC are typically classified into multiple emotion categories, but keywords in other lexicons are mostly classified into single emotion categories. Results indicated that NRC outperformed the other lexicons in the classification task.

Chuttur et al. [14] used NRC emotion lexicon technique to detect emotions from text. They collected data from 3000 online reviews for hotels. They detected eight emotions: trust, disgust, anticipation, surprise, joy, sadness, fear, and anger. The NRC achieved an accuracy of 76.9%.

Kusen et al. [15] Used a rule-centered approach for comparing the performance of three lexicons in order to find the best lexicon using social media texts. They evaluated the results of the NRC, EmosenticNet, and DepecheMood lexicons using various NLP approaches on Twitter and Facebook, despite the fact that these lexicons had different word-emotion pairs. They used ISEAR dataset with the results of a poll in which people were asked to assign emotions to Twitter and Facebook accounts in order to determine the ground truth. They showed that DepecheMood, EmosenticNet and NRC performed better in classification. NRC outperformed DepecheMood in identifying joy, fear, and anger, whereas NRC outperformed DepecheMood in detecting sadness. They claimed that insufficient words in all lexicons affected their model's overall performance.

Wang et al. [16] introduced a constraint optimization method for detecting emotions from social media. They employed a vector representation of emotions, which allows for text-based emotion ranking and thresholding. Moreover, their model is designed to extract a single or several emotions. They also presented various unique constraints such as topical and emotional constraints, as well as an efficient inference method based on the multiplicative update rule, which they proved to be convergent. They developed a generic method for automatically tuning model parameters from the training dataset. They evaluated the model on three different real-world datasets (semeval, ISEAR, and Twitter datasets) and found that it surpasses existing state-of-the-art algorithms for detecting emotions. They also thoroughly test each component of their model, demonstrating its durability in the face of noisy ground truth labels.

3.3 Machine Learning Approach

Machine learning algorithms, both supervised and unsupervised, are used to detect textual emotions, with a model that uses a sample of the dataset to train a classifier before putting it to the test with the rest of the data. In the supervised approach, a labelled emotion dataset is used to train and test the supervised classifier. The most widely used classifiers are Naive Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT). The data used in 'Unsupervised Classification' is not labelled with the classes. The classifier starts with a list of emotion-specific seed words, which are then compared to the phrases. In this way, relevant emotions are given to phrases. This helps to train the classifier model, which is subsequently applied to the testing data to label it. Despite the fact that the unsupervised method is more versatile, supervised classification is more accurate in most instances.

Published by Arab Journals Platform, 2022

We divided machine learning approach into two categories: first, traditional approaches; and second, deep learning approaches.

3.3.1 Traditional Approaches

This section discusses some of the traditional techniques, such as, SVM, NB, Logistic regression, KNN, Random forest and Decision tree architectures. Text classification applications have been successfully implemented using these designs.

1) Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a supervised learning technique for converting text data to vector format. The vector's dimension is the number of keywords. SVM is built on the principles of statistical learning theory and structural risk minimization. In the feature space, it calculates the maximum linear distance between multiple classes. It establishes the hyperplane, or the position of decision boundaries that offer the optimal result for class separation. "Maximal-margin-hyper-plane" is the name of this approach [17]. SVM maps both the non-linearity and the feature map when it encounters non-linearity between classes [18]. Kernels are responsible for this mapping. A hyperplane is generated as a result of this, which offers direct mapping to non-linear structure in the feature space. Figure 2 shows an illustration of such a hyperplane. SVM was created with binary classification in mind. When SVM is used to a multi-class classification issue, it separates the task into multiple binary classier problems, each of which is addressed with a massive number of SVMs [18].

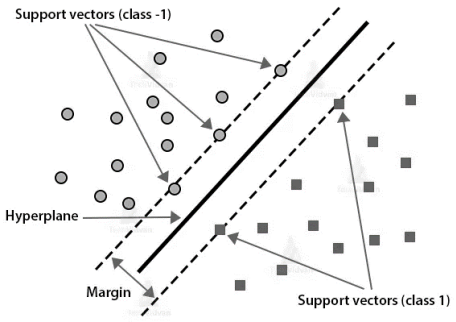

Figure 2: The structure of SVM

2) Naive Bayes Classifier

It evaluates each feature in the feature vector independently since they are all totally independent. In Naive Bayes, the conditional probability is described as follows:

$$P(X|Yj) = \sum_{i=1}^{m} P(x_i|y_i) \quad (1)$$

'X' stands for the feature vector, which is specified as $X = \{x_1, x_2, \ldots x_m\}$, and y_j stands for the class label. In our work, the Naive Bayes classifier effectively uses different independent variables such as emotions, emotional keywords, positive and negative keyword counts, and positive and negative hash tags count for classification. The associations between characteristics are not considered by Naive Bayes. As a result, it is unable to capitalize on the connections between the parts of speech tag, emotional keyword, and negation.

3) Logistic regression

Logistic regression is an effective classification method, where the output variable's probability is estimated based on a set of attributes. A binomial logistic regression is the challenge here [19], with two possible values for the response variable: 0 and 1. In binary classification, we suppose that x is a feature and y is the result, which might be 0 or 1. The likelihood that the output will be 1 given the input is described as follows:

$$P(y = 1 \mid z).$$
$$\log\left(\frac{P(x)}{1-P(x)}\right) = \beta_0 + \beta_1 X \quad (2)$$

The left-hand side is known as the logit or log-odds function, while $p(X) / (1 - p(X))$ is referred to as odds. The ratio of the probability of success to the probability of failure is known as the odds. Maximum likelihood estimate is the method that we use [20]. There are an endless number of

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

possible regression coefficients. A collection of regression coefficients having the best chance of finding the data we've seen is called a maximum likelihood estimate. When working with binary data, the probability of each result is simply 1 if it was successful and 0 if it was not.

4) Decision tree

Decision trees (DTs) [21] are fundamental supervised learning techniques used for classification. Conditional decision statements give rise to the concept of decision trees. The purpose on the provided dataset is to find the optimum tree with the lowest cross-validation error. When data is sparse, decision trees are useful, but they struggle with large-scale classification issues.

5) Random Forest

Random forests [22] are a classification learning approach that involves training a large number of decision trees and assigning a class label depending on the frequency of their outputs. Random forests are an effective barrier to decision trees tendency to over fit their training set.

6) K-Nearest Neighbor (KNN)

K-Nearest Neighbours (KNN) is a classification technique that uses a distance function between train and test data to obtain classification results and the number of nearest neighbors. The distance function is a cosine similarity function. The cosine similarity function is one of the most extensively used functions in document categorization to detect similarities across documents [23] The KNN scoring function is shown in (3). Document class is determined by voting on the K closest neighbour. The K document with the highest similarity value is the nearest neighbor.

$$\boldsymbol{score(c, d_1) = \sum_{d_2 \epsilon S_k d_1} I_c(d_2)\, cos(vd_1, vd_2)} \quad (3)$$

Score (c, d_1) = test document scores. d_1 denotes a test document, while d_2 denotes a training document. v_{d1} = vector testing document, v_{d2} = vector training document. I_C = 1 if d_2 is in class c, 0 otherwise. $S_K d_1$ = the set of k nearest neighbors in the test document.

