[Koushik Chowdhury]

[Towards a Deep Learning approach to regularise discourse of collaborative learner]

[Master's Thesis]
submitted on [12.05.2023]

Saarland University

Faculty of Mathematics and Computer Science
Department of Computer Science

Human Computer Interaction Lab
Saarland University

Advisor
Birk Thierfelder
Research Assistant, PhD Student
Department of Educational Technology
Saarland University

1st Supervisor
Prof. Dr. Anna Maria Feit
Human-Computer Interaction
Department of Computer Science
Saarland University

2nd Supervisor
Prof. Dr. Armin Weinberger
Head and Founder
Department of Educational Technology
Saarland University

Human-Computer Interact Lab
Campus - Building E1.7
Saarland Informatics Campus
Saarland University
66123 Saarbrücken
Germany

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Koushik Chowdhury (e-sign)

Saarbrücken, [12.05.2023]

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Koushik Chowdhury (e-sign)

Saarbrücken, [12.05.2023]

Acknowledgments

I would like to express my gratitude to Prof. Dr. Armin Weinberger for the opportunity he provided me with to work on my Master's Thesis in the Education Technology Department, as well as for the invitation he extended to me to attend their Master's Thesis seminar and for the comments I received during the seminar.

In addition, I would also like to express my gratitude to Prof. Dr. Anna Maria Feit for agreeing to be my first supervisor despite the fact that she has a very full schedule. In addition, I would like to express my gratitude to her for providing me with insightful comments regarding the master's thesis seminar presentation as well as the report.

Finally, a big thank you to Mr. Birk Thierfelder for his assistance in guiding me through the process of implementing and writing my thesis.

Abstract

Collaborative learning is a method of education in which a group of learners solves a particular task. A collaborative setting encourages learners to take a more active role in knowledge construction. However, when they communicate on a virtual platform such as a chat platform, it is important that they can refer to each other correctly so that they can improve their learning activities with the help of each other, but learners can be sidetracked, which retards their learning progress. To address this issue, this thesis practiced text classification approaches to regularize the conversation between learners so they could refer to each other correctly. The dataset was collected from a focus group experiment designed for students in the Educational Technology Department at Saarland University. The report gives a clear idea of how the collected dataset has been coded and validated with the help of intercoder reliability measurements. After data preprocessing, state-of-the-art data augmentation techniques such as spelling, insertion, substitution, and synonym augmentation are applied. The thesis examines various neural network models to identify the best model for the dataset. Among them, Bidirectional Encoder Representations from Transformers (BERT) provides the best performance with an accuracy of 0.94 and a 0.17 loss value for the augmented preprocessed dataset, where recurrent neural network models tend to overfit. In the evaluation part, a summary of performance matrices is shown, and to evaluate the model, a new dataset with similar data is generated with the help of the OpenAI API Key. The BERT model is able to classify 960 responses out of 1005, where both recurrent neural networks are classified less than 200. The thesis also discussed the issue of model poisoning so that when the model is updated, it can tackle the unclassified responses. Finally, a simple demo of how this BERT model is used to regularize the discourse of two collaborative learners is presented with the help of the Jupyter interface.

Contents

1 Introduction

This section focuses on the thesis's motivation, context, research questions, contributions, and challenges in general. Additionally, this part highlights the overall works and provides a brief explanation of the structure of the thesis.

1.1 Context and Motivation

Collaborative learning often involves students working together to improve their education. Through the use of collaborative learning, learners from a collaborative group are encouraged to participate in order to accomplish a common goal as a group rather than on an individual basis. Collaboration learning environments in education encompass not only learning but also learning and teaching involving groups of learners collaborating to solve a problem [24]. Learners have the opportunity to converse with agreement and disagreement, as well as to share diverse perspectives on a particular task. The primary objective of collaborative learning is not for learners to converse with one another while performing individual tasks but rather to assist one another in completing the assignment or task. In collaborative learning environments, learners share their ideas and assist one another by providing feedback, which entails learners referencing one another on a given task to improve learning outcomes.

In collaborative settings, it is necessary to regularize learners so that they can share their thoughts and beliefs on a specific task rather than other non-specified tasks. This is one of the challenging problem to motivate learner to help each other on a specifc task or assignment so that the main goal of collaborative learnig is fullfiled. The development of chatbots in the educational field is not new. Most of the chatbots developed in the educational field serve in a one-way setting, which means most of the chatbots are designed to teach learners to develop their insight behaviors, which means those chatbots work as a teaching agent rather than motivating learners to assist each other on a specific problem [23]. These chatbots have a predetermined function in meeting the needs of learners [23]. Therefore, one of the main issues in the collaborative learning field is the regularization of collaborative learners so that they can assist each other in collaborative settings.

When it comes to overcoming this challenge in contexts that include collaboration, informatics plays a key role here. This issue is the primary emphasis of this thesis, and the primary solution proposed is to regularize collaborative learning environments, such as chatbot platforms so that students can enhance their learning performance by collaborating with one another. Throughout the entirety of the thesis, a number of different computation tasks have been explored and put into practice in order to make it practical.

The first chatbot system, named 'ELIZA', was implemented in the 1966s and was based on a discussion between a machine and a human [46]. So studying and creating chatbots is nothing new. Nearly all businesses and services that interact with customers now have their own AI chatbot. It is now popular in the educational sector as well. Many educational institutions have created their own chatbots to assist and boost student productivity [7]. The chatbot functions on data that is provided to it with the use of natural language processing.

Typically chatbots are based on human-to-agent communication, in which the human asks a question and the agent responds based on the question. However, this thesis is concerned about two-way settings, in which people essentially talk to one another and agents react based on their activity. The agent will interrupt the learner to continue the conversation that the other learner started, for instance, if one learner asks a question or started a discussion and the second learner did not respond for a while or responded to a different conversation.

The goal of this thesis's originality is to provide a platform where a chat agent can maintain the dialogue between two collaborative learners with the help of a deep learning approach. This includes regularizing the discourse of collaborative learners from the point of view of educational technology by ensuring boundaries in the educational chat structure. It enables an easier understanding of the discussion concept between two learners. Following are the objectives to achieve the goal:

- To preprocess the collected data as well as apply augmentation techniques to enlarge the dataset so that neural network models can learn more diverse patterns to improve accuracy.
- To apply different types of neural network models to find which neural network model works better with the collected data.
- To apply the evaluation technique to evaluate the neural network model results to select the best model for the chatbot.

