Understanding the performance of AI algorithms in Text-Based Emotion Detection for Conversational Agents

Sheetal D. Kusal, 0000-0002-9830-6619

Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, 412115, India, sheetal.kusal.phd2020@sitpune.edu.in

Dr Shruti G. Patil, 0000-0002-4903-1540

Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, 412115, India, shruti.patil@sitpune.edu.in

Dr Jyoti Choudrie, 0000-0003-0881-5477

University of Hertfordshire, Hatfield, Hertfordshire, United Kingdom. j.choudrie@herts.ac.uk

Dr Ketan V. Kotecha, 0000-0003-2653-3780

Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, 412115, India, director@sitpune.edu.in

Current industry trends demand automation in every aspect, where machines could replace humans. Recent advancements in conversational agents have grabbed a lot of attention from industries, markets, and businesses. Building conversational agents that exhibit human communication characteristics is a need in today's marketplace. Thus, by accumulating emotions, we can build emotionally-aware conversational agents. Emotion detection in text-based dialogues has turned into a pivotal component of conversational agents, enhancing their ability to understand and respond to users' emotional states. This paper extensively compares various AI - techniques adapted to text-based emotion detection for conversational agents. This study covers a wide range of methods ranging from machine learning models to cutting-edge pre-trained models as well as deep learning models. The authors evaluate the performance of these techniques on the benchmark unbalanced topical chat and empathetic dialogue, balanced datasets. This paper offers an overview of the practical implications of emotion detection techniques in conversational systems and their impact on user response. The outcomes of this paper contribute to the ongoing development of empathetic conversational agents, emphasizing natural human-machine interactions.

CCS CONCEPTS • Computing Methodologies • Applied Computing • Human-Centered Computing

Additional Keywords and Phrases: Artificial Intelligence, Natural Language Processing, Machine Learning, Deep Learning, Text-based emotion detection, and pre-trained models.

1 Introduction

Artificial Intelligence (AI) has enabled machines to mimic human behaviour. Many applications like speech recognition, machine translation, sentiment analysis, and conversational agents are considerably changing human-machine interactions with the help of AI and bringing machines closer to human behaviour. One of the applications of AI, conversational agents, has the ability to understand and produce responses in a natural language like human beings. In current years, CAs are gaining popularity in various industries and areas due to their divergent characteristics, namely simple Interface, 24/7 availability, prompt response, engaging, cost-effective Omni channel, and talk to people, systems, and things. Businesses [1], customer care [2], education [3], entertainment [4], and healthcare [5] are a few areas to mention where conversational agents have been used extensively. Similarly, Emotions are crucial in human-computer interaction as they help computers comprehend human behaviour and become more responsive [6]. Moreover, social media channels allow people to communicate their thoughts. So, communications on social channels open to research social trends by studying people's emotions, tracking consumer feedback to analyze and create business plans, assisting consumers in decision-making tasks and many more. In business, especially in marketing, emotion recognition is crucial to increasing brand awareness. From social media sites, it is possible to gain customer attention. Studies have shown that there is a correlation between high emotional performance and social media sources. A good link with the brand is created when advertisements from brands include humour and narration to improve brand visibility and social media participation. Emotion detection can also be utilized in customer service to increase the effectiveness of call center representatives. AI-based customer-representative matching can be aided by examining user emotions and word choices. If users' emotions and word choices are matched with the appropriate representative, this will result in a call success ratio with customers. Consequently, technology assisted by emotions will be significant in decision-making to the broader range of application domains, including public monitoring, finance, education, management and marketing, user interaction, healthcare, etc.

Current conversational agents are lacking in understanding the emotions and sentiments of the user. Most conversational agents use keyword-based methods, which cause them to respond with specific answers, which leads to the failure of tone or mood comprehension of the user. Building conversational agents replicating human

conversational characteristics and behavior is necessary to comprehend the user's feelings, thoughts, emotions, and mood. Employing sentimental and emotional analysis in the conversational agents with the help of advanced NLP and AI Systems is needed. The amalgamation of text-based conversational agents with emotion detection can make conversational agents to understand users' emotions, tone or mood. Figure 1 shows that analyzing the user's emotions will help conversational agents to reply emotionally and correctly, understanding the situations, context, and perspective of the user [7]. Existing research work can detect the emotions in the user input. However, there is a lack of completeness regarding human emotional analysis, which affects response generation in conversational agents.

Here, the authors propose adding an emotion detection module in the conversational agent to identify the emotions in user input. This emotion analysis will also help the conversational agent to respond emotionally. The objective of the work is to annotate the emotion label of user input in a dialogue between the conversational agent and the user. Text-based emotion detection has been widely implemented with different approaches. Popularly utilized techniques are based on keyword [8], rule [9], machine learning [10], and deep learning-based [11]. In this work, to obtain the fine-grained emotions of user input, the authors have implemented these techniques on the user input. The authors evaluated these algorithms on two conversational datasets – the topical chat dataset and the empathetic dialogue dataset. We compare the results of both datasets with respect to deep learning, machine learning and the pretrained BERT model.

The following is a summary of our contributions:

1. Evaluation and analysis of the following machine learning algorithms – Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Decision Tree (DT), Support Vector Machine (SVM), and Ensemble voting method on both datasets to classify the emotion labels of user conversation.
2. Evaluation and analysis of the deep learning techniques: Long-Short Term Memory (LSTM), Bi-directional Long-Short Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU) and Bi-directional Gated Recurrent Unit (Bi-GRU) on both datasets to classify the emotion labels of user conversation.
3. Evaluation and analysis of the transformer-based BERT model for emotion detection from conversational dialogues.
4. Comparison analysis of machine learning, deep learning and pre-trained models for emotion detection in conversations.

The remainder of the paper describes how the paper is organized. Some of the most current work in this field is described in Section II. The techniques and models utilized in this study are described in depth in Section III. Results and discussions of the implemented algorithms are described in Section IV, which is followed by a possible conclusion and future scope in Section V.

2 Related work –

Making conversational agents more empathetic requires the addition of emotions. Emotion analysis [12] is targeted to pull out fine-grained emotions from text, like anger and happiness and analyze them. Emotion extraction is considered as a classification task. The various approaches used for analyzing emotions from the text are deep learning, machine learning, keyword, and rule-based approaches. [13] discussed recent advancements with different techniques and their advantages and disadvantages in building the chatbot. The first emotionally aware chatbot was PARRY, developed by [14] in 1972. This rule-based system was designed on assumptions and was able to generate emotional responses by considering the change in the weights of user utterances. [15] developed a conversational approach that can detect and adapt the affective state of the user in dialogues. Naive Bayes, k-Nearest Neighbor, and Centroid Document, three conventional classifiers, have been used to assign sentiment classes to text samples in the process of detecting affective states through sentiment classification. Sentence-level sentiment classification systems developed by [16] using a rule-based approach addressed the polarity shift problem. [17] designed a system to detect the emotions from human-chatbot interaction using BERT, Bi-LSTM-based deep learning model and logistic regression, SVM, and random forest in machine learning models. The second-order Markov model was developed for emotion prediction in industrial chatbot [18]. [19] designed a pedagogical agent that can understand the affective state of the user. The authors used reinforcement learning with LSTM to recognize the sentiments of the user. In [20], transformers are used to predict the emotion of response based on context. [21] presented a deep learning-based chatbot that could learn long-term dependencies in conversations using LSTM, a variant of RNN. Expressing the emotions in response generation was achieved by [22] by utilizing the seq2seq model. [23], [24] implemented deep learning-based emotion detection systems using LSTM and GRU for conversational datasets. [25] developed shared GRU self-attention based a conversational emotion detection system for emotion based dialogues. [26] proposed a model for affect identification in conversational text using RNN based deep learning model. [27] proposed a model for conversational text data using a graph convolutional neural network with dependency syntactic analysis. [28] used graph attention neural networks for sentiment and context-aware conversational data with three various datasets. The authors can infer from the literature that with the development of technology, researchers are favoring deep learning techniques for conversational emotion recognition. Transformers, encoder-decoder models, and graph neural networks receive special attention in deep learning.

Table 1: Details of literature reviewed

Ref.	Dataset	Text Pre-processing	Feature extraction methods	Classification Methods	Performance metrics	Challenges	Application
[17]	Indian chat room	Punctuation Removal, Expanding Contractions, Tokenization, Normalization	Word embeddings	Machine learning, Bi-LSTM	0.77 (F-score)	Accuracy for the label happy is smaller than the angry, sad.	English user-chatbot interaction.
[29]	(FAQs) from the internet in the e-business domain	Tokenization, stop words removal, stemming	Word2doc	AIML And LS analysis	0.97 (Precision)	Only designed for FAQs. General questions are not considered.	E-business chatbot
[25]	Topical Chat dataset	Not mentioned	Encoder	Shared GRU encoder with self-attention	0.23 (Fre) / 0.19 (Rare) (F-score)	Due to the scarcity of labels in the dataset, models may have been trained to detect the most common words. Unable to elicit responses to specific emotion labels such as angry, sad, fearful, and disgusted due to a lower frequency of labels.	User Interaction
[30]	Chinese Valence-Arousal Words	Repetition removal, stop words removal, sentences with 20 words retained.	one-hot encoding,	Bi-LSTM	0.6181 (F-score)	Not achieved state-of-the-art performance. The system addressed FAQ-type dialogues. Need to address intent detection.	Customer service in healthcare
[23]	NLPCC dataset	Not mentioned	Pre-trained word embeddings	Bi-LSTM	0.623 (Accuracy)	Unable to accurately extract conversational emotional aspects; Syntactic, lexical, grammatical, and information associated with emotional factors are not taken into account	Open Domain
[24]	NLPCC dataset	Removal of punctuation and symbols	Encoder	GRU	0.9658 (Emotion Accuracy)	Incapable of producing intriguing and informative responses. A model cannot replicate the empathy factor in human communication.	Open Domain
[37]	Manually built	Not mentioned	Not mentioned	Machine Learning approach, Classifiers SVM, decision tree, NB, and the tree bagger	0.769 (Accuracy)	Ignoring contextual information resulted in misclassification. No semantic representation in the model.	User Interaction

[26]	SemEval-2019	Tokenization, Spell correction, word normalization, word segmentation, lowercasing	Word2Vec	Pretrained language models ULMFiT, BERT, OpenAI's GPT	76.86 (F-Score)	Accuracy for the happy class is smaller than angry, sad classes.	Open Domain
[19]	Short posts in Web forums and Wikis,	Not mentioned	Word2vec	Deep learning-based – LSTM	Not mentioned	Needs a lot of data for its practical use. Needs protection of personal information in a highly interactive tool.	User Interaction
[27]	IEMOCAP and MELD	Not mentioned	Pre-trained word embeddings	CNN, Graph Convolutional Neural Network	63.5 (Accuracy) 60.9 (Accuracy)	class imbalance in the MELD dataset. Need to improve the classification accuracy of the model on the minority class.	User conversations
[28]	Daily Dialog, MELD, EmoryNLP	Not mentioned	GloVe embeddings	GCN, DialgogueRNN, Sentic graph attention.	(F-Score) 54.45 59.19 36.59	An imbalanced number of samples in the dataset makes the model weak in performance.	User conversations

Table 2: Overview of Artificial Intelligence (AI) techniques used in text-based emotion detection.