Asghar et al. [20] compared Naïve Bayes (NB), Logistic Regression (LR), the Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Stochastic Gradient (SGD), Random Forest, and Back Propagation Neural Classifier (BPN) in terms of performance to find the most effective machine learning method for detecting emotions. They identified five emotions using the ISEAR dataset: joy, sadness, shame, fear, and guilt. Their findings revealed that the logistic regression algorithm surpassed the others with a precision of 67%, a recall of 67%, an accuracy of 66.58%, and an f-score of 66%. Their work has the following limitations:

1. They presented five emotion categories used in the experiment. A distinct combination of emotions, on the other hand, has not been tried.
2. The studies use just one data set ("ISEAR"), with a subsample of 5,000 data points.
3. Using the random splitting technique, the data in the experiment is separated into training and testing.
4. On traditional machine learning classifiers namely SVM, Naïve Bayesian, Logistic regression, Random forest, XGboost, KNN, Logistic regression, and SGD classifier, emotion detection experiments are being carried out.
5. Traditional feature selection procedures, such as TF-IDF and TF-IDF, are used in the studies and must be changed.

Suhasini [24] used a supervised machine learning technique to detect the four emotions from a Twitter dataset. They compared two machine learning methods, K-Nearest Neighbours (KNN) and Nave Bayes (NB). NB performed well when compared with KNN. Accuracy 72.60%.

Published by Arab Journals Platform, 2022

Nasir [25] created a graphical user interface (GUI) model to anticipate emotions based on Ekman's six primary emotions: joy, sadness, shame, anger, disgust, and guilt. They implemented four supervised Machine learning classification methods (SVM, Naïve Bayes, K-NN and Decision trees) to determine the best performance. Their results show that Naïve Bayes outperformed the other methods with an accuracy of 64.08%, however, the model had the limitation: Model can be made more refined and precise, no semantic analysis done and Model mainly worked on static dataset (ISEAR). They highlighted that when adding some features or rules-based approaches could improve system performance.

Ruposh et al. [11] Proposed a technique for categorizing Bengali texts into six groups (Happy, Anger, disgust, Fear and surprise) using SVM and Naïve Bayes. They created a corpus of 1200 emotional words. They obtained half of the data from the Cambridge English Corpus by translating material from English to Bengali using Google Translate, while the other half came from online blogs, Facebook pages, and Bengali newspapers. Their results showed that, the SVM outperformed the NB with an accuracy 73% as compared to 60%.

Alotaibi et al. [26] Presented a supervised Machine learning method for textual emotion detection using ISEAR dataset. They applied a preprocessing to clean the text prior to its feeding to the Logistic regression classifier. They detected five emotions that is, Sadness, Shame, Guilt, Fear and Joy. They conducted two experiments. The model's performance is evaluated using several evaluation measures including precision, recall and f-measure. Performed comparison of emotion classification results with different classifiers like KNN, SVM and XG-Boost. Their findings revealed that logistic regression (LR) surpassed the other classifiers, with a precision of 86%, recall of 84%, and F-score of 85%. Furthermore, they emphasized that using deep learning techniques might increase performance.

Winarsih et al. [27] examined the performance of four different classification algorithms for identifying emotions in Indonesian text, including Support Vector Machine-Sequential Minimal Optimization (SVM–SMO), Nave Bayes (NB), and Decision tree, K-Nearest Neighbor (KNN). To extract the features, the following preprocessing methods were used: tokenization, stop word, case normalization, stemming, and term. The best results were obtained by employing SVM-SMO.

3.3.2 Deep Learning

Deep learning architectures have already been employed in a variety of applications, such as natural language processing (NLP), pattern recognition, computer vision, and Multi-level feature representations can be learned using deep learning architectures. The architectures aim to find learning models that are built on numerous levels of hierarchical nonlinear information processing. Text mining applications have been successfully implemented using deep learning architectures such as CNN, RNNs, LSTM and GRUs, architectures.

1) Convolutional Neural Network (CNN)

CNNs [28] are data-processing architectures based on deep neural networks with a grid structure. On CNN, a unique form of the mathematical process known as convolution was used. One or more convolution layers have used the convolution procedure. A typical CNN design comprises an input layer, a hidden layer, and an output layer. CNN's hidden layers are made up of numerous layers, including convolutional, pooling, and fully connected layers. Convolutional layers extract feature maps from input data using the convolution method. In order to introduce nonlinearity to the design, activation functions such as RELU have been used in conjunction with feature maps. In pooling layers, the neuron clusters'

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

outputs have been merged. It increased the models' ability to accept overfitting by reducing the spatial scale of function spaces. Maximum pooling has been used in the pooling layer. To produce the architecture's ultimate output, the fully connected layers were used.

2) Recurrent Neural Network (RNN)

RNNs are neural networks that function with sequential data and have interdependent outputs and inputs [29]. This interdependence is usually helpful in anticipating the state of the input in the future. RNNs, like CNNs, require memory to retain the entire information gathered during the sequential process of deep learning modelling, and they typically only perform well for a few back-propagation stages. Figure 3 shows the basic architecture of RNN.

Where x_t denotes the input, s_t represents the hidden state beneath it, and o_t represents the output at time step t. The U, V, and W are hidden matrices' parameters, and their values can change with each time step. The hidden state is calculated using $St = f(U(x_t) + W_s(t_1))$.

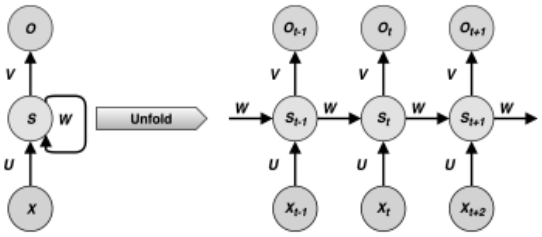

Figure 3: Basic Architecture of RNN

The sensitivity of the RNN to the loss of gradients is the key issue that impacts its overall performance [30]. In other words, throughout the training phase, the gradients may decay exponentially and be compounded by many tiny or large derivatives. However, over time, this sensitivity decreases, resulting in the forgetting of the initial inputs. LSTM is used to provide a block between the recurrent connections in order to avoid this problem. Each memory block holds the network's temporal states and contains gated units that govern the inflow of incoming data. The remaining connections are usually very deep, which helps to reduce the gradient problem.

3) Long Short-Term Memory (LSTM)

LSTMs [31] are a type of RNN-based deep neural network architecture. LSTM is one of the most widely used RNN variants, with the capacity to manage the vanishing gradient problem that affects standard RNNs and the ability to detect long-term dependencies. They become more powerful and versatile as a result of this.

Fig. 4 shows the architecture of LSTM. It is

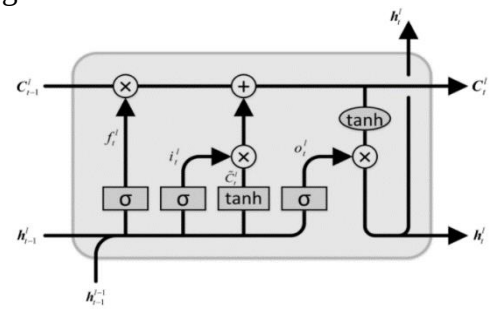

Figure 4: Structure of LSTM

made up of three gates: forget gate f_t, input gate i_t, and output gate o_t. The forget gate f_t determines which information to dump from a cell at time t by integrating the values of h_{t1} (prior state information) and x_t (current input) to produce a value of 0 or 1, with 0 indicating total dump and 1 indicating complete keep.

The state must then be updated, which is done by combining these two stages: In first step, the value to be updated is determined by the input gate. In the second stage, the *tanh* layer creates a vector with new candidate values C'$_t$, which will be sent to the cell's state. The previous state C_{t-1} is multiplied by forget gate f_t to create a new state C_t, and the amount by which the state is updated is determined by the new candidate values $i_t * C'_t$. A sigmoid layer is used to run the output gate o_t, which determines which part of the cell state will be used as an output. Finally, the cell state is sent through a *tanh* layer to keep the value between -1 and 1. The value obtained is multiplied by o_t.