In an ideal situation, for example, two learners are conversing online about a particular topic while sitting in different locations. What kind of issues could arise in this scenario, for instance: a learner could become sidetracked from the discourse and begin to discuss unrelated issues or write unrelated material. Second, people have a propensity to multitask when they are online, which can be detrimental to the purpose of a conversation or lead to a lack of comprehension of the proper subject or inquiry. Now imagine that some researcher wishes to enhance or create a system based on this chat response, but when the researcher analyzes the data, they find that there are many responses that are illogical or irrelevant. In other words, if someone wants to develop or build some kind of system, such as a chat platform in the educational field, for learners or students, they cannot adequately train the dataset. If the chatbot is concerned with the text classification approach, then it would be difficult for the chatbot to correctly classify texts if the machine learning model trains with this illogical or irrelevant data. Consider another scenario in which two learners are attempting to tackle an academic assignment on a chat platform designed for educational learning, and one learner abruptly responds out of topic because the other learner does not grasp the other learner's context. This would be detrimental to their ability to learn from each other.

Some of the most cutting-edge chatbots are covered in Section 2 of this document. The earlier chatbots all focused on a human inquiry and a platform where numerous learners could contribute and improve their academic abilities.

1.2 *Research Questions*

The main intention of this thesis is to go through different types of state-of-the-art neural network models to find the best model for constructing a chatbot that can regularize the discourse of collaborative learners. Having a thorough understanding of the most recent state of the art in this sector is essential to achieving this goal. The following constitutes the thesis's research questions.

A. How can the classification technique be used to classify responses from a user to check whether he or she refers to another user or not?

B. How would the different tactics used by the different neural network models lead to different results?

C. Why is it necessary to apply data augmentation techniques?

D. How well does the evaluation strategy work in assessing the results?

To answer question A, a rule-based classification technique is explored, which helps to identify messages so that when both users respond, it will classify both responses to see whether they are in the same class or not. In this research, various neural network models are explored to answer question B because finding the right model is crucial. This thesis investigates different data augmentation techniques such as insertion augmentation, spelling augmentation, substitute augmentation, and synonym augmentation for research question C. This research uses an evaluation technique such as the confusion matrix as well as tests all of the implemented neural network models on a new dataset that is generated with the help of the OpenAI API key to answer research question D. The dataset is called 'EduTech Dataset' and was collected from the Department of Educational Technology.

1.3 *Challenges*

The challenges are listed below.

A. Al agents struggle to connect with people: Many AI agents are unable to comprehend commands well enough to respond appropriately if the commands are slightly out of sequence. This thesis focuses on the regularization of human discourse. Therefore, in order to maintain the discussion between two humans, the chatbot must comprehend the context of the dialogue. The problem is attempted to be solved by using the best architecture.

B. It is absolutely necessary to test the result to see if it is similar to or close to the training result because even after finding the best architecture, it is still possible that the agent will not respond appropriately.

C. A proper pre-processing step must be followed before using the algorithm in order to avoid the risk of receiving a better result. There is always a chance to get very good and very bad results from algorithms where the model trains the data very well, such as overfitting, or the model trains the data accurately but predicts inaccurately, such as underfitting.

D. Although the algorithm produces good results, there is a chance that the evaluation will be subpar.

E. A large portion of earlier works relied on one-way chatbots, in which a person would ask a question to the chatbot, which would then attempt to grasp the context and response respectively. Since the chatbot in this thesis will not respond to humans but it will instead regulate a conversation between two humans so that they adhere to the same conversational discourse, therefore, creating a chatbot that can regularize conversation between two humans is one of the biggest challenges.

1.4 Contributions

The following are the contributions of this research.

A. A pre-trained neural network model that outperforms non-pretrained neural network models and boosts overall performance, and that is capable of correctly classifying texts 75% more often than non-pretrained models.

B. A non-textbook strategy in the evaluation part, in addition to the usual evaluation matrices, in which a similar dataset has been developed with the help of the OpenAI API Key in order to test how well all of the employed neural network models would be able to accurately classify responses in the new dataset.

C. A non-traditional chatbot that does not deliver answers to users but rather assists learners in collaborative learning to check how they refer to each other in order to improve the learners' available learning opportunities.

1.5 Structure of Thesis

The thesis is structured as followed:

Introduction: Introduces to the audience the context of the thesis, the motivation for the thesis, the research questions, the expected contribution, the research method and any challenges.

Related Work: Describes previous work on chatbots as well as how the knowledge gained from previous work is helpful to this thesis.

Background Theory: To help the audience understand the entire thesis, give a brief explanation of the suggested algorithm, the theoretical concept underlying data preprocessing, the data augmentation method, and the evaluation strategy.

Implementation: Introduce Dataset, dataset shortcomings, and the procedure used in the Background Theory chapter for data preprocessing and data augmentation. The final step is to explain how neural networks are implemented.

Evaluation and Results: This section displays the evaluation report to assess the output.

Discussion: The results of the evaluation are briefly discussed. Additionally, it defines future work, such as how to incorporate new data into the model and how many modifications the entire process would need.

Conclusion: Learn how this thesis addresses the issues, overcomes the difficulties, and ensures that the expected contributions are made.

2

Related Work

Discussion of previous work on conversational agents is presented in this section. There are two sections that focus on the recent work of computer scientists and educational scientists on conversation agents. The remainder of the section describes how the literature will help our study and what the primary differences are between it and ours.

2.1 Chatbots in Education

An early study on collaborative learning by Diziol, Dejana, et al. [13] demonstrates how an intelligent tutoring system can assist learners in offering support for learners' problems by examining learner interaction. Their research demonstrates how regular collaborative contact on a platform can improve learners' group output. The prior iteration of this research relied on fixed collaborative scripts, and there were a number of fixed collaborative script options available to support learners in interaction. For instance, a predetermined script in a reciprocal teacher helps students switch between the roles of instructor and pupil. The amount and organization of research on collaborative learning are growing rapidly. Recently, the European Commission addressed how to include and motivate learners to develop their skills through collaborative learning [9].