References	Approach	Description	Algorithms used	Advantages	Disadvantages
[31] [32] [33] [34]	Machine Learning	Based on the ability to learn from experience and develop	Logistic regression, Naïve Bayes, Support Vector Machine, Random Forest, Decision Tree, Artificial Neural Network, K-Nearest Neighbour	- Widely implemented - Better detection results.	- Not robust. - Not explicitly extract semantic information.
[35] [36] [37] [27]	Deep Learning	Based on learning without human intervention from data	Convolutional Neural Network, Recurrent Neural Network and its variants – LSTM, Bi-LSTM, GRU, Bi-GRU	- More robust - Extract deeply hidden details.	- Need extensive data to train the system.
[35] [38] [39], [40]	Pre-trained Language Models	Based on machine learning models trained on large amounts of text data and can be fine-tuned for specific task	Transformer-based models – BERT, RoBERTa, DistillBERT, GPT	- Requiring less data than designing a model from scratch. - Improved performance - Reduced training time	- High computational cost - Generalization - require a lot of storage, complex deployment, and maintenance

Table 1 shows the different articles reviewed by the authors with different perspectives, such as text pre-processing techniques, feature extraction methods preferred, classification techniques used, performance metrics and application areas. The authors also presented a summary of Artificial techniques used in text-based emotion detection in Table 2. Table 2 compares techniques based on technology, description, algorithms used, merits and demerits.

3 Emotion detection model in Conversational Agents -

In this section, a text-based conversational agent system is presented. The architecture of a conversational agent consists of three subsections where input is sequentially processed in steps. 1) Natural Language Processing 2) Natural Language Understanding 3) Natural Language Generation. In NLP, input messages from the user are processed. Figure 1 depicts different examples of dialogues in various contexts and situations with responses to conversational agents. Initially, input message text cleaning and normalization steps are performed. Then, vectorization of the input message is performed in feature analysis. Afterwards, emotion analysis, one of the tasks in NLU, is done with machine learning, deep learning and pretrained model BERT. Emotion analysis helps in understanding the user's emotions to respond accordingly. Input messages from users with recognized emotions are passed to NLG. In NLG, a response to a user message can be generated with the help of generative-based methods. The generated response will be returned to the user. The architecture of the conversational agent is shown in Figure 2. In this paper, the authors have focused on the emotion analysis of conversations in conversational agents.

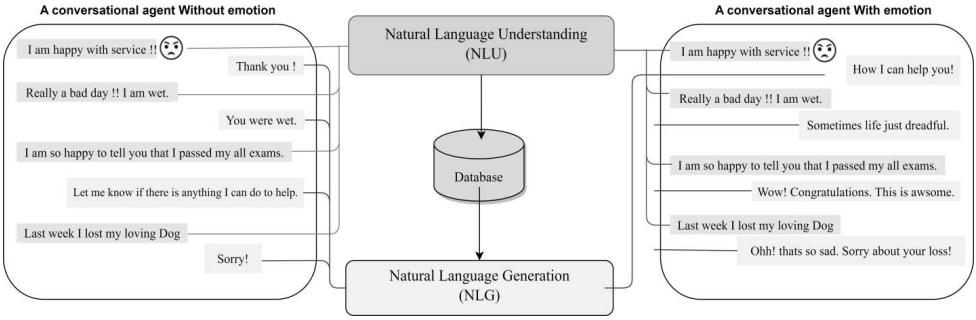


Figure 1: The response generated by conversational agents without emotion and with emotion.

3.1 Datasets –

Very few datasets have been developed for conversational agents. In most literature, researchers have curated datasets according to the research needs. Some researchers have built emotions annotated datasets for conversational agents. These datasets play an essential role in understanding the emotions and context of the user sentences and replying accordingly. [31] [32] surveyed conversational datasets with annotated emotions. The authors decided to go with the Topical chat dataset [33] and the Empathetic dataset [34] by analyzing different datasets from the literature survey. Table 1 presents the statistics of datasets with training and testing sets and no of emotions included. The topical chat dataset has a total of 11,319 conversations divided into training and testing sets. This dataset is annotated for eight emotions: Disgusted, Angry, Fearful, Happy, Sad, Surprised, Neutral and Curious to dive deeper. While the EMPATHETIC DIALOGUES dataset has 25000 conversations. It is also divided into training and testing sets with 32 labels of emotions - Surprised, Angry, Excited, Annoyed, Sad, Proud, grateful, Lonely, scared, alone, Guilty, Afraid, Hopeful, Anticipating, Disgusted, Impressed, Nostalgic, Furious, Confident, Anxious, Joyful, Sentimental, Prepared, Embarrassed, Jealous, Content, Ashamed, Devastated, Caring, Trusting, Faithful, Apprehensive. The topical chat dataset has an unbalanced distribution of emotion labels, and EMPATHETIC DIALOGUES has an approximately balanced distribution of emotion labels. Authors preferred Topical chat and EMPATHETIC DIALOGUES datasets to evaluate different models on balanced and unbalanced datasets.

Table 2: Dataset Details.

Name of Dataset	Training	Testing	No. of Emotions included
Topical-Chat Dataset	9058	1130/1131 (Freq/Rare)	08
EMPATHETIC DIALOGUES	19533	5317	32

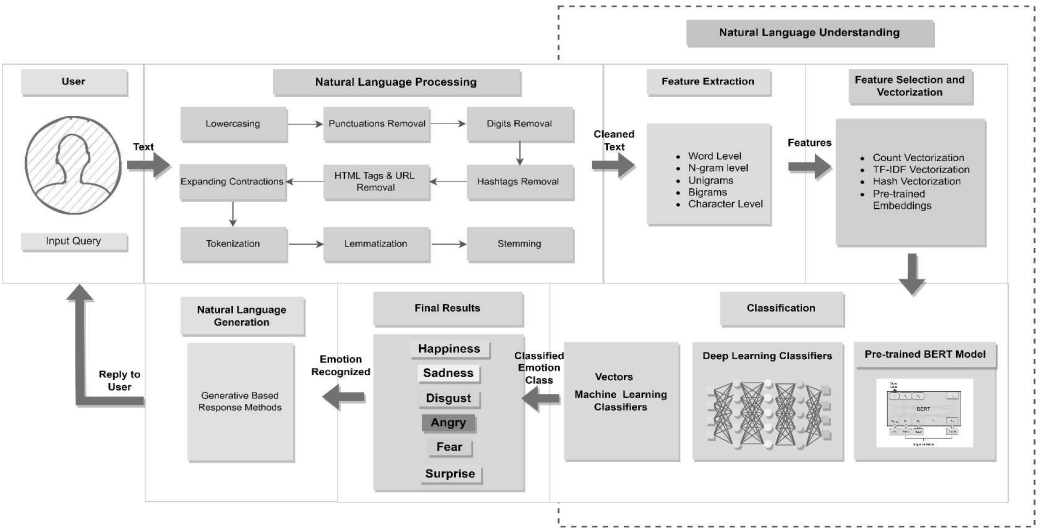


Figure 2: Working framework of a conversational agent with different techniques.

3.2 Pre-processing of Data –

The authors pre-processed the datasets in this module with natural language processing steps. NLP steps are shown in Figure 2. Basically, NLP steps include data cleaning, normalization, and vectorization.

1. Data Cleaning – In the data cleaning, input messages are processed through the following steps -
   - Lowercasing – Input messages from the user are transformed to lowercase so that all data can be processed in the same case format.
   - Punctuation and digits removal – Punctuation marks are removed to treat all data equally. Similarly, since numerals don't provide vital information, they are eliminated from input messages.
   - Removing Stop words – Stop words generally refer to common words from the language. These words don't really add anything to the meaning of a sentence; thus, they can be safely ignored without changing the significance of the sentence. Stop words are common, general words. When we do natural language processing, we don't worry about the general words present everywhere. We need the actual unique words from which we can infer information. Common words don't add a lot of value. So, we have removed stop words from user input.
   - Hashtag, HTML tag, URL and white space removal - Generally, hashtags, HTML tags, URLs, and white space are considered as noise in the NLP tasks. So, we have removed noise from user input.
   - Expanding contractions - Because the user text has so many contractions, we must expand it in order to eliminate them.
2. Normalization - Lemmatization and Stemming contain abbreviating a word to its origin form. However, there is a difference between lemmatization and the stemming process. Stemming is a method used to reduce a word to its stem or root format by removing suffixes or prefixes. In comparison, lemmatization reduces words to a standardized form. It uses a dictionary to trace word variations back to their original forms. So, we have used lemmatization in our work.
3. Tokenization - Tokenization is the process of breaking up a long block of text into tokens. Words, letters, or sub-words can all be used as tokens. Tokenization splits characters, words, and phrases. Essentially, it is breaking up text or strings into tokens representing words. The authors have used the simplest white space tokenization method in which text is tokenized based on whitespace within a string as a delimiter of words.

3.3 Feature Extraction and Selection –

To transform the text into a vector or matrix of features, feature extraction techniques are necessitated. Data feature extraction is the common name for this transformation task. This feature extraction process is called vectorization. In deep learning methods, word embeddings are the same as vectorization. The authors have evaluated different vectorization methods based on word level. In machine learning methods, we have used three vectorization techniques. Although in Figure 2, various vectorization techniques have been shown, the authors have used count vectorization, TF-IDF vectorization, and hashing vectorization methods. In Deep

Learning methods, one hot encoding has been preferred. These methods convert given text tokens into vectors or matrices of features. In the pre-trained BERT model, the BERT tokenizer is used to split the text into BERT tokens and transform them into vectors.