Published by Arab Journals Platform, 2022

4) Gated Recurrent Units (GRU)

Gated Recurrent Units [32], or GRUs, are a condensed version of LSTMs aimed to alleviate the latter's computation problems that is introduced by Cho et al. in 2014. In LSTMs, the forget and input gates are merged into a single 'update gate'. By using a single 'reset gate', the cell state and hidden states are likewise combined and computed. The following are the operations that are currently being carried out:

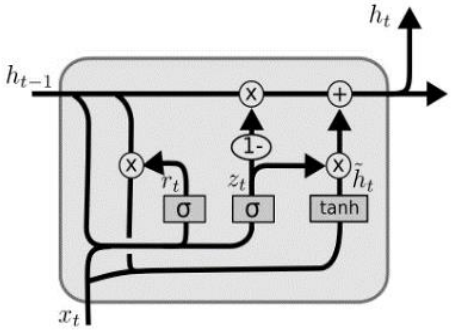

Figure 5: Structure of GRU

$$z_t = \sigma(w_z.[h_t - 1, x_t]) \quad (4)$$

$$r_t = \sigma(w_r.[h_t - 1, x_t]) \quad (5)$$

$$\tilde{h}_t = \tanh(w_c.[r_t * h_t - 1, x_t]) \quad (6)$$

$$\mathrm{h_t} = (1 - \mathrm{z_t}) * \mathrm{h_t} - 1 + \mathrm{z_t} * \tilde{\mathrm{h}}_\mathrm{t} \quad (7)$$

Unlike LSTMs, GRUs offer the advantage of controlling the flow of information without requiring an explicit memory unit. It uncontrollably exposes the node's hidden content. The performance is nearly same to that of LSTM, but with more efficient computing. However, when dealing with huge amounts of data, higher expressiveness LSTMs may produce superior outcomes.

Abdullah et al [33] designed a deep learning model based on CNN-LSTM for detecting emotions in Arabic tweets. Two experiments were carried out. In Experiment 1, a feed-forward phase was utilized, whereas in Experiment 2, a CNN-LSTM phase was used. In Experiment 1, they fed an input vector with a 4908-dimensional into a feed-forward network with three hidden layers comprised of 80, 200, and 500 neurons and fully connected layers, respectively, using the ReLU activation function. A sigmoid function was created as a consequence, which could predict sentiment and emotion levels. They used stochastic gradient descent (SGD) to optimize this network. In Experiment 2, they fed input with 300 vectors into a CNN with 3 kernel size, 64 filters, and ReLU activation. After applying a MaxPool with a size of 2 (The vectors were then fed into the LSTM model). The LSTM has two hidden layers with 80 and 200 neurons, respectively. As in Experiment 1, the SGD optimizer was utilized in conjunction with the ReLU and the Sigmoid activation functions. Experiment 1 had a 40% accuracy rate, whereas Experiment 2 had a 60% accuracy rate.

Shrivastava et al. [34] presented a model that combined convolutional Neural Networks CNN and self-attention by using a TV show dataset. Sadness, anger, disgust, fear, happiness, and surprise are all labeled separately. The proposed model consists of a 1D convolutional layer with max pooling operations and a nonlinear activation function to extract features from input data, and a fully connected layer with a softmax classifier to classify them into seven emotion classes. The filters are convolved across a sequence of words to extract the features. The convolution layer captures local and context attributes from the word vector representation, which is then sent through the max pooling layer to the fully connected layer. The CNN context characteristics are then sent to the attention model, which automatically focuses on the words with the greatest impact on classification and extracts the most significant semantic information in a phrase. The suggested model's testing accuracy on fine-grained emotions (7 emotions) is 77.54%, while course grained emotions (3 emotions) are predicted with an accuracy of 80.99%. On the basis of multiple evaluation metrics, the results are

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

compared to LSTM and a random forest classifier.

Haryadi et al. [35] used deep learning algorithms to classify emotions into seven categories (thankfulness, love, joy, anger, sadness, fear, and surprise). The authors used two methods: LSTM and nested LSTM. They had preprocessed data, 980,549 sentences for training, and 144,160 sentences for testing. Their findings demonstrate that LSTM outperformed SVM and Nested LSTM, with an accuracy of 99.167%.

Polignano et al. [36] created a model that combines self-attention, BiLSTM, and CNN. They claimed that word embeddings improve the performance of a text-based emotion detection system, so they concentrated on effective word embeddings to improve text-based emotion detection performance. Accordingly, they compared the results of 3 different word embeddings: GloVe, Google and FastText. using three different datasets which are ISEAR, SemEval 2019 Task3, and SemEval 2018 Task 1, and they embedded their intended model. In terms of the labelled emotion classes, GloVe and FastText performed much better in precision, recall, and F1 score, with the exception of sadness, where baseline techniques performed better. In the SemEval-2019 Task3 and SemEval-2018 Task1 datasets, FastText offered higher accuracy for their annotated emotion labels. However, they recommend using FastText embedding in future studies.

Ragheb et al. [37] proposed an attention-based model for identifying emotions from conversations using SemEval-2019 Task 3 dataset. It comprises dialogues that are annotated for anger, happy, sad, and other emotions. Their method consisted of two steps: the encoder and classification phases. The data was tokenized and entered into an encoder, after which Bi-LSTM units were trained using average stochastic gradient descent (ASGD). To reduce overfitting, dropouts were used between the LSTM units. Then, in order to focus on relevant emotion-carrying conversations, a self-attention mechanism was used, followed by average pooling. To classify data into the specified categories., a dense layer was employed, as well as a softmax activation. The model achieved an F1 score of 0.7582.

Park et al. [38] presented an emotion detection model using CNN. They collected emotional tweets data based on eight emotions. They created an emotional embedding model using the emotional twitter data, then they utilised the embedding model to extract emotions from the e ROCStories dataset. They use the NLTK VADER sentiment analyzer and emotional hashtags for emotion annotation in Tweet data to compute the cosine similarity between the selected emotional terms to identify the emotion of each phrase in the stories. For anger, anticipation, disgust, fear, joy, trust, sadness, and surprise, their accuracy was 0.367, 0.567, 0.55, 0.48, 0.733, 0.517, 0.45, and 0.43.

Fei et al. [39] introduced a variational-based model named the Implicit Objective Network (ION) for detecting implicit emotions. The model is comprised of two main components: the variational module and the classification module. The variational module stores the semantically rich representation in latent variables while recreating the input phrase with the Variational Auto-Encoder VAE. The classification module then takes advantage of this prior knowledge and employs a multi-head attention mechanism to efficiently extract information about the sentence's purpose. They carried out two experiments using two separate datasets, ISEAR and IEST. According to experimental results, the ION model outperforms strong baselines, achieving state-of-the-art performance. The ISEAR dataset has a precision of 0.755, a recall of 0.746, and an f1-score of 0.752, whereas the IEST dataset has a precision of 0.664, a recall of 0.647, and an f1-score of 0.658.

Rashid et al. [40] presented Bi-LSTM was presented to detect emotions in textual and emoji utterances such as sad, angry, and happy. They compared the performance of

Published by Arab Journals Platform, 2022

word2vec, glove, and fast text, three distinct word embeddings. The glove embeddings outperformed the fasttext and word2vec with an F-measure of 0.7185.

Karna et al. [41] Explored the effectiveness of LSTM based on deep learning methods for textual emotion detection. They detected seven emotions that is worry, Happiness, Sadness, Kindness, rage, Affection and Astonishment. They realized that LSTM outperformed the SVM and Nested LSTM with an accuracy of 94.15%.