In higher education institutions, there is a significant drop-out rate in online courses as well as a general lack of motivation and engagement [9]. According to MOOCs [9], the issue arose due to a lack of commitment from teachers to students and students to teachers, as well as the long-standing practices used by an educational institution. In order to help teachers and students in an open online course, Stavros Demetriadis et al. [9] suggested integrating Conversation agents and screening techniques. By encouraging peer interaction in a particular collaborative group as well as in individual settings, conversation agents assist the student in becoming more involved in the online course. The project is generally referred to as "colMOOC" and is primarily an EC-sponsored initiative. One of the project's initial objectives is to assist humanities students in developing their digital proficiency in areas that have not yet been addressed by their studies. They create eLearning tools to support the learning process that offer instant feedback, track participant activity, and use gamification to encourage participants to engage in new activities. Students can explore various facts in pertinent studies with the conversational agent's "teacher" setting. The conversation agent has the ability to comprehend complex issues in order to offer workable solutions in addition to compatible and opposing points of view. Similar to this, one of the most recent research at Technische Universität Braunschweig, Strohmann, Timo et al. [44] designed BrAInstorm, which provides creativity stimulating content and encourages group idea generation. It aids collaborative learners in coming up with a variety of solutions to a given challenge while they are brainstorming ideas on a certain subject.

A chatbot solution for learners on an online platform has been offered by Neto, A.J.M., et al. [33], who also addressed the problem of collaborative learning. The chatbot aids students in collaborative learning and encourages learners to participate in group discussions. The proposed system enables the student to complete their essay, arbitrate class debates, plan activities, etc. The approach will generally motivate learners to make progress in accordance with their goals.. Similar to this study, Charuta Pande et al. [34] presented a hybrid conversational AI to help students with their assignments by assisting them in understanding the assignment as well as encouraging students for a partial solution. There are numerous drawbacks to working in groups, including low contribution rates, reliance on teammates, and poor quality outcomes. To address this problem, their conversational agent supports self-regulated learning in group tasks. Their proposed conversational agent is both 'task' and 'non-task' oriented. A task-oriented AI will assist in achieving a specific goal, whereas a non-task-oriented AI will simply carry out the conversation. The dialog management was split into two directions: "downwards" and "upwards." "Downward" concerns completing the task and breaking it down into smaller tasks like "how" and "where" related issues, for example, "how did you.." and "where is the.." In contrast, "upwards" justify the subtask such as a "Why" related issue, for example, "Why did you.." Reasoning and intent recognition were the two techniques used. When a keyword is discovered in a domain node during intent recognition, Al will comment with the relevant node. To accomplish that, CRFClassifier, an NLP library, was used to perform a Named Entity Recognition (NER). 'Ontology inferencing' is discussed in "Reasoning" to assist the system in identifying the subject, predicate, and object based on student comments. The task or subtask that needs to be solved is known as the Subject. The activity of a task or subtask is represented by the predicate. Objects represent a task or subtask's solution.

Mondal, Anupam et al. [32] created a chatbot for the education sector by using the ensemble learning method. The chatbot solely asks users for their questions and answers. Random forest with K-fold was utilized for the chatbot's validation. Another interesting work by Chopade, Pravin, et al. [6] using neural networks where project's primary objective is to forecast learners' performance in collaborative learning. The information was taken through eye tracking, text, audio, and video. The Hadoop data analysis platform was used to handle the data, and there, data clusters and clustered indexes were constructed. The researcher for this study then used convolutional neural networks to process and compute data clusters for machine learning-based evaluation.

2.2 *Other Chatbots*

Florian Peters [35] develops a chatbot solution for a betting company named 'GAMING1' in his master thesis at the University of Liège, Belgium. The datasets were extracted from Zendesk using an API crawler which included more than 1.4 million tickets related to customer support queries and answers. There are 9 different languages on those tickets, and 48% are in French. In addition to texts, the dataset contains information about the date, time, and user status of tickets, as well as the level of support. Initially, the French tickets dataset was fed directly to the neural network since it is the biggest dataset. To start the training, preprocessing was necessary, such as converting all uppercase to lowercase, filtering punctuation characters, tokenizing texts, etc. Four RNN variants have been proposed to be run on the dataset. From there, the Inverted GRU layer offers the highest accuracy out of four networks tested, including Single LSTM, Single GRU, Inverted GRU input, and GRU bilayer. The dataset supports nine different languages, so the main objective was to train a network that performs well across all of the languages. The main problem with working with different languages is that homographs between languages can hinder performance, making it impossible for the embedding layer to distinguish between two words from two different languages. To overcome this, they chose the first words because it would require less training time to get around this. Finally, a test dataset was used to evaluate each classifier. English classifiers are ineffective at understanding user needs and responses in chatbots because they only achieve accuracy rates of less than 71 percent, compared to more than 80 percent for French and Dutch classifiers. Similarly, ROC has been used for further evaluation, and once more, French and Dutch yield superior outcomes to English.

Manyu Dhyani et al. [11] used bidirectional RNN with an Attention model rather than RNN to create an open domain chatbot. To increase the input quantity to the network and handle unfixed input data, bidirectional RNN (BRNN) is used in place of RNN. The capacity of the model to remember longer sentence sequences is increased by BRNN. The Reddit Dataset [8] was selected to deploy the chatbot. The Reddit dataset mainly contains public comments on the Reddit platform from January 2015, and the dataset is uncompressed [8]. Training data is fed to a multiprocessing tokenizer to create the training model, which tokenizes comments based on space and punctuation. Each token will be considered a piece of vocabulary, and RedX, a regular expression tester, was used to create the vocabulary search pattern. Three evaluation matrics, including the bleu score, perplexity, and learning rate, were examined to assess the model. To determine how well a model fits the dataset, the bleu score compares a text from one language to another reference language. Perplexity concerns prediction error. The loss function is minimized by the learning rate. Aside from the low learning rate, the perplexity and bleu scores after training 23000 steps are 56.10 and 21.67, respectively. 23000 steps are necessary because the model completes an epoch at that point. An Epoch denotes a single forward and backward pass through an RNN of the training datasets. In this study, the main challenge is pre-processing the data set before creating the training sample, as the data set is relatively large with more than 10 million comment-response pairs and a size of 2.42 GB.

An RNN-based conversational agent was also developed by Silje Christensen et al. [21]. The primary goal is to examine the effectiveness of several RNN architectures, and the secondary goal is to examine whether the conversation agent is able to keep track of previous inquiries and responses. In the end, they created an RNN encoder-decoder model and used LSTM cells as the baseline. They used the Ubuntu Dialogue Corpus which is a text-based repository of technical information pertaining to the Ubuntu operating system. In order to preprocess the dataset, the researchers first extracted the question-answer pairs from the UDC datasets. Following that, they replaced all out-of-vocabulary (OOV) words with vocabulary words using the fastText library. Grid LSTM, LSTM, and GRU are the three variants of RNN used in this experiment. When comparing the training complexity of three different variants, LSTM and GRU models achieve almost similar scores, whereas Grid LSTM provides worse results. The fluctuation made it difficult to compare the validation perplexity score. For this purpose, they used exponential regression and noticed Grid LSTM provided a better perplexity score than LSTM and GRU while, in the training set, Grid LSTM received the lowest score. A human evaluation was also conducted on a single question from the UDC conversation. Due to not enough time reduction, GRU provides weaker results, whereas LSTM provides better results than the other two RNN variants. Finally, it appears that Grid LSTM cells have a low chance of getting better results because of the differences in training, validation, and human evaluation.