3.4 Classification –

After feature extraction, i.e., generating vectors or word embeddings, classifying emotions of the input text is an essential stage in conversational agents to comprehend the situation or context and then respond accordingly. The most popular methods for recognizing text-based emotions are those that rely on keywords, rules, machine learning, and deep learning. The two categorization methods that the majority of researchers in the area of text-based emotion detection use are deep learning and machine learning. Figure 2 shows the classifiers used in the proposed architecture. The authors have evaluated machine learning, deep learning approaches, and the pretrained BERT model in the paper.

3.4.1 Classification using Machine Learning –

In text-based classification, Machine learning classifiers are expansively and significantly used. Due to the fact that they use datasets with annotations or labels, these classifiers are data-driven [35]. Machine learning models learn from experiences. Machine learning models are trained on large annotated datasets. These supervised machine learning classifiers contain inputs and desired output labels. The authors have evaluated the conversational datasets on Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Decision Trees (DT), Support Vector Machine (SVM), and Voting Classifier (Ensemble Approach) in machine learning classifiers. The authors have chosen these classifiers, which are the most suitable for text-based emotion classification. Detecting emotions in dialogues is a multi-class classification problem. Therefore, machine learning algorithms that give good results on multi-class classification problems in text processing are chosen by the authors. Generally, a multi-class classification problem is modelled with multinoulli probability distribution with each sample. Some machine learning binary classifiers have trained multiple binary classification models for each class compared to all other classes, i.e., the one vs Rest strategy for multi-class classification. Therefore, the authors have selected SVM, LR, DT and MNB. Voter classifier, A machine learning model known as the ensemble technique, learns by being trained on a variety of different models, and it predicts class based on the highest possibility of being the output. To predict the class based on the majority of votes, it simply averages the results of each classification model that was presented into the voting classifier.

1. Logistic Regression (LR) – A fundamental and well-liked approach for categorization and predictive analysis is logistic regression. Logistic regression is a kind of statistical machine-learning model. It is often referred to as the logit model. Logistic regression determines the probability of an event happening based on a collection of independent variables. Given that the outcome is a probability, the dependent variable's range is 0 to 1. The logit formula used in logistic regression is the probability of success divided by the probability of failure to change the odds. The following formulas serve as representations for this logistic function, referred to as the natural logarithm of odds. It is defined as

$$\sigma(t) = \frac{e^t}{e^t + 1}$$

In the logistic equation, t is the input variable. For the multi-class classification model, multinomial logistic regression has a dependent variable and three or more possible outcomes. Multi-class classification is solved as a binary classification problem using the one vs all method, in that a class is denoted by one, and the rest of the classes become zero.

2. Multinomial Naïve Bayes (MNB) – For the study of categorical text data, Multinomial Naive Bayes represents one of the most often used supervised learning classifications. Multinomial Naive Bayes is a probabilistic learning method based on the Bayes theorem and predicts the categories of a text. For a given sample, it calculates the probabilities of each category and then returns the category with the most significant likelihood. An important consideration is that the feature categorized by the Naive Bayes differs from any other feature. Using knowledge about the circumstances of an occurrence, the Bayes theorem determines the likelihood that it will occur. To calculate the probability of class A, predictor B is provided. Its principal equation is as follows:

$$P\left(\frac{A}{B}\right) = P(A) * P\left(\frac{B}{A}\right) * P(B)$$

3. Decision Trees (DT) – A decision tree is the easiest and most popular algorithm to understand and interpret. It can solve classification and regression problems. A decision tree creates a training model by learning simple decision rules. These decision rules are used to predict the class or category. Predicting the label of a class starts from the root of the tree. Values of the root are compared with attributes of the class, and based on the comparison results, the corresponding branch is taken and jumped to the next node in the tree. The decision tree follows the sum of product (SoP) representation. The key consideration is to find which attributes to consider at the root and at internal levels for decision rules in decision tree implementation. This is known as attribute selection and is performed by entropy or information gain.
4. Support Vector Machine (SVM) – It is another simple machine learning algorithm. It is a highly preferred algorithm as it produces significant accuracy with less computational power. It is used for both classification and regression tasks. The target of the support vector machine algorithm is to locate a hyperplane, or decision border, in an N-dimensional space; that is, N is the number of features which distinctly categorize the data labels. The maximum divergence between data labels from both

classes at this decision boundary—which is the SVM's objective—is the maximum margin. By adding some support by raising the margin distance, future data points can be labelled with more certainty. SVMs are fundamentally two-class classifiers. One-versus-rest or ``one-versus-all'' is the most preferred technique for multi-class classification, and choose the class which classifies with the greatest margin.

5. Voting Classifier (Ensemble Approach) - An ensemble is a hybrid model that combines many classifiers with weak performance to create a classifier with improved performance. Here, each classifier casts a vote, and the majority vote on the prediction label represents the outcome. Compared to a base classifier or a single classifier, ensembles offer better accuracy. Ensemble methods are capable of parallelization by allocating each base learner to a different device. The ensemble technique, a meta-algorithm, combines several machine learners into a sole predictive model to improve performance. An approach called stacking can improve predictions, a tactic called boosting can lessen bias, and ensemble approaches can minimize variance using a bagging technique. A voting classifier is a specialized machine learning estimator that builds several base classifiers or estimators and then generates predictions by averaging their output. Voting can be combined with the aggregating criterion for each predictor output.

3.4.2 Classification using Deep Learning –

Deep learning classifiers are categorized as unsupervised, semi-supervised, supervised or learning. Deep learning learns and extracts features from the data or text automatically. RNNs have been used for sequential or temporal data. For sequential input, such as speech, text, etc., RNN performs better. Since variants of RNN can take into account long-term relationships between significant events over an infinite length, the authors have used supervised deep learning methods - LSTM (Long Short-Term Memory), Bi-LSTM (Bi-directional Long Short-Term Memory), GRU (Gated Recurrent Unit) and Bi-GRU (Bi-directional Gated Recurrent Unit). These methods have been evaluated by the authors on conversational datasets.

1. LSTM (Long Short-Term Memory) – LSTM is a special Recurrent neural network (RNN). It has the ability to learn long-term dependencies. It is also able to address the vanishing gradient problem. LSTMs have a chain of modules of different structures instead of a single neural network. LSTM consists of three structures or gates which help to select how much information is allowed in each node state. i.e., what information to throw away and what information to retain for the next node. Cell state is a kind of conveyor belt that runs straight down the entire chain with linear interactions. Forget gate uses the sigmoid function, which decides what information to throw away from the previous time stamp. The candidate gate uses the Tanh function that determines what information to store with the help of input. Output state depends on cell state. Equations 1, 2,3 and 4 depict the mathematical formula for the forget gate, candidate gate, and output gate.

Forget gate →	$ft = \sigma(wf[ht-1, xt]) + bf$	(1)
Candidate gate →	$Ct = \tanh(wc[ht-1, xt]) + bc$	(2)
Output gate →	$Ot = \sigma(w0[ht-1, xt]) + b0$	(3)
Hidden output →	$Ht = Ot * \tanh(Ct)$	(4)

Notations as follows -

wf = weight for forget gate
$ht-1$ = output from the previous timestamp
xt = new input
bf = bias for forget gate
bc = bias for candidate gate
wc = weight for candidate gate
Ot = output state
Ht = new hidden state

2. Bi-LSTM (Bi-directional Long Short-Term Memory) - An adaptation of the LSTM is the bidirectional LSTM network. The Bi-LSTM has dual hidden layers. The first hidden layer processes the input sequence forwardly, whereas the second hidden layer is backwardly processed. The output layer fuses these hidden layers, giving it access to the prior and succeeding context of each point. Both LSTM and its bidirectional variants have been demonstrated to be of great use. They might discover when and how to forget specific facts, as well as when and why not to use particular architectural entrances. The benefits of a Bi LSTM network include a quicker learning rate and better performance.
3. GRU (Gated Recurrent Unit) – The LSTM architecture is simplified in the GRU architecture. While an LSTM has one gate and internal memory, a GRU only has two gates. GRU uses an update and resets approach to get over the vanishing gradient issue. The model uses the update gate to determine how much historical data from earlier time stages should be used in subsequent steps. The network's long-term memory is the update gate. The amount of data that is discarded is decided by the reset gate. The reset gate is accountable for the short-term memory of the network, i.e., the hidden state. Equations 5 and 6 depict the mathematical formula for the update gate and reset gate. Equation 7 represents the mathematical function with the update and candidate gate.

Update gate →	$ut = \sigma(wu[xt] + Uu.ht-1 + bu)$	(5)
Reset gate →	$ht = tanh(w[xt] + Uht-1 + b)$	(6)
Function for GRU –	$f(ht-1, xt) = ut * ht + (1-ut) * ht-1$	(7)

4. Bi-GRU (Bi-directional Gated Recurrent Unit) - Bi-GRU is a variant of the GRU network. A paradigm for processing sequences that uses two GRUs is called Bi-GRU, one processing the information forward and the other in the backward direction. The input gate and forget gates are present in the bidirectional recurrent neural network of Bi-GRU.

3.4.3 Classification using Pre-trained BERT model –

The pre-trained BERT model is a transformer deep learning model-based attention process. The BERT is the acronym for Bidirectional Encoder Representations from Transformers. By equally considering left and right contexts, it gains a thorough understanding of texts. This model is used to solve NLP issues and train general-purpose language models on large datasets. The BERT model comes into two sizes: BERT-large and BERT-base. BERT- 24 Transformer blocks with a concealed size of 1024 and 16 self-attention heads make up BERT-large. BERT-base includes 12 Transformer blocks with a hidden size of 768 and 12 self-attention heads. BERT trained on the BooksCorpus and English Wikipedia datasets. Therefore, the authors evaluated the pre-trained BERT model on both dialogues' datasets.

4 Results and Discussion –

In the experiments, the authors used Python with the Jupyter framework. The experiments were run on a computer with 12th Gen Intel(R) Core i7 processor at 2.10 GHz with 16GB RAM running the Linux subsystem. The authors have evaluated the proposed systems using different performance measures. The authors evaluated balanced and unbalanced datasets for F1-score, Accuracy, classification reports, confusion matrix, ROC curves, Precision, and Recall so that emotion class-wise results will be evaluated. Accuracy, which is the direct resultant of the confusion matrix, is one of the most frequently utilized metrics in multi-class classification. The accuracy is determined by adding the True Negative and True Positive observations. Generally, accuracy works well if data is balanced as it does not take into account data or sample distribution.