Zhang et al. [42] explained a multi-label learning approach to emotion detection in online social networks. The goal of this study is to examine multiple-level emotion identification in online social networks from the user's perspective and develop a novel multi-label-based emotion detection system. They also presented a factor graph model to account for the previously mentioned correlations. They devised a multi-label learning method to overcome the problem. On the other hand, the suggested system had various problems that needed to be addressed, such as the usage of a limited size dataset owing to a lack of personnel to generate a larger dataset. In terms of study domains, these difficulties are still relatively new, yet they present an opportunity for scholars and development groups.

MA et al. [43] presented a deep learning model. The model has two steps: extracting features of each sentences and built the representation of dialogue from the features of three sentences. In the first stage, the embedding of each sentence is input into the Bi-LSTM to create the word representation for each word. Then the attention network is used to obtain the attention weight of the corresponding word, and the model uses the inner product of them to represent the word and feed it into the bi-LSTM layer. After the pooling step, the model receives the representation of each phrase. The features of the three phrases generated from the previous phase are sent into the LSTM layer as temporal information for emotion categorization in the classification layer.

3.4 Attentions

Although CNN and RNN can perform well in text categorization tasks, their drawbacks include a lack of intuitiveness and interpretability. Recently, attention mechanisms are also being developed based on the aforementioned architectures. In the area of natural language processing, attention mechanisms [44] are a typical mo del of LSTM mechanisms (NLP). The most significant distinction between CNN and RNN is that the attention mechanism-based model can visually display the contribution of each word to the results.

Ma et al. [45] proposed a model for identifying emotions from conversations based on a novel hierarchical attention network with residual gated recurrent unit framework. They obtained context-dependent representations for each token of each sentence in a conversation using a pre-trained BERT-Large model. HAN is created to gather long-range contextual information about the conversation structure. Furthermore, they add position embedding to the multi-head attention input in order to properly describe the model position information of sentences in a conversation. They used two textual dialogue emotion datasets to test their hypothesis (Friends and EmotionPush). These experiments indicated that their model outperforms state-of-the-art baseline approaches by a significant margin.

Saxena et al. [46] developed a model for recognizing emotions in conversations using two datasets: friends and Emotion Push. The model is a hierarchical attention network (HAN), with the first component being a word-level encoder with the attention layer, encoding each word in an utterance. The second component is an utterance-level encoder, which encodes every utterance in the dialogue. For emotion detection, the HAN is combined with a linear chain CRF classification layer.

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

The utterance level emotion detection problem is considered as a sequence labelling problem since the emotion in an utterance is dependent on the emotions of previous utterances.

3.5 Word Embedding Approach

Word Embeddings are a numerical vector representation of corpus text in which each word in the corpus vocabulary is mapped to a set of real-valued vectors in a pre-defined N-dimensional space. It tries to capture the contextual, syntactic meaning and semantic of each word in the corpus vocabulary based on its use in sentences. Words with comparable contextual meanings and semantic have similar vector representations, however each word in the lexicon has its own set of vector representations.

Traditional word embeddings, static word embeddings, and contextualized word embeddings are the three types of word embeddings.

1) Traditional word embedding

Based on frequency, which evaluates the entire document and determines the relevance of rare words, counts occurrences of each word, and word co-occurrences.

Traditional word embeddings are categorized into Count vecror [47] TF-IDF [48]and co-occurrence [49].

2) Static word embedding

Prediction based approach that transforms each word into a vector and assigns a probability to each word.

Lookup tables, which convert words into dense vectors, are used to train static embeddings. The context of this embedding does not change once it is learned, and the embedding tables do not change between sentences. Word2vec [50], Glove [51], and Fast text [52] are the three types of static word embeddings.

Word2vec [50] an approach that combines two models: continuous bag-of-words (CBOW) and continuous skip-gram. The CBOW model predicts a current word using the average/sum of context words as input. The skip-gram model predicts each contextual word by using the current word as input. Word2vec contains fewer dimensions than previous embedding approaches, making it more flexible, faster, and more adaptable to a wide range of NLP tasks. Despite its great generality, it cannot be flexibly modified for individual purposes or to address the polysemy problem.

GloVe [51] is a tool for word representation that uses a count basis and statistics from the global corpus. It generates global co-occurrence statistics with a fixed-size context window first, then utilizes stochastic gradient descent to minimize its least squares objective function, thereby factorising the log co-occurrence matrix. It can enable parallelization and is quite fast, but it consumes more memory than word2vec.

The fastText [52] can deal with out-of-vocabulary (OOV) words by predicting their word vectors using learnt character n-grams embedding Despite the fact that it needs low training time and does not share parameters, it has poor generalisation for large output spaces.

3) Contextualized word embedding

Language models that have been pre-trained learn semantic global and considerably boots NLP tasks, such as text classification, successfully [53].

Unsupervised approaches are typically used to automatically mine semantic knowledge and then generate pretraining objectives so that machines can learn to grasp semantics. Pre-trained language modelling has produced new results on numerous downstream NLP applications such as, data classification, question answering and sentiment analysis among others, and can be considered an equivalent to ImageNet in NLP [54]. Feature-based and fine-tuning language models are the two types of pre-trained language models. ELMO [55], OpenAI GPT [56], BERT [57], and XLNet [58] are examples of pre-trained models

Published by Arab Journals Platform, 2022

that may be fine-tuned to specific NLP tasks.

ELMo [55] is a model-independent deep contextualized word representation model. It can learn alternative representations for varied linguistic settings and model complicated word features. The bi-directional LSTM is used to learn each word embedding based on the context words.

GPT [56] uses supervised fine-tuning and unsupervised pre-training to build general representations that may be used to a variety of NLP tasks with minimal adaption. Furthermore, the target task's domain does not have to be the same as the unlabeled datasets. The GPT algorithm's training approach usually consists of two stages. First, a modelling objective on an unlabeled dataset is used to train the initial parameters of a neural network model. The associated supervised goal can then be used to accommodate these parameters for the target task.

BERT [57] presented by Google significantly improves performance on NLP tasks, such as text classification, by pretraining deep bidirectional representations from unlabeled text using joint conditioning on both left and right context in each layer. It is fine-tuned by adding just one more output layer to develop models for a wide range of NLP Tasks, such as machine translation, question answering and sentiment analysis In comparison to these three models, ELMo is a feature-based LSTM methodology, while BERT and OpenAI GPT are fine-tuned Transformer techniques. Although BERT and ELMo are bidirectional training models, OpenAI GPT is a left to right training model. As a consequence of integrating the benefits of ELMo and OpenAI GPT, BERT obtains a superior result. Transformer-based models are popular for NLP applications because they can parallelize computation without taking sequential information into consideration, making them suitable for large datasets. As a result, other works are used for text classification tasks, and they perform excellently.

XLNet [58] is a pre-training approach that uses a generalized autoregressive model. To learn the bidirectional context, it optimizes the predicted likelihood across all factorization order permutations. It can also use an autoregressive formulation to overcome BERT's flaws and incorporate Transformer-XL principles into pre-training.