At Universidad Politécnica de Madrid, Álvaro Carrera et al. [5] created a chatbot application of cognitive computing for pedagogical question-answering that incorporates social dialogue. Students can use the application to learn Data Science related techniques. The chatbot used Speech Act Classifier to analyze the questions and select modules based on the responses. Research has been conducted on two modules, namely 'QA Module' and 'Small Talk Module'. 'QA Modules' concern when the user inquires about a specific thing, while 'Small Talk' concerns small words or sentences such as 'yes', 'I do', 'lol', etc. In addition to the two modules, the responses are categorized into 15 categories using 'Speech Text Classifier'. Multiple algorithms were compared, and Support Vector Machine (SVM) provided the highest accuracy and precision. For the evaluation, Partial Least Squares (PLS) analysis was used.

Disha is a chatbot solution developed by Md. Moshiur et al. [37] for Bengali speakers in the healthcare system that aids in disease diagnosis and alerts patients from health jeopardy. They used a dataset that included about 5000 samples spread across 41 diseases. At first, they only kept the positive features while getting rid of the negative ones. When a person has a particular disease, for example, it is positive; otherwise, it is negative. then changed the feature field's default value of "positive" to the precise name of the disease. In the end, they translated the text into Bengali from English. One of the main challenges when working with Bengali texts is that it is difficult to recognize keywords correctly as it is part of the Indo-Aryan language so they do not share the same alphabet with Germanic families like English, German, French, etc. Support vector machines (SVM), which perform better than other traditional classifiers, fit with the preprocessed dataset. Finally, the proposed system operates as follows. The method begins by collecting basic health information like age, blood type, etc. After that, filter the input by eliminating white spaces and punctuation marks. The keywords are then extracted from the input and used to build the 'typ' command by identifying the command input. Now, if "typ" is included in the disease classification, use the support vector machine, which compares features using a cosine similarity measurement before recommending a suitable solution.

Zijian Ding et al. [12] conducted one of the most thorough studies on conversation agents using a Support vector machine (SVM). To support the analysis of cognitive ability, they created TalkTive, a backchanneling ability-based agent that gathers speech input from adult participants. They include two categories of backchanneling, such as reactive backchanneling ("hmm" or "lol" e.g.) as well as proactive backchanneling ("please carry on"). Their findings indicate that people favor reactive over proactive backchanneling. Both SVM and neural network techniques were used to create the prediction model. The findings demonstrate that, for reactive backchanneling, SVM outperforms LSTM in terms of accuracy and F1 score during the training phase, despite LSTM having a higher Recall score. Additionally, the SVM score for proactive backchanneling in the validation and test set is adequate for creating the backchanneling-based agent with SVM.

Due to the small dataset and lack of stacked auto-encoders when pre-training the data with a neural network, SVM techniques appear to be superior in this case.

2.3 *Discussion (Related Works)*

The world of today is filled with digital information, so research on conversational agents will never become obsolete because it requires data, and data is growing every day. The majority of previous research on chatbots was centered on a one-way dialogue between the learner and the chatbot. Other interactions involve a chatbot and a group of learners, in which one learner interacts with the others to refine their skills with the aid of the chatbot. Although neither of the two scenarios is the main objective of this thesis but this thesis focuses more on regularizing learner-to-learner communication such that it never veers off course. Since this thesis is likewise based on natural language processing, the approach to dataset analysis, algorithm application, and evaluation procedure are similar even though the goals are different. When examining the most recent state-of-the-art studies, it was observed that the majority of researchers used Recurrent Neural Networks (RNN) and Support Vector Machines (SVM) in their work. The researcher who used SVM also used more established machine learning classifiers. SVM frequently surprises everyone in terms of accuracy. When it comes to machine learning, SVM is without a doubt the most accurate method. The main issue is that SVM does not perform well when working with large datasets or when there is a certain amount of noise in the data. Choosing the proper kernel for SVM is another issue, as SVM cannot forecast effectively if the dataset is unbalanced. Not only SVM, but these issues are also relevant for other traditional machine learning classifiers. Support vector machine technology was employed by Álvaro Carrera et al. [5] to construct a chatbot that encourages students to develop their data science skills. The chatbots are solely interested in nouns and verbs. Only nouns and verbs are searched to retrieve information from the text. The issue arises if debate centers on 'how' rather than 'what'. For instance, what is the most used supervised algorithm? is distinct from "how supervised algorithm can be used"? If the first question's response is 'linear regression' where the answer to the second question would be a method or process to carry out supervised learning. To solve this problem, RNN cells were introduced by Silje Christensen et al. [21], Manyu Dhyani et al. [11], and Florian Peters [35]. These researchers looked at various RNN cells to determine the best outcome.

Consequently, in order to get better results, it is, therefore, preferable to switch to neural networks. From the field of neural networks, RNN is the preferred method for creating a chatbot. It was noticed that the majority of recent papers use various RNN variants, including LSTM, Grid LSTM, GRU, and others, to determine which network is best for their results. As a result, for this thesis, neural network techniques have been used instead of traditional classifiers like support vector machine.