- True Positives – True positives are those values that the model is trying to identify, predictions of the model are also correct, and the actual values are positive.
- True Negatives – True negatives are those values that the model is not trying to identify, predictions are correct, and the actual values are negative.

In the numerator and adding up all the confusion matrix entries in the denominator, i.e., how many positives and negatives observations are correctly classified by a model. Accuracy is the percentage of all correctly classified observations.

To validate the different models in deep learning and machine learning techniques, the authors have a test set of 5317 utilized for the unbalanced dataset, whereas a test set of 2261 for the balanced dataset. The accuracy values in percentage (%) have been computed for all machine learning algorithms using confusion matrices shown in Figure 3, using count, TF-IDF and Hashing vectorization for the topical chat dataset (unbalanced) and Empathetic Dialogues dataset (balanced datasets), respectively. The results for different machine learning models - Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Decision Trees (DT), Support Vector Machine (SVM), Voting Classifier (Ensemble Approach) with topical chat dataset and Empathetic Dialogues dataset with vectorization methods are presented in table 3. From Table 3, the authors can conclude that SVM with TF-IDF has shown the highest accuracy with 75 %. In contrast, MNB and DT with hashing vectorizer have shown the lowest accuracy with 45% for the unbalanced topical chat dataset. DT with count vectorization showed an accuracy of 93% for the balanced, empathetic dialogues dataset, whereas MNB with hashing vectorizer showed the lowest accuracy of 38% in Table 3. In Vectorizer analysis, the count vectorizer showed high performance with all machine learning algorithms compared to TF-IDF and hashing vectorizer.

Table 3: Results of different machine learning algorithms for Unbalanced (Topical-Chat Dataset) and Balanced datasets (Empathetic Dialogues)

Machine Learning Methods	Dataset					
	Unbalanced (Topical-Chat Dataset) Accuracy (%)			Balanced (EMPATHETIC DIALOGUES) Accuracy (%)		
	Count Vector	TF-IDF	Hashing	Count Vector	TF-IDF	Hashing
Logistic Regression (LR)	49	50	50	45	65	45
Multinomial Naïve Bayes (MNB)	48	45	43	44	41	28
Decision Tree (DT)	45	45	45	33	31	31
Support Vector Machine (SVM)	56	49	49	79	44	30
Ensemble Method (Voting Classifier)	60	44	43	50	45	34

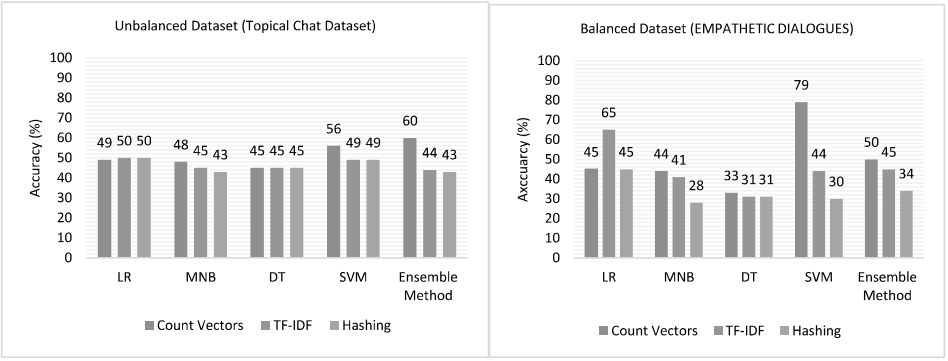
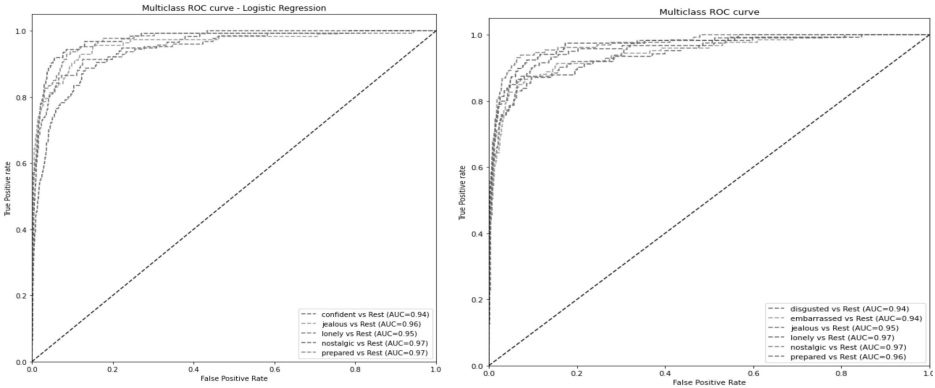


Figure 3: Accuracy Score comparison using various vectorization methods for machine learning techniques on the Unbalanced (Topical-Chat Dataset) and Balanced dataset (Empathetic Dialogues)

A graph called the Receiver Operating Characteristic Curve (ROC curve) demonstrates the reciprocity between the TPR (True Positive Rate) and the FPR (False Positive Rate). It indicates how well your model ranks the predictions. Generally, ROC curves should not be used when your data is heavily imbalanced. These are preferred in the evaluation of balanced datasets. The authors have shown ROC curves on Empathetic Dialogues (balanced dataset). The authors have shown the ROC curve for the top five emotions in Logistic regression and Multinomial Naive Bayes using count, TF-IDF and hashing vectorization in Figure 4. From ROC curves, in count vectorization, LR and MNB models have remained dominant at predicting the jealous, lonely, and nostalgic emotion labels. In TF-IDF vectorization, LR and MNB models have remained prevalent in predicting the jealous, lonely, nostalgic and prepared emotion labels. Whereas in hashing vectorization, LR and MNB models have remained dominant at predicting the lonely, nostalgic, terrified, and prepared emotion labels. The authors can infer that both models are prevalent in identifying lonely and nostalgic emotion labels. The accuracy values in percentage (%) for deep learning models computed have been shown in Figure 5 for the balanced and unbalanced datasets. The results for different deep learning models – LSTM, Bi-LSTM, GRU, and Bi-GRU are also evaluated based on training accuracy and validation accuracy on the topical chat dataset and Empathetic Dialogues dataset shown in Tables 4 and 5.



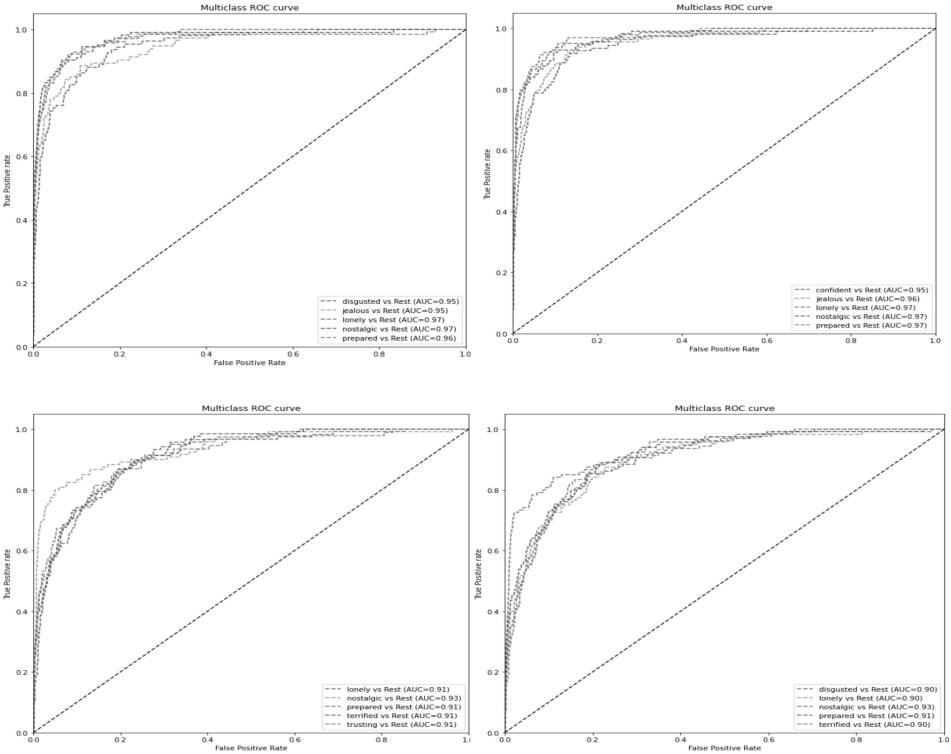


Figure 4: ROC AUC curve for top 5 emotions in Logistic regression and Multinomial Naïve Bayes using count vectorization, TF-IDF vectorization, and hashing vectorization on Empathetic Dialogues (balanced dataset).

Hyperparameters used for deep learning models are shown in Table 4. The same hyperparameters are used to evaluate the Empathetic Dialogues dataset, except the embedding feature vectors used are 80. The metric combination of training loss and validation loss over time is one of the most used. The training loss shows how well it matches the training data, while the validation loss shows how accurately the model fits the test data. Figure 6 and 7 shows the accuracy curves and loss curves for various deep-learning models on balanced and unbalanced datasets. Different colors are used to represent the different models in curves, and the solid line represents training accuracy and loss. In contrast, dotted lines represent validation accuracy and loss in Figures 6 and 7. In deep learning algorithms, Bi-GRU showed the highest training accuracy of 61%, whereas LSTM and Bi-GRU have shown the highest validation accuracy on the balanced Empathetic Dialogues Dataset. So, the authors concluded that Bi-GRU had shown the best performance for the unbalanced dataset. GRU has demonstrated the highest training accuracy of 69%, whereas LSTM and Bi-LSTM showed the highest validation accuracy on the balanced, empathetic dialogues dataset. Therefore, GRU has demonstrated the best performance for the balanced dataset. The authors also observed that there was an increase in the training accuracies, but validation accuracy was not increasing. So, they have mentioned the best average training and validation accuracies for both unbalanced and balanced datasets.