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

Table 1. Word embedding, description, advantage and disadvantage

Word Embedding		Description	Advantage	Disadvantage
Traditional embeddings	Count vector	Technique counts the number of occurrences of each word in a text.	It may be used to produce accurate word counts on real-world text data.	similarity between documents and semantic information were not allowed.
	TF-IDF	calculated by multiplying a word's frequency in a sample by the frequency of the word in the entire text.	It can easily determine how similar two documents are.	Semantics and co-occurrences in various documents were not captured.
	Co-occurrence vector	Built on the idea of comparable terms occurring together in the same context.	It has the ability to keep the semantic relationship between the words and promotes it.	To store the co-occurrence matrix, a huge memory size was required.
Static word embeddings	Word2vec	The technique generates word embedding using dense representation. It's a model for assigning probability to terms that performed well in word similarity tests.	It can convert unlabeled data into labelled data by matching the target word to the context word. Sub-linear relationships are not implicitly specified.	Lack of global information adaption.
	Glove	An unsupervised model for creating word vectors. glove has two components: the local context window method and the global matrix factorization method.	Glove uses both local and Global statistics to understand the meaning of words (word co-occurrence). It is capable of deducing semantic links. It can predict surrounding words by using a log-bilinear model with a weighted least square objective to maximize probability. Sub linear relationships can be comprehended using word vectors.	Glove needs more memory for storage
	Fast text	An extension of word2vec. n-grams are used to classify the words. It provides an efficient vector representation of uncommon words.	Based on word fragments and may provide vector representations for words that are not found in OOV terms (dictionary). Fast text can handle invisible words.	Fast text doesn't provide any contextual information.

Published by Arab Journals Platform, 2022

Word Embedding		Description	Advantage	Disadvantage
Contextualized word embeddings	ELMO	A character-based and context-dependent. Depending on the context in which a term is used, it may have many meanings.	For a single word, ELMo delivers several word embeddings.	ELMo is a shallow bidirectional system because it can't use both left and right contexts at the same time.
	BERT	The first deep unsupervised bidirectional system that employs a multi-layer bidirectional Transformer encoder.	used to understand the contextual relationships between words or subwords. It stores the syntactic and semantic meanings of text. It can make assumptions for the empty word in between sentences.	Only short sentences can be processed by BERT.
	GPT	Generative Pre-Trained Transformer 2 (GPT2). It is a decoder only transformer	It can predict the next word by looking at parts of a sentence. Probability estimation can be used to give more than 10 possible predictions for the next word.	GPT-2 necessitates a lot of computation and has a high probability of producing incorrect results because it is trained on millions of websites.
	XLNET	Used a generalized autoregressive model.	Maximize the log probability of all feasible factorization sequences to discover bidirectional context information. Used the features of auto-regression to overcome the limitation of BERT	Under perform on short sequences. XLNET takes longer to train and to infer.

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

3.6 HYBRID APPROACH

A hybrid approach for emotion detection in text combines any two or all of the methods mentioned to gain the benefits of several methods while achieving the highest level of accuracy. Previous studies have demonstrated that integrating several emotion detection methods produces better results than using separate approaches.

Shah et al. [59] a hybrid technique that combines the usage of a lexicon and learning-based methods. They compared the performances of two lexicons, which are EmoSenticNet and WordNet-Affect. They also used different classification algorithms (SVM, decision trees, Naïve Bayes,). The SVM provided the best results, therefore it was selected. The results showed that EmoSenticNet outperformed the WordNet-Affect lexicon with the help of SVM, with an accuracy of 89%. Furthermore, they emphasized that the use of deep learning techniques could enhance performance.

Huang et al. [60] combined the hierarchical LSTMs model and the BERT model in order to detect emotions from tweets. The data was preprocessed after it was collected, with the emoji package being used to extract and translate emojis into texts (which made up a larger fraction of the overall data) and the ekphrasis package being used to manage normalize tokens and misspellings. After applying the Bert-Large pre-trained model with 24 layers, the hierarchical LSTMs model was employed. This was done to ensure that semantics in text expressing emotions could be retrieved correctly. In angry, happy, and sad emotions, the model received a macro-F1 of 0. 779.As a result of the large number of multiclassified emotions, there was an increase in misclassification.

Hasa [46] n introduced a model for identifying emotions from texts called Emotex. This model is based on emotion dictionaries and supervised learning approaches. Their approach is comprised of two parts: an offline and an online classification task. The offline task entailed the implementation of their Emotex technology, which allows them to create models for emotion classification. Emotex was constructed using classifiers like NB, SVM and decision tree to classify emotiona from Twitter. To create the training datasets for the classification model, the data was preprocessed and feature vector constructions were performed. The online technique uses the offline approach's model to classify live streams of tweets in real time. The model achieved an accuracy of 90%, but it had some loose semantic features.

Adoma et al. [61] designed a model using ISEAR dataset .Model consist of two stages: Bert fine-tuning and Bi-LSTM. BERT fine-tuning was based on the Transformer model architecture, the self-attention mechanism that learns contextual relationships between words in a text. The data after collection was tokenized and feed in to an encoder followed by Bi-LSTM. Bi-LSTM is a classifier stage, which is, consists of four layers: Input layer, Mask layer, Bidirectional LSTM layer and dense layer. The input layer received the output of BERT. The softmax activation function was utilized to extract the seven emotion classes (shame, disgust, joy, anger, sadness, fear, and guilt) using a bidirectional layer made up of 100 neurons and a dense layer containing seven units. The model achieved an f1-score of 73%.

Al-Omari et al. [62] proposed a system called Emodet2 that uses a deep learning approach. The model determined the sentiment and emotions in English textual conversations and classified them into four emotions: angry, sad, happy, and others based on semeval 2019 task 3. Emodet2 has been built by assembling models with different sub models. The Glove, BERT embeddings, and a collection of psycholinguistic characteristics make up the major inputs to the proposed model. Furthermore, Emodet2 combines a fully connected neural network with BiLSTM. They obtained an f-score of 0.748.

Published by Arab Journals Platform, 2022

4. DATASETS AND PERFORMANCE EVALUATION

This section shows the most frequently used datasets in recent research, along with a brief explanation of the amount of data in each dataset. The descriptions of each dataset are shown in Table 2. Also, this section presented the adopted evaluation metrics.

Table 2: Datasets for detection emotions from text

Dataset	Data size	Description
ISEAR[63]	7666 Sentence	obtained from 1096 people from various cultural backgrounds who completed questionnaires on seven emotions. Emotions: Shame, Disgust, Joy, Anger, Sadness, Fear, Guilt.
EmoBank[64]	10k English sentences	collected from news headlines, blogs, fiction, letters, newspapers, essays and travel guides of writers and readers, thus spanning a wider domain. Annotated for six emotions (disgust, happiness, surprise, fear, anger and sadness).
Daily Dialog[65]	13118 Dialogue	collected from conversations and annotated for seven emotions Emotions: Surprise, Fear, disgust, sadness, Anger, happiness and others
CrowdFlower	40K Tweets	Constructed from tweets and Classified into 13 emotions (Love, sadness, worry, Empty, Happiness, Relief, Anger, Fun, Boredom, Hate, Surprise, Enthusiasm, Neutral)
The valence and arousal Facebook post [66]	2895 Facebook	Constructed from Facebook posts
Affective Text	1200 Headlines	Annotated for six Ekman emotions and the polarity orientation.
WASSA-2017 Emotion Intensity (EmoInt).[67]	7097 tweets	Constructed from tweets and Classified into 4 emotions (Anger, Sadness, Joy, Fear)
(EmoLex)	14,182 unigrams (words)	Annotated with negative and positive emotions such as anticipation, trust, fear, sadness, anger, surprise, joy disgust.
SemEval2019 Task 3	38,424 Sentences	Collections of labeled conversations. Each conversation consists of three turn talk between two persons. Conversation classified in to 4 classes (Happy, Anger, Sad and Others).
SemEval 2017 Task 4	1250 text	Collected from headlines, tweets, Google news and other major newspaper. Annotated for 6 basic emotions.