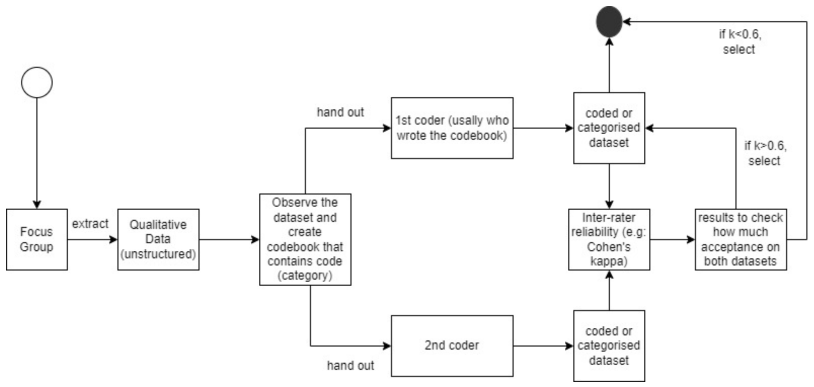
There would be a lot of data processing necessary to properly fit the data for training because the data is based on conversational text. The process of analyzing and processing data from prior study is fixed, and it is described in full in the chapters on 'Background Theory' and 'Implementation'. To increase the datasets, Sennrich et al. [38] and Wei et al. [45] developed data augmentation strategies. By doing that, it is feasible to increase the dataset because huge datasets enable more neural networks to successfully train. Confusion metrix [35], perplexity [21], bleu score [11] [21], and partial least squares [5] are utilized for evaluating the results. The confusion matrix method is the main emphasis of this thesis as a performance evaluation indicator. In natural language processing, the confusion matrix evaluates the model by determining how confidently it can predict text. As mentioned above, the chatbot will not produce new text on its own; rather, it will regularize the conversation between two learners by identifying texts that are nearly identical to one another among references. This is done with the aid of a rule-based approach, which will enable the chatbot to check if the responses refer to each other or not. This thesis gathered knowledge and information that can be used to implement the chatbot from the type of research indicated above.

3
Background Theory

Here is a discussion of the theoretical aspects that were used to assist the thesis's implementation and evaluation. It will assist the reader in acquiring the necessary knowledge by reading this part of the document, which will then help them understand the following part of this thesis paper. It opens with a description of the data collection technique used as well as a discussion of how we might analyze the dataset for the remainder of the technical part of this thesis. The purpose of this part is to provide the reader with brief details of the technical components used in the process of putting the overall thesis into action. A thorough explanation of how data augmentation works, how algorithms work, and how algorithm outputs can be evaluated is presented. At the end, the potential thread issue is discussed, as are the countermeasures that could be used against it.

3.1 Qualitative Data Collection

In qualitative research, non-numerical data are gathered to better understand people's perspectives, experiences, beliefs, and actions. There has been a lot of interest in this kind of research execution in recent years. The goal of qualitative research is to learn about a topic from different people's points of view by conducting in-depth interviews and holding focus groups. The data collected through



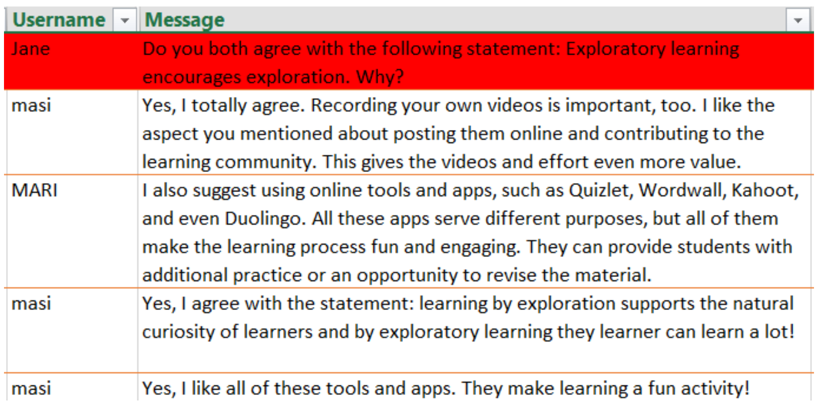
Focus Group
extract
Qualitative Data (unstructured)
Observe the dataset and create codebook that contains code (category)
hand out
1st coder (usally who wrote the codebook)
hand out
2nd coder
coded or categorised dataset
coded or categorised dataset
Inter-rater reliability (e.g: Cohen's kappa)
results to check how much acceptance on both datasets
if k>0.6, select
if k<0.6, select

Figure 3.1: Data collection in qualitative research.

qualitative research is more open to interpretation. When conducting qualitative research, a large sample size of participants is not required. Even with a smaller sample size, it is possible to reach a level of saturation in the datasets. When conducting research, one of the primary purposes of using a qualitative method is to increase the amount of information that can be gleaned from a limited number of participants. It is feasible to arrive at a completely unique viewpoint by utilizing qualitative research methods. Participants in qualitative research are encouraged to engage in in-depth thought, which helps researchers extract people's thoughts on how well they grasp certain concepts. Lab studies and field studies are the two types of studies that are used in qualitative research to acquire data. Both types of studies have their advantages and disadvantages. Studies in the laboratory are carried out using apparatus that is arranged in a controlled setting. Studies conducted in laboratories have high internal validity since the experimental design allows for the control of a variety of variables. The findings of the studies may be skewed if they do not have a high level of internal validity. Field studies are conducted in conditions or environments that are more natural or realistic, and this type of research has high external validity because the experiment is carried out in these more natural or realistic conditions. Studies conducted in a laboratory and those conducted in the field each have benefits and drawbacks that are unique to themselves; nevertheless, the selection of one study over the other depends not only on the objectives of the research but also on the resources that are readily available. Interviews, ethnography, and participation in focus group discussions are just a few of the qualitative research data collection methods [43]. This thesis is concerned with the use of focus groups as an approach to data collection. Therefore, the discussion that follows will only be about focus groups.

3.1.1 Focus Group

A focus group is a special kind of data collection in which a number of people meet face-to-face or electronically to talk about a specific subject. In response to an open-ended inquiry, participants are encouraged to share their thoughts, experiences, and opinions. The use of focus groups can be helpful in the generation of new ideas, as well as various attitudes, beliefs, and habits. The ability of more than one participant to provide their thorough opinions on a given topic is the primary benefit of a focus group. This results in a variety of responses to an open-ended inquiry. This indicates that one can obtain rich and detailed data from a focus group if one so chooses. The most significant drawback of using this strategy is the possibility that a person with strong opinions will be able to overwhelm a participant who is less outspoken. The results shown in the following examples were gathered from the focus group for this thesis.



Username	Message
Jane	Do you both agree with the following statement: Exploratory learning encourages exploration. Why?
masi	Yes, I totally agree. Recording your own videos is important, too. I like the aspect you mentioned about posting them online and contributing to the learning community. This gives the videos and effort even more value.
MARI	I also suggest using online tools and apps, such as Quizlet, Wordwall, Kahoot, and even Duolingo. All these apps serve different purposes, but all of them make the learning process fun and engaging. They can provide students with additional practice or an opportunity to revise the material.
masi	Yes, I agree with the statement: learning by exploration supports the natural curiosity of learners and by exploratory learning they learner can learn a lot!
masi	Yes, I like all of these tools and apps. They make learning a fun activity!

Figure 3.2: Data Collection via Excel.