Table 4: Results of different deep learning algorithms for Unbalanced (Topical-Chat Dataset) and Balanced datasets (Empathetic Dialogues)

Deep Learning Methods	Dataset			
	Unbalanced (Topical-Chat Dataset) Hyperparameters Used – Vocab_size = 5000 Embedding_feature_vector = 200 Epochs = 50 Activation function = SoftMax Loss function = Categorical entropy Optimizer = Adam optimizer Vectorization method = One Hot Encoding		Balanced (EMPATHETIC DIALOGUES) Hyperparameters Used – Vocab_size = 5000 Embedding_feature_vector =80 Epochs = 50 Activation function = SoftMax Loss function = Sparse categorical entropy Optimizer = Adam optimizer Vectorization method = One Hot Encoding	
	Accuracy (%)	Validation Accuracy (%)	Accuracy (%)	Validation Accuracy (%)
LSTM	60	45	58	40
Bi-LSTM	68	45	57	39
GRU	69	44	54	39
Bi-GRU	67	44	61	40

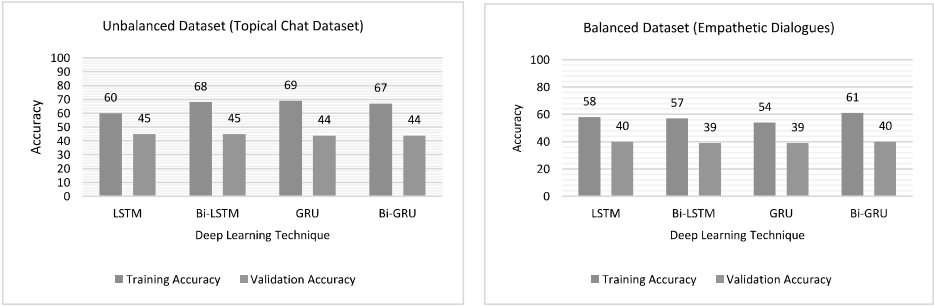


Figure 5: Accuracy Score comparison of deep learning techniques on the Unbalanced (Topical-Chat Dataset) and Balanced dataset (Empathetic Dialogue)

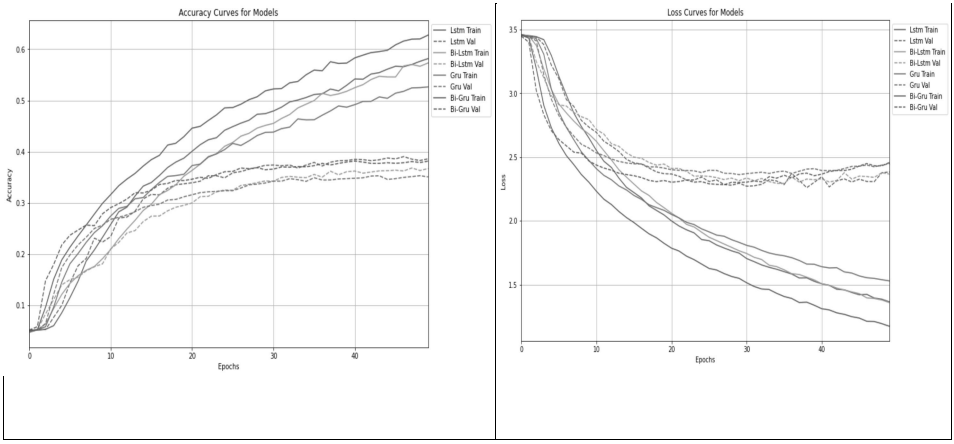


Figure 6: Accuracy curves and loss curves for the Unbalanced (Topical-Chat Dataset).

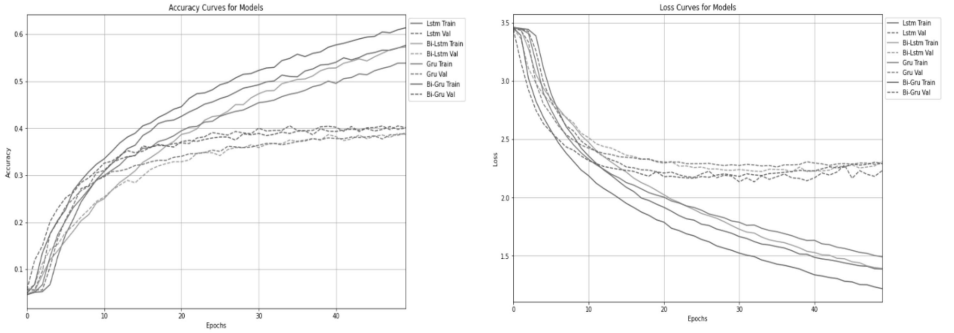


Figure 7: Accuracy and loss curves for the Balanced dataset (Empathetic Dialogues).

The pre-trained BERT model was also evaluated with accuracy performance measures on the topical chat and Empathetic Dialogue datasets. Hyperparameters used for the pre-trained BERT model are as follows:

Hyperparameters Used –
Epochs =12
Tokenizer = Bert-base-uncased
Batch size = 10
Optimizer = Adam optimizer
BERT Model Parameters - 12-layer, 768-hidden, 12-heads,
110M parameters (Trained on lower-cased English text)

The pre-trained BERT model showed an accuracy of 53% on the unbalanced topical chat dataset. The pre-trained BERT model showed an accuracy of 56% on the balanced, empathetic dialogues dataset. Table 5 shows the result of the pre-trained BERT model with both datasets.

Table 5: Results of Pre-trained BERT model.

Pre-trained Model	Unbalanced (Topical Chat Dataset) (% accuracy)	Balanced (Empathetic Dialogues) (%accuracy)
	BERT Tokenizer	
BERT	0.53	0.56

A firm conclusion cannot be drawn from accessing the models based solely on the accuracy values. Therefore, an additional study was conducted. For each method, classification reports were created in order to assess the models' performance further. The performance of a model is examined using additional parameters, such as f1-score, Precision and recall, and in some circumstances with an unbalanced dataset.

The pertinent forecasts among all the identified values might be referred to as Precision. When making the wrong prediction might be more expensive than making the right one, accuracy is crucial. The following formula can give Precision:

$$\text{Precision} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Positives})}$$

Figures 09, 12, and 15 show the precision score comparison for different machine learning techniques using count vectorization, TF-IDF vectorization and hashing vectorization methods.

Next is recall, which is the proportion of true labels to properly predicted labels in a model. If we don't want to overlook any predictions at the expense of making incorrect predictions, recall is essential. Recall can be formulated as follows:

$$\text{Recall} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Negatives})}$$

Figures 8, 11, and 14 compare the recall score for different machine algorithms using count vectorization, TF-IDF vectorization and hashing vectorization methods. Essentially, recall is influenced by positive labels and less affected by negative labels. In Precision, all positive labels are considered, whether accurately or inaccurately labelled. So, recall takes care of correctly classifying all positive samples.

Finally, the harmonic average of Precision and recall results in as F1- score. Figures 10, 13 and 16 show the F1-score comparison for different machine algorithms using count vectorization, TF-IDF vectorization and hashing vectorization methods.

$$\text{F1-Score} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

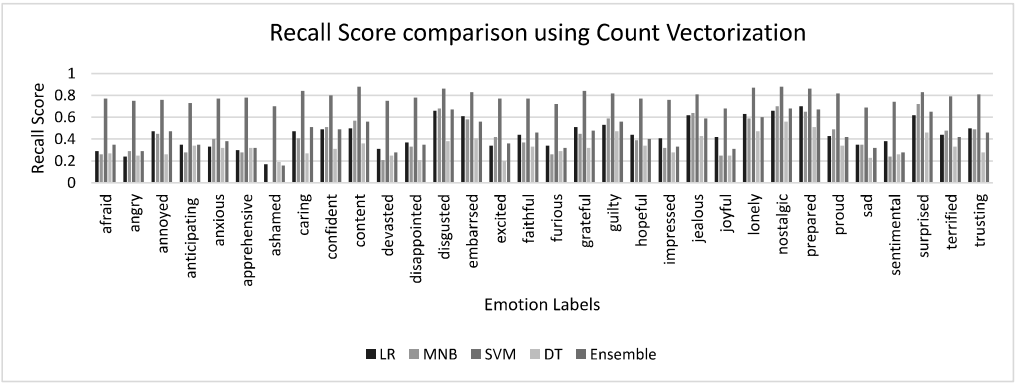


Figure 8: Recall score comparison for machine learning algorithms using count vectorization on Balanced Dataset (Empathetic Dialogues).

Table 6 summarises the results with different performance metrics of different classification models with various feature vectorization methods on balanced Datasets (Empathetic Dialogues) and unbalanced datasets (topical chat dataset). Table 6 outlines that the authors evaluated two datasets, the Empathetic Dialogue dataset and the Topical chat dataset, with count

vectorization, TF-IDF vectorization, and hashing vectorization, on baseline machine learning algorithms, logistic regression, multinomial naïve Bayes, Support vector machine, decision trees, and ensemble methods. Similarly, both datasets are also evaluated on baseline text classification deep learning models such as LSTM, Bi-LSTM, GRU, and Bi-GRU with One hot encoding. Moreover, the pre-trained BERT model was evaluated on both datasets with a BERT tokenizer. All machine learning, deep learning and pretrained BERT model were evaluated for different performance metrics, as shown in Table 6, F1-score, accuracy, Precision and recall. From Table 6, the authors can infer that SVM with a count vectorizer has the highest accuracy in machine-learning models for balanced datasets. An ensemble with count vectorization has the highest accuracy for the unbalanced dataset. Similarly, the authors also compared their results with some of the state-of-the-art techniques from various articles. Table 7 compares the results of some research articles with different machine learning and deep learning models with results evaluated by the authors. The authors compared their results with the results of other works based on techniques used and whether balanced or imbalanced datasets. It has been observed that SVM classification in machine learning models has given good results similar to other research articles. Whereas Bi-LSTM in deep learning models has performed well on unbalanced datasets.