There are a variety of metrics that can be used to assess the efficacy of any approach. Accuracy, recall, precision, and F1-measure are some of the most widely used metrics. The ratio between the number of texts that were accurately classified, and the total amount of texts is known as accuracy. The ratio of accurately identified texts among all texts belonging to that class is therefore defined as recall. While precision is defined as the proportion of correctly classified texts to all texts that can be assigned to a class. Finally, for precision and recall, the F1-measure can be applied to the symmetrical average. These measurements are used to determine how well a technique performs in the end. Furthermore, these values can be anticipated using the contingency table for a specific test set (see Figure 6). Figure 6 shows how the metrics are calculated using a confusion matrix.

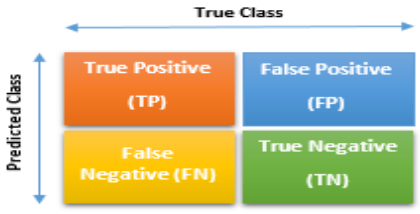

Figure 6: Confusion metrics

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

$$F - score = \frac{(2*Precision*Recall)}{(Precision+Recall)} \quad (11)$$

5. SUMMARY OF EXISTING APPROACH

The following section summarizes the emotion detection techniques employed by researchers, as well as their descriptive features and limitations, datasets and emotion recognized used, and results (Shown in table.3).

Table 3. summarizes the emotion detection techniques employed by researchers, as well as their descriptive features and limitations, datasets and emotion recognized used, and results

Reference No.	Approach	Dataset	Emotions	Methodology	Limitations
[20]	Machine Learning	ISEAR	Fear, Guilt, Joy, Shame and Sadness.	Applied several machine learning algorithms (LR, KNN, SVM, Random Forest, SGD, NB and BPN) to determine the best performance used for emotion detection. LR outperformed others with an accuracy of 66.58%.	Only one dataset used in their experiments. Works with five emotions however, data set consists of seven emotions.
[24]	Machine Learning	Twitter messages	HappyActive, UnhappyActive HappyInactive, and UnhappyInactive	Demonstrated the effectiveness of Nave Bayes in contrast to the KNN. Accuracy 72.60%.	There is a Low extraction of contextual information.
[25]	Machine Learning	ISEAR	Anger, Shame, disgust, Guilt and sadness	Build a GUI to predict emotions. Implemented four supervised machine Learning Algorithms (NB, SVM, KNN and decision tree) NB outperformed others with an accuracy of 64.08%.	No semantic analysis done. Model mainly worked on static dataset (ISEAR). Model can be made more refined and precise
[11]	Machine Learning	Build data set from Cambridge English Corpus, Facebook pages, online blogs, Bengali newspapers	Surprise, Happy, disgust, Anger, disgust and Fear	Created a new corpus based on Bengali text. Used SVM and NB to detect emotions. SVM achieved an accuracy of 73% and NB achieved an accuracy of 60%.	The use of a small dataset has an impact on the generalization and standardization results. Works with Bengali Language only. Absence of results on benchmark dataset.
[26]	Machine Learning	ISEAR	Sadness, Shame, Guilt, Fear and joy	Used LR to predict emotions. Evaluated Performance of model using different evaluation metrics and compared model with different machine learning algorithms Results: Precision: 86% recall: 84% F- Score :85%.	Used only one dataset. Works with five emotions however, data set consists of seven emotions.
[27]	Machine Learning	Various websites	Fear, sadness, joy, surprise, disgust and anger.	Presented different machine learning algorithms (SVM, Decision tree, KNN, NB). SVM achieved the best performance of 80.86 % accuracy.	Works with Indonesian Language only. No syntactic or semantic features are used.
[68]	Hybrid approach	Tweets	HappyActive,,Unhappy Active and Happyinactive Unhappy Inactive	Applied three different algorithms (SVM, decision tree and NB). Achieved an accuracy of 90%	Loose semantic feature extraction

Published by Arab Journals Platform, 2022

Reference No.	Approach	Dataset	Emotions	Methodology	Limitations
[33]	Deep Learning	Arabic tweets	Anger, Joy, Sadness, Fear, Sentiment.	Compared the performance of feedforward neural networks with CNN-LSTM. Achieved an accuracy of 40% and 60%, respectively.	Used a small amount of data and small number of hidden layers.
[34]	Deep Learning	Build a new corpus from a TV show transcript.	happiness, surprise, sadness, anger, fear, disgust and neutral.	Introduced an attention mechanism based on CNN. Results: fine-grained emotions with an accuracy of 77.54%, while coarse-grained emotions are predicted with an accuracy of 80.99 %.	The time complexity of the training classifiers has not been considered. Also, they used a small data set. Training dataset can be expanded by collecting more seasons of a TV show to avoid overfitting.
[35]	Deep Learning	Twitter	Thankfulness, Surprise, Sadness, Joy, Anger, Fear, and Love.	Implemented two Methods LSTM and Nested LSTM. Achieved a very high accuracy of 99.167%.	Only one dataset used in their experiments The authors' results were not significantly different from those of other models, which was a problem that needed to be addressed.
[61]	Hybrid approach	ISEAR	Shame, Disgust, Joy, Anger, Sadness, Fear, Guilt.	Designed a model based on encoders and transformers. F1 score: 73%.	Lack of vocabulary words. To overcome this limitation, they can improve the word embedding and thus increase the likelihood of success for this proposed model.
[39]	Deep Learning	ISEAR and IEST	ISEAR: Shame, Disgust, Joy, Sadness, Fear, Guilt, Anger. IEST: sad, disgust, fear, anger, surprise,and joy	Proposed a variational based model named as Implicit Objective Network for implicit emotion detection. F1-score: 75.2 % on ISEAR dataset and 65.8 % on IEST dataset.	Does not perform well for identifying the sadness emotion.
[45]	Deep Learning	Friends and EmotionPush	Friends: Sadness, surprise, joy, anger and neutral. EmotionPush: sadness, joy, surprise and neutral.	Proposed a new Hierarchical attention network based on residual gated recurrent unit for detecting emotions from conversations. Accuracy: 76.5 % on Friends dataset. 69.9 % on emotion push dataset.	Satisfactory accuracy results.
[46]	Deep Learning	Friends and EmotionPush	Friends: Sadness, surprise, joy, anger and neutral. EmotionPush: sadness, joy, surprise and neutral.	Used Hierarchical Attention Network (HAN) model to learn data representation at both utterance level and dialogue level. They formalized the problem as sequence labeling task and used a linear CRF as a classification layer. Accuracy: 55.38% on Friends. 56.73% on EmotionPush	Accuracy wasn't very high. They did not compare their model with different model to verify the effectiveness of their model.