3.1.2 Coding Approach

Since qualitative data are not structured, it is essential to organize them into appropriate categories. The practice of 'coding' qualitative data is one of the most frequent approaches to categorizing qualitative information. The 'open coding' methodology is the sole focus of this thesis. During open coding, researchers look over the dataset that was acquired from the focus group experiment and identify the concepts and categories that emerge from the datasets. Consider the following example:

I believe that humans can be more compassionate toward animals so that they can comprehend the anguish that the animal experiences during the slaughtering process. (veganism-discussion)

If we didn't eat meat, the world would be overrun with animals and I do not t think it is something we want to see happen. Therefore, I believe that more people should consume more meat. (veganism-discussion)

A significant number of people's lives were impacted by the Corona virus, and the illness had a more negative effect on the lives of those who were already struggling financially. (pandemic)

If we look at the first two examples, we can see that both statements are in direct opposition to one another; nonetheless, they are still included in the same category. This is due to the fact that both statements belong to the same category, despite the fact that they are in conflict with one another. When you are in the process of coding, it is essential to keep in mind that contradictory statements may fall under the same categories, even if they are about the same subject. The content in the third example is about the Coronavirus, which is why it is classified as belonging to a different category called pandemic. Similar to the above examples, it is essential to draw out the category based on the knowledge gained from the datasets. Following the completion of the categorization process, a code book should be compiled using the categories that have been identified, with the categories themselves being referred to as "code." Now we just need to assign those codes to each and every dataset, and we will finally have a structured dataset.

3.1.3 *Evaluation of Coding*

It is necessary to do an analysis of the coding that has been applied to the dataset; if this is not done, there may be a problem with bias. In order to accomplish this, we will need to draft an appropriate instruction for the code and determine its true meaning. It is necessary to give a different coder both the code and the guidelines because coding requires two different coders to complete. Before passing off the code to another coder, it ought to be properly discussed first. The second coder, depending on his or her comprehension, also assigned the code from the codebook to the dataset. Now we need to assess the inter-rater reliability of the coders, which checks, at its core, whether or not the two coders consistently give the same code (category) to the dataset. There are not many methods that are commonly used to measure inter-rater reliability. The Cohen's Kappa, the Fleiss's Kappa, and the Krippendorf's Alpha methods are some of the more well-known methodologies for determining inter-rater reliability between coders [15]. When it comes to determining the level of reliability between raters, the Cohen's Kappa test is one of the most frequent methods [30].

3.1.4 *Cohen's Kappa*

Cohen's Kappa: Cohen's Kappa compares how much two coders agree in a dataset to how much they would agree by chance. Cohen's Kappa [30] equation is,

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

Pr(a) stands for the number of times that both coders agree on a case. Pr(e) means the number of times the data is coded by chance. When the k value is less than 0.4, it means that the coders did not agree on much. If the number is between 0.6 and 0.8, it means that both coders agree to a moderate degree. If the number is higher than 0.8, it means that both coders agree strongly.

As a result, it is essential to conduct an analysis of the dataset in order to determine the degree to which the categories that have been assigned are appropriate. This is due to the fact that, in the event that the coding or categories that have been assigned are not accurate, it will thereafter be challenging to obtain satisfactory outcomes from the trained model. Because the thesis is concerned with computational approaches like the higher model, it is essential that the k value be higher than 0.6; otherwise, it will create computation complexity in the future. If the value of k is less than 0.6, it is suggested that the coding be done again and the more appropriate category or code be assigned.

3.2 Data Augmentation

We live in a world filled with data, and to build anything computationally, it requires data. Data plays the main role in machine learning. The more data fed into machine learning, the better the model's performance. Therefore, if the data is not enough or the machine learning model trains a smaller dataset with many complexity factors such as spelling errors, unfinished sentences, etc., the model can be overfitting, which means the model performs poorly on the testing dataset even though it performs well on the training data. To stop this problem, a data augmentation technique is introduced. When working with small datasets with limited variability, data augmentation can be useful. Data augmentation is nothing but a technique to increase the size of datasets. It can be done by creating slightly modified copies of the original data. That's how a dataset can be enlarged without repeating the same data index over and over again. Data augmentation helps to prevent overfitting because, with the help of data augmentation, the model is exposed to more diverse examples. There are a few techniques in data augmentation that help make the dataset larger and more balanced. The following are the data augmentation techniques that have been used in this thesis to implement the product:

3.2.1 Spelling Augmentation

Spelling augmentation is known as SpellingAug, which is a data augmentation technique specifically for text data. SpellingAug helps to increase the variability of training data by replacing words with misspelled words. SpellingAug uses a dictionary that contains misspelled words and their corresponding correct spellings. This dictionary can be obtained based on the requirements of the task. Some of the approaches are domain-specific corpora, which are domain-based (e.g., educational data), manual creation, etc. The book "Common Errors in English Usage' by Paul Brians can also be used as a resource to create a dictionary [4]. SpellingAug selects words randomly in the training datasets; for example, SpellingAug selects a word and checks if the word is in the dictionary or not. If it is in a dictionary, it replaces the word with the corresponding misspelled word. It cannot replace all words because some words may not be found in the dictionary. It only replaces those words that are found in the dictionary. Also, it is important to add the probability of replacement in order to assess the likelihood that any particular word in the input text will be replaced with an incorrectly spelled version taken from the dictionary. If the probability of replacement is set to high, the augmented text would be difficult to understand. Therefore, it is always possible to control the SpellingAug augmentor by setting up the replacement probability. In general, there is no standard value for this probability. It always depends on the specific task and the amount of variation in the dataset. Examples from the original datasets for spelling augmentation are given below.

Example 1:
Original Text: Effective feedback is clear, specific, and concrete.
Augmented Text: Effective feedback is clear, specific, and concreat.

Example 2:
Original Text: Blended learning will be more resourceful and helping in times.
Augmented Text: Blended learning will be more resourcefull and helping in times.

In example 1, spell 'concrete' and replace it with the mispelled word 'concreat'. Similarly, for example 2, it is noticeable how 'resourceful' becomes "resourcefull' in the augmented text. Now, the question would be, 'Why is it necessary to make a slightly modified sentence or text with a misspelled word?' It is important for a few reasons. First of all, it increases the data. Second, it introduces text variation and provides flexibility to users, as an example of chatbot users. Finally, it helps a natural language processing-based model reduce overfitting.