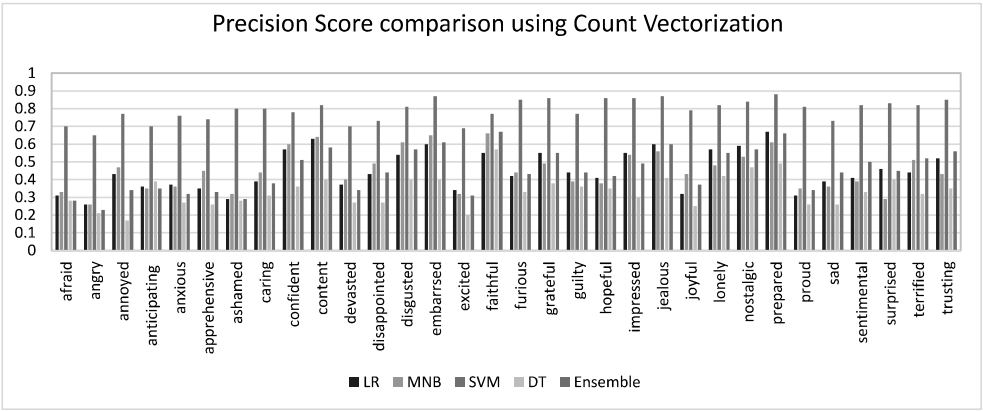


Figure 9: Precision score comparison for machine learning algorithms using count vectorization on the balanced dataset (Empathetic Dialogues).

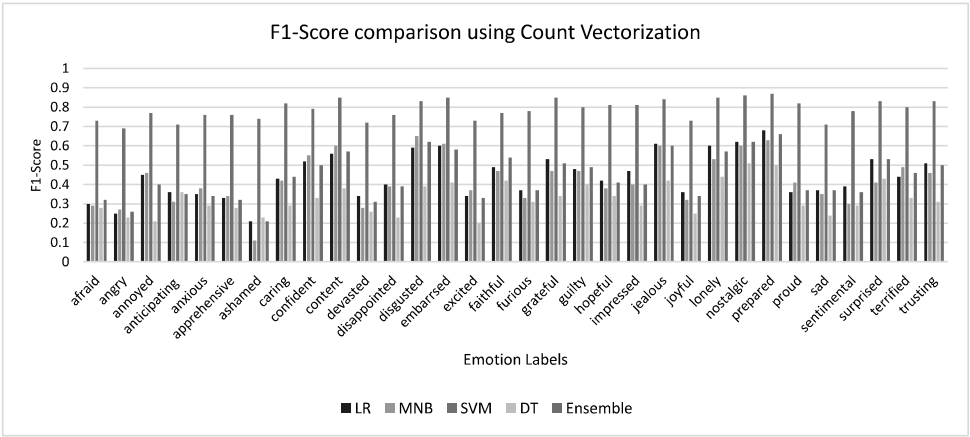


Figure 10: F1-Score comparison for machine learning algorithms for emotion classes using count vectorization on Balanced Dataset (Empathetic Dialogues).

Table 6: Different Performance metrics of classification models with various features on balanced and unbalanced datasets.

Dataset	Metric	Model and Features																		
		LR Count Vector	LR TF-IDF Vector	LR Hashing Vector	MNB Count Vector	MNB TF-IDF Vector	MNB Hashing Vector	SVM Count Vector	SVM TF-IDF Vector	SVM Hashing Vector	DT Count Vector	DT TF-IDF Vector	DT Hashing Vector	Ense-mble Count Vector	Ensemble TF-IDF Vector	Ensemble Hashing Vector	LSTM (One Hot Encoding)	GRU (One Hot Encoding)	Bi-LSTM (One Hot Encoding)	Bi-GRU (One Hot Encoding)	BERT (BERT Tokenizer)
Balanced Dataset (Empathetic Dialogues)	F1-Score	.4457	.6449	.4413	.4268	.4080	.2954	. 7887	.4432	.3087	.3279	.2996	.3104	.4952	.4488	.3390	.5224	.4781	.5362	.5875	.5528
	Accuracy	.4513	.6522	.4456	.4363	.4104	.2829	. 7893	.4447	.30	.3288	.3063	.3148	.5015	.4532	.3358	.5759	.5384	.5727	.6134	.5569
	Precision	.45	.67	.46	.45	.53	.40	.79	.46	.31	.33	.30	.31	.50	.49	.37	.76	.73	.73	.76	.55
	Recall	.45	.64	.44	.43	.39	.28	.79	.45	.30	.33	.30	.31	.50	.45	.34	.40	.36	.42	.48	.56
Imbalanced Dataset (Topical Chat Dataset)	F1-Score	.52	.5312	.5349	.50	.5813	.60	.5234	.54	.5542	.4918	.5428	.55	.5983	.5969	.6009	.40	.41	.41	.41	.5125
	Accuracy	.4894	.4984	.4974	.4781	.4489	.4307	.5558	.4882	.4889	.4479	.4461	.4467	.6000	.4400	.4295	.5943	.6679	.6850	.6743	.5307
	Precision	.57	.60	.62	.54	.92	1.00	.71	.67	.73	.57	.78	.83	.99	.98	1.00	.40	.40	.41	.40	.52
	Recall	.49	.50	.50	.48	.45	.43	.45	.49	.49	.45	.45	.45	.44	.44	.43	.41	.42	.42	.42	.53

Table 7: Comparison of SOTA methods with our results.

Reference	Dataset	Performance Metric (in %)	Method
[36]	ISEAR	72.43 (Accuracy)	SVM Classification
[17]	SemEval-2019	77.00(F-measure)	BERT
[37]	Aman	73.89(Accuracy)	Naïve Bayes And SVM
[38]	SemEval-2018	58.80 (Accuracy)	Bi-LSTM
[39]	ISEAR, SemEval	42.2(Precision)	LSTM
[26]	SemEval-2018	39.80(Accuracy)	Bi-GRU
[40]	Manually built	76.9(Accuracy)	SVM, Decision Tree, NB,
[41]	SemEval-2018 (Imbalanced Dataset)	52.00 (Accuracy)	SVM, Logistic Regression
Our Best results	Empathetic Dialogues (Balanced)	78.93(Accuracy)	SVM Classification Using Count Vectorization
Our Best results	Topical Chat Dataset (Unbalanced)	68.50 (Accuracy)	Bi-LSTM

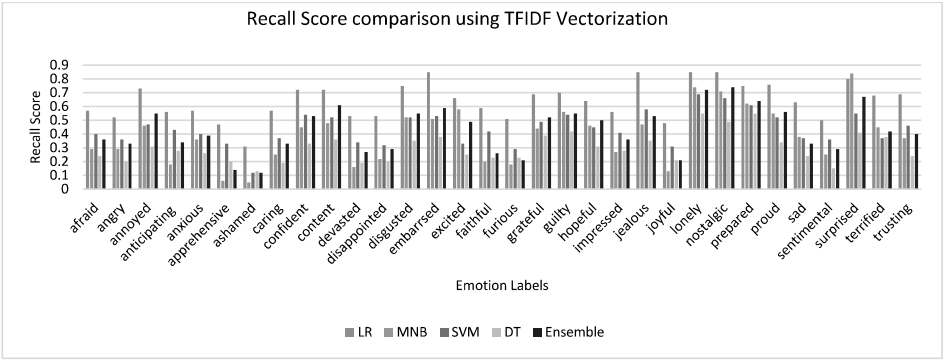


Figure 11: Recall Score comparison for machine learning algorithms for emotion classes using TF-IDF vectorization on Balanced Dataset (Empathetic Dialogues).

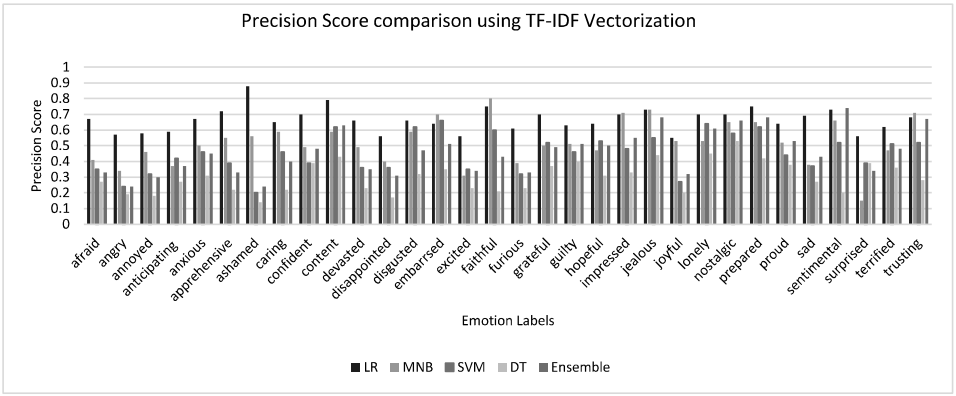


Figure 12: Precision Score comparison for machine learning algorithms for emotion classes using TF-IDF vectorization on the balanced dataset (Empathetic Dialogues).

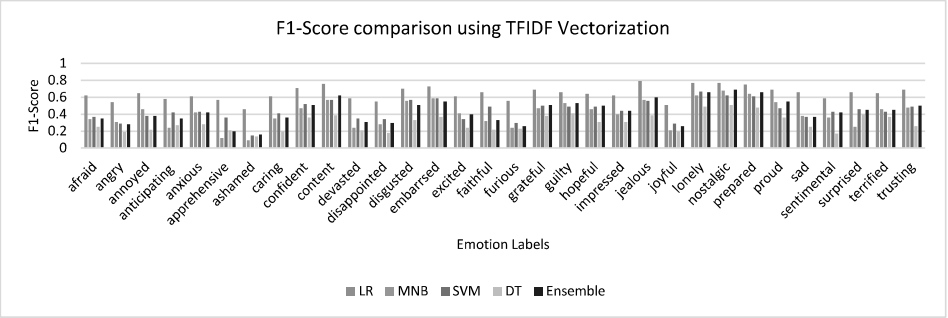


Figure 13: F1-Score comparison for machine learning algorithms for emotion classes using TF-IDF vectorization balanced dataset (Empathetic Dialogues).

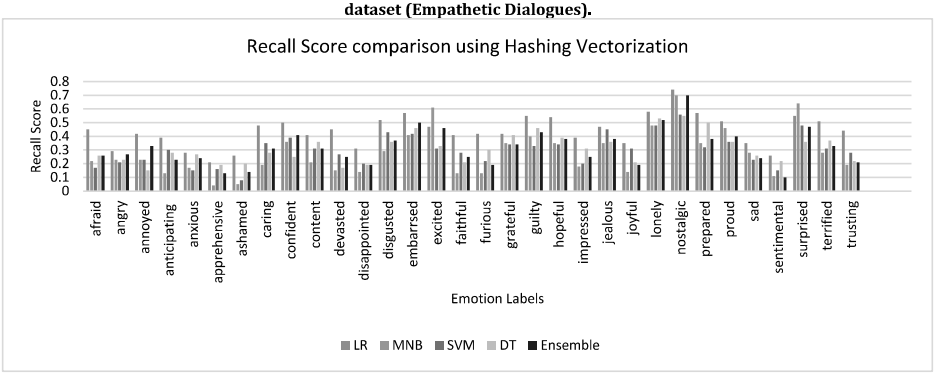


Figure 14: Recall Score comparison for machine learning algorithms for emotion classes using hashing vectorization Balanced Dataset (Empathetic Dialogues).