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

Reference No.	Approach	Dataset	Emotions	Methodology	Limitations
[40]	Deep Learning	Semeval 2019	Happy,Anger and sad	Used Bi-LSTM to detect emotions from text Bi-LSTM obtained an f1-score of 0.7185.	Restricted catrgories of emotion classes.
[36]	Deep Learning	ISEAR. Semeval 2018 task1. Semeval 2019 Task 3.	ISEAR: Shame, Disgust, Joy, Anger, Sadness, Fear, Guilt. Semeval 2018: Sadness, Anger, Joy and Fear. Semeval 2019: Happy, Anger, Sad and others.	Using Google Word Embedding, Glove, and fastText to compare performance. FastText outperformed the others. Results: ISEAR: 63% SemEval 2019 task 3: 90.6% SemEval 2018 task 1: 83.6%	The model has a high level of complexity.
[37]	Deep Learning	SemEval 2019 Task 3	Happy, Sad and Angry	Presented an attention-based model. Model obtained an F1 Score of 78 %	Does not perform well for identifying the happy emotion.
[38]	Deep Learning	Tweets dataset and ROC story data	Eight emotions	Created an embedding emotion model using CNN. Accuracy:51.2%	Negate sentence were not achieved. Does not perform well for detection the Anger emotion. Accuracy was not very high Absence of the comparisons to verify the effectiveness of model.
[41]	Deep Learning	Emotion classification dataset	worry, Happiness, Sadness, Kindness, rage, Affection and Astonishment	Explored the effectiveness of LSTM deep learning method. Accuracy: 94.15%.	Used small dataset.
[42]	Deep Learning	Twitter dataset	Happy, Sad, Anger, fear, disgust, and surprise.	The factor graph model is used to detect multiple emotions or Online Social Networks, as well as to propose a multi-label learning algorithm and they obtained contextual information. Achieved a F1 score of 62.7%	Small amount of annotated text has been used.
[43]	Deep Learning	Semeval 2019 task 3	Happy, Sad and Angry	Developed a deep learning model based on Bi-LSTM networks, as well as an emotion-oriented attention network method for extracting emotion information from utterances. F1 score: 75.57%	Does not perform well for identifying the sad emotion. Limited types or classes of emotions.
[60]	Hybrid approach	Semeval 2019 task 3	Happy, Sad, and Angry	proposed a novel approach Hierarchical LSTMs for Contextual Emotion Detection (HRLCE) Accuracy: 0.779%	Misclassification were high
[14]	Lexicon approach	Collected dataset from an online booking website for hotels	anticipation, disgust, sadness, trust, fear, joy, Anger and surprise.	Extracted eight emotions using NRC Lexicon. They obtained an accuracy of 76.9 %	Deep learning techniques are recommended for improved performance.
[15]	Lexicon approach	Facebook and Twitter Messages	Anger, Fear, sadness, and joy	Identified emotion categories and Valence using emotion lexicons. NRC lexicon achieved better results that compared with other lexicons and EmoSentiNet gave the least values. Recall: 86%.	The lexicon has a limited number of terms.
[12]	Lexicon approach	Tweets	Angry, Happy, sad, and Anxious	Creates an emotion tracker. Downloads tweets, predicts overall emotions over time. Creates and displays emotion	Lake of identifying emotion interactions.

Reference No.	Approach	Dataset	Emotions	Methodology	Limitations
				time sequence graphs. Accurately Calculates the degrees of enthusiasm in a cricket match between characteristics of play and feelings toward a product Accuracy: 50.7%.	
[13]	Lexicon approach	SemEval-2007	Joy, Surprise, Disgust, Fear, Sadness and Anger	Compared the performance of ESN Lexicon, NRC Lexicon, DPM lexicon and TDPM lexicon. NRC performed better. F1-Score: 48%	Weak context information extraction
[16]	Lexicon approach	Twitter dataset	Anger, Fear, Joy and sad	An optimized framework for large datasets has been developed, and multi-label emotions have been identified. They obtained a precision of 43% and a recall of 67%	Proposed model misclassified for Twitter dataset
[59]	Hybrid approach	AIT-2018 dataset	Anger, Fear, Joy and sadness	Proposed a lexical-based model based on WordNet-Affect and EmoSenticNet with several supervised classifiers. EmoSenticNet outperforms WordNet with an accuracy of 88.23 % when using SVM.	Deep learning techniques are recommended for improved performance.
[62]	Hybrid approach	Semeval 2019 task 3	Angry, Sad, happy and others	Proposed Emodet2 model using Bert and bilstm. They obtained an F-score of 0.748	

6. CONCLUSION

This paper introduces the existing models for textual emotion detection. Firstly, we presented some primary emotion models, and then we introduced the different approaches to emotion detection such as keyword approach, lexicon approach, machine learning approach (traditional machine and deep learning), word embedding approach, and hybrid approach. In addition, we have summarized some recent works on this topic in a table that includes the features and limitations of their work. Datasets and an evaluation matrix are described. In conclusion, the traditional machine learning technique improves text classification performance mainly by improving the feature extraction scheme and classifier design. In contrast, the deep learning model enhances performance by improving the presentation learning method, model structure, and additional data and knowledge. In addition, we provide an overview of the various techniques of word embedding. Traditional, static, and contextualized word embeddings are the three principal types of word embeddings. Contextualized word embedding has made amazing progress in important NLP tasks. Contextualization based on BERT is more efficient since it can be pretrained and fine-tuned. Word embedding may be used to improve model accuracy and excel in emotion recognition. Furthermore, hybrid approaches seem to perform better than systems using just one source of information. Finally, we hope that through this paper, the reader will have a better understanding of the research done on this topic, including the datasets used, features extracted, methodologies used, and the results reported by various researchers.

REFERENCES

[1] K. Sailunaz, M. Dhaliwal, J. Rokne, and R. Alhajj, "Emotion detection from text and speech: a survey," Social Network Analysis and Mining, vol. 8, no. 1, pp. 1-26, 2018.

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz and J. G. Taylor, "Emotion recognition in human-computer interaction," IEEE Signal processing magazine, vol. 18, no. 1, pp. 32-80, 2001.

[3] F. Cavallo, F. Semeraro, L. Fiorini, G. Magyar, P. Sinčák and P. Dario, "Emotion Modelling for Social Robotics Applications: A Review," Journal of Bionic Engineering, vol. 15, no. 2, pp. 185 - 203, 2018.

[4] E. Paul, "Basic emotions," Handbook of conition and emotion, vol. 98, pp. 45-60, 19999.

[5] A. Ortony, G.L. Clore, and A. Collins, The Cognitive Structure of Emotions. Cambridge, UK: Cambridge Univ. Press, 1990.

[6] W. G. Parrott, "Emotions in social pyschology: Essential readings", Psychology press, 2001.

[7] R. Plutchik, "A general psychoevolutionary theory of emotion," in Theories of emotion, Elsevier, 1980, pp. 3-33.

[8] J. A. Russell, "A circumplex model of affect," Journal of personality and social psychology, vol. 39, no. 6, p. 1161, 1980.

[9] F. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," arXiv preprint arXiv:1103.2903, 2011.

[10] G. A. Miller, "WordNet: An electronic lexical database," 1998.

[11] H. A. Ruposh and M. M. Hoque, "A computational approach of recognizing emotion from Bengali texts.," in 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019.

[12] A. Joshi, V. Tripathi, R. Soni, P. Bhattacharyya, and M. J. Carman, "EmoGram: An Open-Source Time Sequence-Based Emotion Tracker and Its Innovative Applications," in Workshops at the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[13] F. S. Tabak and V. Evrim., "Comparison of emotion lexicons," in 2016 HONET-ICT, IEEE, 2016, pp. 154 - 158.

[14] Y. Chuttur and R. Tencamah, "Analysing and Plotting Online Customer Emotions Using a Lexicon-Based Approach," in Soft Computing and Signal Processing, Springer, 2021, pp. 181- 190.

[15] E. Kusen, G. Cascavilla, K. Figl, M. Conti, and M. Strembeck, "Identifying emotions in social media: comparison of word-emotion lexicons," in 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2017.

[16] Y. Wang, A. Pal, "Detecting emotions in social media: A constrained optimization approach," in Twenty-fourth international joint conference on artificial intelligence, 2015.

[17] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in European conference on machine learning, 1998.

[18] T. Joachims, "A statistical learning learning model of text classification for support vector machines.," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.

[19] A.I. Schein and L.H. Ungar, "Active learning for logistic regression: an evaluation," Macine learning, vol. 63, no. 3, pp. 235- 265, 2007.

[20] M. Z. Asghar, F. Subhan, M. Imran, F. M. Kundi, S. Shamshirband, A. Mosavi, P. Csiba, and A. R.

Varkonyi-Koczy,"Performance evaluation of supervised machine learning techniques for efficient detection of emotions from online content," arXiv preprint arXiv:1908.01587, 2019.