3.2.2 *Insertion Augmentation*

This technique involves inserting a new phrase into a specific existing text from a dataset. Similarly to spelling augmentation, insertion augmentation also has some common reasons for being used. First, it increases the data volume. Second, along with increasing the data volume, it adds a new phrase to a sentence or text without changing the semantic meaning of the sentence or text. Third, because of insertion augmentation, the dataset provides variation, which helps to improve the robustness of the natural language-based model as well as provide diversity. Diversity because it is possible to identify the same text or sentence with different phrases. Finally, if the target domain is sports-related, it can add common phrases that are used frequently in sports. Therefore, the model can accurately predict the domain as well as learn more patterns in the dataset. These phrases can be generated randomly or selected from a predefined list. Random generation is mainly based on the random generation approach and the rule-based generation approach. When a phrase is selected from a predefined list, it is necessary to check the list to see if it is training data domain-related or not. An example from the original datasets for insertion augmentation is given below.

Original Text: Blended learning will be more resourceful and helping in times.
Augmented Text1: Blended learning will be more productive and resourceful and helping in times.
Augmented Text2: Blended learning will be more efficient and resourceful and helping in times.
Augmented Text3: Blended learning will be more resourceful and helping in

busy times.

In this example, in augmented text 1, the 'productive' phrase is added, and the 'efficient' phrase is added for augmented text 2. In both cases, the semantic meaning is the same, but for augmented text 3, 'busy' was added before time. In practice, the 'busy' word does not go with the sentence structure, but it would still cause no harm to the dataset since the semantic meaning of the sentence is still the same. Overall, this augmentation technique is so powerful that it may also change the semantic meaning of a text or sentence; therefore, it is important to always generate domain-based phrases or select phrases from a domain-based predefined list.

3.2.3 Substitute and Synonym Augmentation

Substitute augmentation is almost like spelling augmentation. The main difference is that spelling augmentation replaces a word with its corresponding misspelled word, and substitute augmentation replaces a word with a word with similar meaning. These similar words are obtained from a dictionary, which contains a list of words and their synonyms. There are few resources available for this, and 'WordNet' is one of them. WordNet is a database of words that groups words into sets of synonyms called sysnets [ref]. For example, words like 'resourceful' can also be defined as 'clever', 'creative', etc. based on WordNet. The main advantage of using substitute augmentation is that it creates diversity because replacing a word with its similar word, which expresses the same meaning, may vary the meaning, but the symbolic meaning is the same. It gives the opportunity to express a sentence or a text in a different way by changing only a word. Overall, it helps to improve the model's accuracy and helps the model learn at different levels of complexity. An example is the following:

Original Text: Blended learning will be more resourceful and helping in times.
Augmented Text1: Blended learning will be more creative and helping in times.
Augmented Text2: Blended education will be more resourceful and helping in times.
Augmented Text3: Blended learning will be more resourceful and aiding in times.

From the above examples, it is noticeable how 'resourceful', 'education," and 'helping' words are substituted with their respective similar words from the WordNet database. Therefore, one specific sentence and three new sentences are created by substituting words without changing the meaning of the sentence. Substitute augmentation is similar to synonym augmentation; the only difference is that substitute augmentation is done via similar words that can be synonyms or not. For example, in words like 'cat', both 'animal' and 'kitty' can be used, but synonym augmentation only concerns the synonyms of the word, for example, 'kitty' for cat.

3.2.4 *Other Augmentation*

There are also additional augmentation methods available. Data augmentation techniques such as back translation are also common in natural language processing. It involves translating a sentence from an original dataset into a different language and then retranslating it again into the original language. If the dataset is large, it takes more time to run augmentation techniques because, overall, backtranslation is computationally expensive because, to run backtranslation, more processing power, memory, and time are required. Sometimes, if a sentence is long, it may lose meaning and accuracy during the back-translation process. Also, sometimes a grammatical error could be retranslated into a sentence that would have a different meaning. An example follows:

Original Text: He must educate himself about it first to use it effectively and combine it with pedagogy.
German Text (Transalation): Er muss sich erst darüber informieren, um sie effektiv einsetzen und mit Pädagogik zu verbinden.
Retranslate (Augmented Text): He must first inform himself about them in order to use them effectively and to connect them with pedagogy.

The original text concerns specific things by mentioning 'it', but the retranslated text makes it "a group of people" by mentioning 'them," as well as combine and connect, which are two different words. This would be one of the drawbacks of the back translation method but the benefit of back translation is that, along with increasing data volume, it helps expose a wider range of linguistic patterns. There are also other data augmentation techniques available, but those are not frequently used and popular in natural language processing, such as sentence shuffling, masking, domain adaptation, etc., and recourses for this technique are also few.

3.3 *Natural Language Processing*

Natural Language Processing (NLP) employs statistical analysis and various forms of machine learning to assist individuals in better comprehending both human language and language generated by computers. One is able to perform tasks such as text classification, speech recognition, analysis of sentiment, and many others with the assistance of natural language processing. We use products that are based on natural language processing every day, such as smartphone speakers like Siri and Google Assistant, voice-controlled cars, translators like Google Translate and Yandex Translate, and different kinds of chatbots, among other things. Since this thesis is about text classification, the next section goes into more depth about text classification in natural language processing.

3.3.1 Text classification

The practice of organizing various aspects of a text into hierarchical groups is known as text classification. This includes organizing news into categories (such as politics and sports), conducting text analysis based on the opinions of users, and categorizing content according to the subject matter of the text. Rule-based systems, machine systems, and hybrid systems are the main types of text classification methods.

a) Rule-Based Approach

Texts are categorized into different groups using the rule-based approach by applying handcrafted linguistic criteria. The users themselves define a list of terms or keywords that are included in the handicraft linguistic guidelines. Words like Joe Biden and Xi Jinping, for instance, would be classified under the category of politics, whereas words like Leo Messi and Cristiano Ronaldo would be classified under the category of sports. If we want to get more particular with our categorization, we might put Leo Messi and Cristiano Ronaldo in the football category rather than the sports category if our data contains numerous texts that are linked to sports. Imagine we have a dataset that contains texts that are relevant to politics and sports. In the beginning, the essential data processing would be done, such as cleaning the noise in the data, fixing the spelling, removing stop words, tokenizing, and other similar tasks. Following that, two vocabularies called 'politics vocabulary' and 'sports vocabulary' would be made.

'Politics vocabulary' is made up of the most common words about politics, and 'sports vocabulary' is made up of the most common words about sports. For example, words like club, football, soccer, championship, etc. are in the sports vocabulary, and words like trump, election, minister, etc. are in the politics vocabulary. Then, two vectorizers, named politics and sports, would be made. Vectorizer comes from the word "vectorization," and in practice, natural language processing maps words like 'Trump', 'election', and 'football' to a vector of real numbers instead of words that can be used to guess words and find words that are similar to each other. We have two vectorizers called 'politics' and 'sports'. Now it is necessary to set up a threshold score. A threshold score refers to a parameter or a configuration setting.