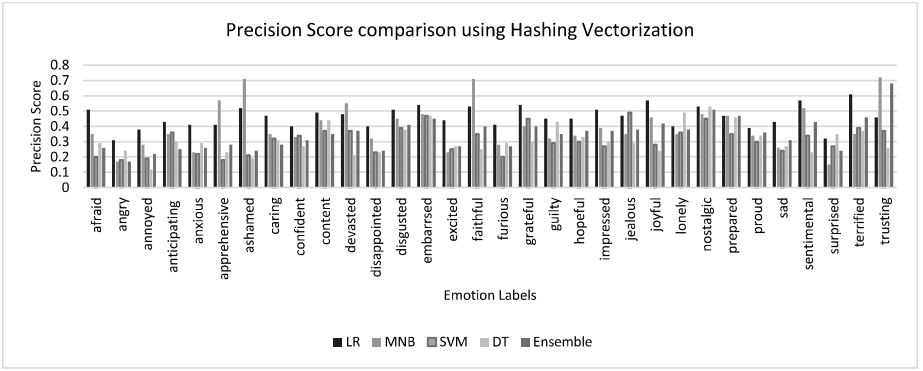


Figure 15: Precision Score comparison for machine learning algorithms for emotion classes using hashing vectorization Balanced Dataset (Empathetic Dialogues).

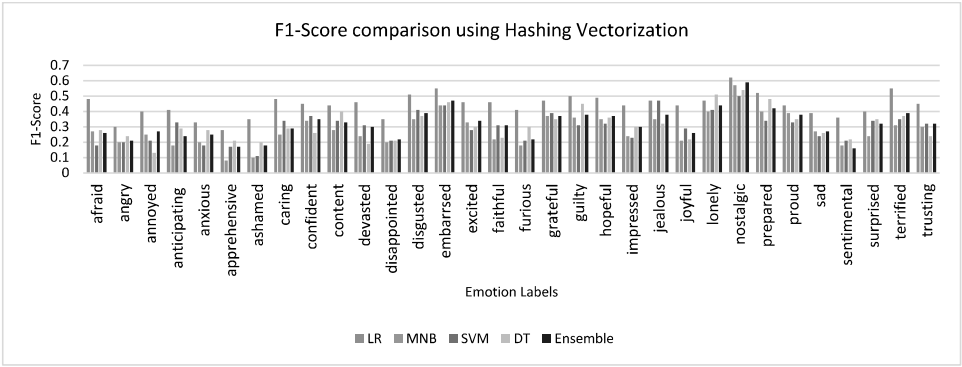


Figure 16: F1-Score comparison for machine learning algorithms for emotion classes using hashing vectorization on the balanced dataset (Empathetic Dialogues).

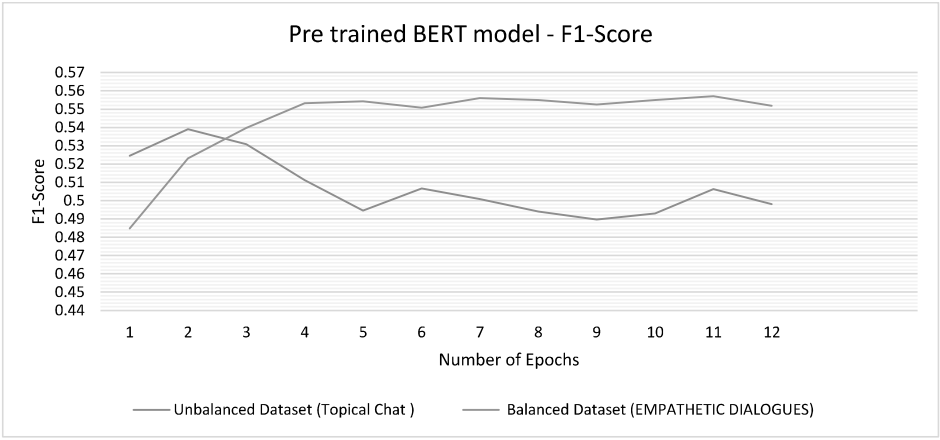


Figure 17: F1-score (weighted) curves using a pre-trained BERT model for the Topical Chat dataset (Unbalanced) and the Empathetic Dialogues dataset (Unbalanced).

Figure 17 shows the F1-score (weighted) average curves of the pre-trained BERT model for the Topical Chat dataset (Unbalanced and Empathetic Dialogues dataset (Unbalanced). The authors can infer from Figure 17 that the pre-trained BERT model has shown greater performance on the balanced dataset (the Empathetic Dialogue) than on the unbalanced dataset (Topical Chat). Figure 6 and 7 signifies variance error in the training and testing performance. This may have occurred due to data imbalance. If there is an imbalance in the training data between the classes, the model may show to bias the dominant class. Even for the minority class, this can lead to strong training performance but poor generalization. The topical chat dataset has an unbalanced distribution of emotion labels, and EMPATHETIC DIALOGUES has an approximately balanced distribution of emotion labels, but still, data distribution has an imbalance. Data imbalance issues can be addressed by sampling techniques such as Random under-sampling, which subtracts samples from the majority class, and random over-sampling, which adds copies to the minority class, are common strategies. Results can be typically enhanced by using more complex approaches, such Smote - Synthetic Minority Oversampling Technique.

Table 8: Summary of research gaps with suggested AI techniques in text-based emotion detection.

Problem Area	Research Gaps	AI-technique	References
Datasets	Lacking labelled/annotated datasets Imbalanced datasets Domain Dependent datasets Language dependent datasets Mislabeled emotions	Deep learning models Pre-trained language models	[50], [51], [48]
Accuracy	Several techniques are not robust Accurateness of current systems	Machine learning models Deep learning models Pre-trained language models	[52], [53] [54], [55], [56]
Quality of text	Incomplete information Typing mistakes Slang words Short texts	Natural language processing Machine Learning models Pre-trained language models	[57], [56] , [12], [58]
Semantic	Inability to recognize implicit emotions Detection of sarcasm, irony, harmony	Pre-trained language models Deep Learning algorithms	[57], [56], [59], [60]

While our work focuses on the many text-based emotion detection findings, there are still specific issues and future research paths that need to be addressed in order to make emotion detection more in line with practical needs. Table 8 summarizes the major problem area and related challenges in the field of Text-Based Emotion Detection using Artificial Intelligence are discussed. The authors can make a conclusion that Artificial Intelligence have played a role in the considerable advancement of text-based emotion detection in recent years. The development of several novel concepts, including pretrained embedding, various attention mechanisms, transformer-based models, and pretrained deep learning models, has led to significant growth in recent years.

5 Conclusion –

This paper compares different AI-based techniques utilized in text-based emotion detection (TBED) for conversational agents. Machine learning, deep learning and pretrained BERT model have been evaluated on the unbalanced topical chat dataset and the balanced, empathetic dialogues dataset. Among all three AI-based techniques, machine models performed exceptionally well compared to advanced deep learning models. In machine learning techniques, vectorization methods such as count vectorizer, TF-IDF vectorizer, and hashing vectorizer have been used for feature extraction and applied to SVM, MNB, DT, LR and ensemble voting learning algorithms. It has been observed that the SVM classifier with TF-IDF performs better for the unbalanced dataset than other machine learning classifiers. For a balanced dataset, DT with a count vectorizer performs well. In deep learning, Bi-GRU results better for unbalanced datasets, while Bi-LSTM results for the balanced dataset. Pre-trained BERT model results better for the balanced dataset. Although machine learning models have shown better performance for the unbalanced and balanced dataset, models have overfitted while training. Therefore, to solve the overfitting problem, the authors suggest techniques such as cross-validation or data augmentation in future work, which enhance the models' performance and help correctly detect the emotions from the text. Similarly, in the case of the unbalanced dataset, the authors recommend resampling techniques in future work. Furthermore, these techniques can be evaluated on other conversational textual datasets such as DAILY DIALOG and NLPCC 2017 in the future.

REFERENCES

[1] R. Bavaresco *et al.*, "Conversational agents in business: A systematic literature review and future research directions," *Comput Sci Rev*, vol. 36, p. 100239, 2020, doi: 10.1016/j.cosrev.2020.100239.

[2] M. Nuruzzaman and O. K. Hussain, "A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks," in *Proceedings - 2018 IEEE 15th International Conference on e-Business Engineering, ICEBE 2018*, Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 54–61. doi: 10.1109/ICEBE.2018.00019.

[3] S. Hobert and R. Meyer von Wolff, "Say Hello to Your New Automated Tutor – A Structured Literature Review on Pedagogical Conversational Agents," *Wirtschaftsinformatik*, pp. 301–314, 2019.

[4] J. Fraser, I. Papaioannou, and O. Lemon, "Spoken Conversational AI in Video Games," pp. 179–184, 2018, doi: 10.1145/3267851.3267896.

[5] J. L. Z. Montenegro, C. A. da Costa, and R. da Rosa Righi, "Survey of conversational agents in health," *Expert Syst Appl*, vol. 129, pp. 56–67, 2019, doi: 10.1016/j.eswa.2019.03.054.

[6] R. W. Picard, "Affective Computing." [Online]. Available: http://www.media.mit.edu/~picard/

[7] H. Prendinger and M. Ishizuka, "The empathic companion: A character-based interface that addresses users' affective states," *Applied Artificial Intelligence*, vol. 19, no. 3–4, pp. 267–285, Mar. 2005, doi: 10.1080/08839510590910174.

[8] J. Tao, "Context Based Emotion Detection from Text Input." [Online]. Available: http://www.isca-speech.org/archive

[9] O. Udochukwu and Y. He, "A Rule-Based Approach to Implicit Emotion Detection in Text."