[21] J. R. Quinlan, "Induction of decision trees," in Machine learning, 1986.

[22] L. Breiman, "Random forests," in Macine learning, 2001.

[23] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, no. 1, pp. 21-27, 1967.

[24] M. Suhasini and B. Srinivasu, "Emotion detection framework for twitter data using supervised classifiers," Data Engineering and Communication Technology, pp. 565-576., 2020.

[25] Nasir, A. F. A., Nee, E. S., Choong, C. S., Ghani, A. S. A., Majeed, A. P. A., Adam, A., & Furqan, M, "Text-based emotion prediction system using machine learning approach," in IOP Conference Series: Materials Science and Engineering, 2020.

[26] F. M. Alotaibi, "Classifying text-based emotions using logistic regression," VAWKUM Transactions on Computer Sciences, vol. 7, no. 1, pp. 31 - 37, 2019.

[27] N. A. S. Winarsih, and C. Supriyanto, "Evaluation of classification methods for Indonesian text emotion detection," in 2016 International seminar on application for technology of information and communication (ISemantic), 2016.

[28] Y. Chen, "Convolutional neural network for sentence classification," University of Waterloo, 2015.

[29] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in ICML, 2011.

[30] D. P. Mandic and J. A. Chambers, Recurrent Neural Networks for Prediction: Architectures, Learning Algorithms and Stability. Chichester, U.K.: Wiley, 2001.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735 - 1780, 1997.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[33] M. Abdullah, M. Hadzikadic, and S. Shaikh, "SEDAT: sentiment and emotion detection in Arabic text using CNN-LSTM deep learning," in 2018 17th IEEE international conference on machine learning and applications (ICMLA), 2018.

[34] K. Shrivastava, S. Kumar and D. K. Jain, "An effective approach for emotion detection in multimedia text data using sequence based convolutional neural network," Multimedia Tools and Applications, vol. 78, no. 20, pp. 29607 - 29639, 2019.

[35] G. Haryadi,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, no. 6, 2019.

[36] 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, 2019.

[37] W. Ragheb, J. Az, S. Bringay and M. Servajean, "Attention-based modeling for emotion detection and classification in textual

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

conversations," arXiv preprint arXiv:1906.07020, 2019.

[38] S.-H. Park, B.-C. Bae, and Y.-G. Cheong, "Emotion recognition from text stories using an emotion embedding model," in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 2020.

[39] H. Fei, Y. Ren, and D. Ji, "Implicit objective network for emotion detection," in CCF International Conference on Natural Language Processing and Chinese Computing, 2019.

[40] U. Rashid, M. W. Iqbal, M. A. Skiandar, M. Q. Raiz, M. R. Naqvi and S. K. Shahzad, "Emotion Detection of Contextual Text using Deep learning," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 2020, pp. 1-5, doi: 10.1109/ISMSIT50672.2020.9255279.

[41] M. Karna, S. Juliet, D and R. C. Joy, "Deep learning based text emotion recognition for chatbot applications," in 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020.

[42] X. Zhang, W. Li, H. Ying, F. Li, S. Tang and S. Lu, "Emotion Detection in Online Social Networks: A Multilabel Learning Approach," in IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8133-8143, Sept. 2020, doi: 10.1109/JIOT.2020.3004376.

[43] L. Ma, L. Zhang, W. Ye, W. Hu, "PKUSE at SemEval-2019 task 3: emotion detection with emotion-oriented neural attention network," Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 287 - 291, 2019.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[45] H. Ma, J.Wang, L. Qian, and H. Lin, "HAN-ReGRU: hierarchical attention network with residual gated recurrent unit for emotion recognition in conversation," Neural Computing and Applications, vol. 33, no. 7, pp. 2685 - 2703, 2021.

[46] R.Saxena, S.Bhat, and N. Pedanekar, "EmotionX-Area66: predicting emotions in dialogues using hierarchical attention network with sequence labeling," Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, pp. 50 - 55, 2018.

[47] D. M. El-Din, "Enhancement bag-of-words model for solving the challenges of sentiment analysis.," International Journal of Advanced Computer Science and Applications, vol. 7, no. 1, 2016.

[48] S. M. H. Dadgar, M. S. Araghi, and M. M. Farahani, "A novel text mining approach based on TF-IDF and Support Vector Machine for news classification," 2016 IEEE International Conference on Engineering and Technology (ICETECH), pp. 112-116, 2016.

[49] S. G. K and S. Joseph, "Text classification by augmenting bag of words (BOW) representation with co-occurrence feature," IOSR J. Comput. Eng, vol. 16, no. 1, pp. 34-38, 2014.

[50] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[51] R. JeffreyPennington and C. Manning, "in: Proceedings of the 2014 conferenceon empirical methods in natural language processing (EMNLP2014)," Glove:

Global vectors for word representation, p. 1532–1543, 2014.

[52] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks forefficient text classification," in the 15th Confer-ence of the European Chapter of the Association for Computa-tional Linguistics, 2017.

[53] M. Zaib, Q. Z. Sheng, and W. Emma Zhang, "A short survey of pre-trained language models for conversational AI-a new age in NLP.," in Proceedings of the Australasian Computer Science Week Multiconference., 2020.

[54] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: a review of BERT-based approaches," Artificial Intelligence Review, 2021.

[55] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," arXiv preprint arXiv:1802.05365, 2018.

[56] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[57] 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, p. 4171–4186, 2018.

[58] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," Advances in neural information processing systems, vol. 32, pp. 5754–5764,, 2019.

[59] F. M. Shah, A. S. Reyadh, A. I. Shaafi, S. Ahmed and F. T. Sithil, "Emotion Detection from Tweets using AIT-2018 Dataset," 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019, pp. 575-580, doi: 10.1109/ICAEE48663.2019.8975433

[60] C. Huang, A. Trabelsi and O. R. Zaane, "Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert," arXiv preprint arXiv:1904.00132, 2019.

[61] A. F. Adoma, N. -M. Henry, W. Chen and N. Rubungo Andre, "Recognizing Emotions from Texts using a Bert-Based Approach," in 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020.

[62] H. Al-Omari, M. A. Abdullah and S. Shaikh, "Emodet2: Emotion detection in english textual dialogue using bert and bilstm models," 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 226- 232, 2020.

[63] Scherer, K. R., & Wallbott, H. G. (1994). Evidence for universality and cultural variation of differential emotion response patterning. Journal of Personality and Social Psychology, 66, 310-328.

[64] S. Buechel, U. Hahn, "Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation," Proceedings of the 11th Linguistic Annotation Workshop, pp. 1-12, 2017.

[65] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, ‘‘Dailydialog: A manually labelled multi-turn dialogue dataset,’’ 2017, arXiv:1710.03957. [Online]. Available: https://arxiv.org/abs/1710.03957.

[66] L. Ungar, and E. P. Shulman, "Modelling valence and arousal in facebook posts," in Proceedings of the 7th workshop on computational

https://digitalcommons.aaru.edu.jo/fcij/vol7/iss1/3
DOI: https://doi.org/10.54623/fue.fcij.7.1.3

approaches to subjectivity, sentiment and social media analysis, 2016.

[67] S. Mohammad and F. B-Marquez, "WASSA-2017 shared task on emotion intensity," arXiv preprint arXiv:1708.03700, 2017.

[68] M. Hasan, E. Rundensteiner, and E. Agu, "Automatic emotion detection in text streams by analyzing Twitter data," International Journal of Data Science and Analytics, vol. 7, no. 1, pp. 35-51, 2019.

[69] Rosenthal, Sara, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," arXiv preprint arXiv:1912.00741, 2019.