[10] Institute of Electrical and Electronics Engineers, *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE).*

[11] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, Feb. 2021, doi: 10.1016/j.future.2020.08.005.

[12] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, "Text-based emotion detection: Advances, challenges, and opportunities," *Engineering Reports*, vol. 2, no. 7. John Wiley and Sons Inc, Jul. 01, 2020. doi: 10.1002/eng2.12189.

[13] M. Mnasri, "Recent advances in conversational NLP : Towards the standardization of Chatbot building," Mar. 2019, [Online]. Available: http://arxiv.org/abs/1903.09025

[14] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial Paranoia'," 1971.

[15] M. Skowron, "Affect Listeners. Acquisition of Affective States by means of Conversational Systems." [Online]. Available: http://www.ofai.at

[16] N. v Kolekar, P. Gauri Rao, S. Dey, M. Mane, V. Jadhav, and S. Patil, "SENTIMENT ANALYSIS AND CLASSIFICATION USING LEXICON-BASED APPROACH AND ADDRESSING POLARITY SHIFT PROBLEM," *J Theor Appl Inf Technol*, vol. 15, no. 1, 2016, [Online]. Available: www.jatit.org

[17] A. Basile, M. Franco-Salvador, N. Pawar, S. Sanjaštajner, M. C. Rios, and Y. Benajiba, "SymantoResearch at SemEval-2019 Task 3: Combined Neural Models for Emotion Classification in Human-Chatbot Conversations."

[18] A. Adikari, D. de Silva, D. Alahakoon, and X. Yu, "A Cognitive Model for Emotion Awareness in Industrial Chatbots," 2019.

[19] M. Feidakis, P. Kasnesis, E. Giatraki, C. Giannousis, C. Patrikakis, and P. Monachelis, "Building pedagogical conversational agents, affectively correct," in *CSEDU 2019 - Proceedings of the 11th International Conference on Computer Supported Education*, SciTePress, 2019, pp. 100–107. doi: 10.5220/0007771001000107.

[20] Y. Xie and P. Pu, "Empathetic Dialog Generation with Fine-Grained Intents," May 2021, [Online]. Available: http://arxiv.org/abs/2105.06829

[21] S. Patil, V. M. Mudaliar, P. Kamat, and S. Gite, "LSTM based Ensemble Network to enhance the learning of long-term dependencies in chatbot," *International Journal for Simulation and Multidisciplinary Design Optimization*, vol. 11, 2020, doi: 10.1051/smdo/2020019.

[22] C. Huang, O. R. Za¨ıaneza¨ıane, A. Trabelsi, and N. Dziri, "Automatic Dialogue Generation with Expressed Emotions." [Online]. Available: http://www.cs.ualberta.ca/

[23] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.01074

[24] Y. Sun and Y. Zhang, "Conversational recommender system," in *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, Association for Computing Machinery, Inc, Jun. 2018, pp. 235–244. doi: 10.1145/3209978.3210002.

[25] S. Ghosh, D. Varshney, A. Ekbal, and P. Bhattacharyya, "Context and Knowledge Enriched Transformer Framework for Emotion Recognition in Conversations," in *Proceedings of the International Joint Conference on Neural Networks*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021. doi: 10.1109/IJCNN52387.2021.9533452.

[26] A. Ezen-Can SAS Inst and E. F. Can SAS Inst, "RNN for Affects at SemEval-2018 Task 1: Formulating Affect Identification as a Binary Classification Problem."

[27] Y. Shou, T. Meng, W. Ai, S. Yang, and K. Li, "Conversational emotion recognition studies based on graph convolutional neural networks and a dependent syntactic analysis," *Neurocomputing*, vol. 501, pp. 629–639, Aug. 2022, doi: 10.1016/j.neucom.2022.06.072.

[28] G. Tu, J. Wen, C. Liu, D. Jiang, and E. Cambria, "Context-and Sentiment-Aware Networks for Emotion Recognition in Conversation," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 5, pp. 699–708, Oct. 2022, doi: 10.1109/TAI.2022.3149234.

[29] N. T. Thomas, "An e-business chatbot using AIML and LSA," in *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*, Institute of Electrical and Electronics Engineers Inc., Nov. 2016, pp. 2740–2742. doi: 10.1109/ICACCI.2016.7732476.

[30] Y. C. Chang and Y. C. Hsing, "Emotion-infused deep neural network for emotionally resonant conversation," *Appl Soft Comput*, vol. 113, Dec. 2021, doi: 10.1016/j.asoc.2021.107861.

[31] L. De Bruyne, O. De Clercq, and V. Hoste, "LT3 at SemEval-2018 Task 1: A classifier chain to detect emotions in tweets."

[32] S. B. Suhasini M, "Emotion detection framework for Twitter data using supervised classifers,"

[33] Merav Allouch; Amos Azaria; Rina Azoulay; Ester Ben-Izchak; Moti Zwilling;, "Automatic Detection of Insulting Sentences in Conversation".

[34] N. , N. T.-H. , N. Y.-N. , D. D. , N. M. and N. V.-H. Nguyen, "Machine learning-based model for customer emotion detection in hotel booking services," *Journal of Hospitality and Tourism Insights*, 2023.

[35] A. Basile, M. Franco-Salvador, N. Pawar, S. Sanjaštajner, M. C. Rios, and Y. Benajiba, "SymantoResearch at SemEval-2019 Task 3: Combined Neural Models for Emotion Classification in Human-Chatbot Conversations."

[36] K. Shrivastava, S. Kumar, and D. K. Jain, "An effective approach for emotion detection in multimedia text data using sequence based convolutional neural network," *Multimed Tools Appl*, vol. 78, no. 20, pp. 29607–29639, Oct. 2019, doi: 10.1007/s11042-019-07813-9.

[37] L. Claudio Diogo Reis, F. Cristina Bernardini, S. Bacellar Leal Ferreira, and C. Cappelli, "ICT Governance in Brazilian Smart Cities:: An Integrative Approach in the Context of Digital Transformation," pp. 302–316. doi: 10.1145/3463677.3463682.

[38] J. Xiao, "Figure Eight at SemEval-2019 Task 3: Ensemble of Transfer Learning Methods for Contextual Emotion Detection." [Online]. Available: https://github.com/fastai/fastai

[39] José Ángel González, Lluís-F. Hurtado, Ferran Pla. "Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. ," *nf Process Manag*, vol. 57(4):102262., 2020.

[40] Cortiz D., "Exploring transformers in emotion recognition: a comparison of BERT, DistillBERT, RoBERTa, XLNet and ELECTRA," 2021.

[41] S. Kusal, S. Patil, K. Kotecha, R. Aluvalu, and V. Varadarajan, "Ai based emotion detection for textual big data: Techniques and contribution," *Big Data and Cognitive Computing*, vol. 5, no. 3, 2021, doi: 10.3390/bdcc5030043.

[42] S. Kusal, S. Patil, J. Choudrie, K. Kotecha, S. Mishra, and A. Abraham, "AI-Based Conversational Agents: A Scoping Review From Technologies to Future Directions," *IEEE Access*, vol. 10, pp. 92337–92356, 2022, doi: 10.1109/ACCESS.2022.3201144.

[43] K. Gopalakrishnan *et al.*, "Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations."

[44] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, "Towards Empathetic Open-domain Conversation Models: a New Benchmark and Dataset," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1811.00207

[45] N. Anantrasirichai and D. Bull, "Artificial intelligence in the creative industries: a review," *Artif Intell Rev*, vol. 55, no. 1, pp. 589–656, Jan. 2022, doi: 10.1007/s10462-021-10039-7.

[46] Lovejit Singh, Sarbjeet Singh, and Naveen Aggarwal, "Two-Stage Text Feature Selection Method for Human Emotion Recognition," *Lecture Notes in Networks and Systems* , pp. 531–538, Sep. 2018.

[47] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2007, pp. 196–205. doi: 10.1007/978-3-540-74628-7_27.

[48] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis."

[49] A. Balahur, J. M. Hermida, and A. Montoyo, "Building and exploiting EmotiNet, a knowledge base for emotion detection based on the appraisal theory model," *IEEE Trans Affect Comput*, vol. 3, no. 1, pp. 88–101, Jan. 2012, doi: 10.1109/T-AFFC.2011.33.

[50] F. ; L. X. ; Y. C. ; Z. S. ; Z. J. ; Q. S. Huang, . "Attention-Emotion-Enhanced Convolutional LSTM for Sentiment Analysis.," *IEEE Trans. Neural Networks Learn*, 2021.

[51] Z. ; J. R. ; E. A. ; B. P. Ahmad, "Borrow from rich cousin: Transfer learning for emotion detection using cross lingual embedding," *Expert Syst. Appl*, 2020.

[52] M. G. S. ; R. L.-F. ; G.-G. J. O. Ortega, "Towards emotion recognition from contextual information using machine learning," *Ambient. Intell. Humaniz. Comput*, 2020.

[53] Y. ; Z. L. ; H. D. ; Z. R. ; W. G. Lai, "Fine-grained emotion classification of Chinese microblogs based on graph convolution networks," *World Wide Web 2020*, 2020.

[54] F. A. ; Acheampong and Nunoo-Mensah, "Transformer models for text-based emotion detection: A review of BERT-based approaches. ".

[55] C. Huang, A. Trabelsi, and O. R. Zaïane, "ANA at SemEval-2019 Task 3: Contextual Emotion detection in Conversations through hierarchical LSTMs and BERT," Mar. 2019, [Online]. Available: http://arxiv.org/abs/1904.00132

[56] Y.-H. Huang, S.-R. Lee, M.-Y. Ma, Y.-H. Chen, Y.-W. Yu, and Y.-S. Chen, "EmotionX-IDEA: Emotion BERT-an Affectional Model for Conversation." [Online]. Available: http://nlp.mathcs.emory.edu

[57] S. M. Mohammad, "Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text," 2015.

[58] C. L. J. C. Shiliang Sun, "A review of natural language processing techniques for opinion mining systems," *Information Fusion*, vol. 36, pp. 10–25, 2017.

[59] A. ; G. G. Kumar, "Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets," *J. Ambient. Intell. Humaniz. Comput*, 2019.

[60] S. Zhang, X. Zhang, J. Chan, and P. Rosso, "Irony Detection via Sentiment-based Transfer Learning.", Information Processing & Management, Volume 56, Issue 5, 2019, Pages 1633-1644, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2019.04.006.