For example, if the score is defined as 0.7, it means that a statement must score at least 0.7 in both the politics and sports vectors for it to be considered a sentence about sports or politics. There is no exact way to describe this threshold score, but we have to keep in mind how rich our data is when defining it. The higher the threshold score, the more likely it is that we will get good classified texts. However, if, for example, a model based on natural language processing gives bad results, this threshold score must be adjusted. Following that, rules would be created to find vectorizer-created texts about politics or sports. The word "rule" refers to a set of criteria for putting text into groups based on certain traits. In this case, a text can be labeled as a sports text if the sports vector has a high score for it in the dataset. The same goes for political related texts. As a result, a rule is used to decide whether a sentence or a text is about sports or politics. It is necessary to ensure the texts meet the required threshold score to be classified as sports and political texts. It is possible to determine how well the system works by utilizing evaluation metrics such as accuracy, precision, recall, and others.

If the outcome is not satisfactory, it is important to apply fine-tune rules-based evaluation, which increases both the precision and the recall of the results. For instance, if the system predicts or incorrectly classifies certain texts as being about sports when they are actually about politics, we have the option of either decreasing the frequency of words related to politics in the texts that are misclassifying or, as an alternative, increasing the frequency of phrases linked to sports in the texts that are incorrectly classifying. In the end, a system that can predict political and sports-related texts will be deployed. This system could be a chatbot or a website that collects data for marketing, sales, etc.

b) Machine System

It is also referred to as a machine-based classifier. Here, texts are already tagged or pre-labeled. If we take the dataset from the previous example, the labels for sports and politics would be defined in this approach from the beginning in the dataset. As like rule-based systems, the essential data processing would be performed initially. Since labels are predefined in this approach, feature extraction would be performed using the 'Bag of Words' technique, which serves text data as a vector of word frequencies. The example is given below,

Label: *sports*
Text1: *premier league is the best.*
Text2: *messi never played in premier league.*
Text3: *ronaldo played in premier league.*

There are three sentences here that must be represented as Bag of words vectors. First, the frequency of the words in this text that can be considered vocabulary needs to be calculated.

Text1: *'premier': 1, 'league': 1, 'is': 1, 'the': 1, 'best': 1*
Vocabulary: *'premier': 1, 'league': 1, 'is': 1, 'the': 1, 'best': 1, 'messi': 0, 'never':0, 'played': 0, 'in': 0, 'ronaldo':0*
Text2: *'messi': 1, 'never':1, 'played': 1, 'in': 1, 'premier': 1, 'league': 1*
Updated Vocabulary1: *'premier': 2, 'league': 2, 'is': 1, 'the': 1, 'best': 1, 'messi': 1, 'never':1, 'played': 1, 'in': 1, 'ronaldo':0*

Text2: *ronaldo: 1, 'played': 1, 'in': 1, 'premier': 1, 'league': 1*
Updated Vocabulary2: 'premier': 3, 'league': 3, 'is': 1, 'the': 1, 'best': 1, 'messi': 1, 'never':1, 'played': 2, 'in': 2, 'ronaldo':1

Now, depending on the position of each word, an index is allocated to each word. When designating an index to each word, it is necessary to remember the frequency from updated vocabulary2, e.g., the index number for 'is' is 2, whereas the index number for 'in' is 1, even though both begin with 'i'. This is because 'in' is more common than 'is'. The following is the word order and new updated vocabulary dictionary with index number:

Order: [best in is league messi never premier played ronaldo the]
Updated Vocabulary: 'premier': 6, 'league': 3, 'is': 2, 'the': 9, 'best': 0, 'messi': 4, 'never':5, 'played': 7, 'in': 1, 'ronaldo':8

Following that, text1, text2, and text3 must be transformed into a Bag of Word vector, which yields a sparse matrix that looks like this:

Term: [best, in, is, league, messi, never, premier, played, ronaldo, the]
Text1: [1, 0, 1, 1, 0, 0, 1, 0, 0, 1]
Text2: [0, 1, 0, 1, 1, 1, 1, 1, 0, 0]
Text3: [0, 1, 0, 1, 0, 0, 1, 1, 1, 0]

Next, the term frequency would be calculated after obtaining the sparse matrix. The equation for the sparse matrix [26] is,

$$Term\ Frequency = \frac{Total\ number\ of\ words\ appearing\ in\ a\ document}{Total\ number\ of\ words\ in\ the\ document}$$

The term frequency matrix in array looks like the following.
Term: [best, in, is, league, messi, never, premier, played, ronaldo, the]
Text1: [1/5, 0, 1/5, 1/5, 0, 0, 1/5, 0, 0, 1/5]
Text2: [0, 1/6, 0/6, 1/6, 1/6, 1/6, 1/6, 1/6, 0, 0]
Text3: [0, 1/5, 0, 1/5, 0, 0, 1/5, 1/5, 1/5, 0]

The above matrix is known as the term frequency (TF). Following that, the inverse document frequency (IDF) must be computed. There are three documents in this example: text1, text2, and text3. Inverse Document Frequency (IDF) [26] looks as follows ,

$$IDF = log\frac{Total\ number\ of\ documents(texts)\ in\ the\ corpus}{number\ of\ documentscontaining\ that\ word}$$

The Inverse Document Frequency (IDF) in array looks like the following.
Term: [best, in, is, league, messi, never, premier, played, ronaldo, the]
IDF = [log(3/1), log(3/2), log(3/1), log(3/3), log(3/1), log(3/1), log(3/3), log(3/2), log(3/1), log(3/1)]

The Inverse Document Frequency (IDF) can be replicated as 4x4 matrix. Since the Inverse Document Frequency (IDF) matrix is a row vector, matrix broadcasting must be used to replicate it so that it matches the Term Frequency (TF) matrix's dimensions. The following step is to compute the Inverse Term Frequency-Document Frequency (TF-IDF). Inverse Term Frequency-Document Frequency (TF-IDF) is an algorithm that looks at how often words appear in a document to figure out how important those words are. The equation [26] looks like follows,

$$TF - IDF = Term\ Frequency * Inverse\ Document\ Frequency$$

Then, a new matrix named Term Frequency-Document Frequency (TF-IDF) would be constructed. After that, an algorithm or classifier would be trained on the Inverse Term Frequency-Document Frequency (TF-IDF) matrix in order to classify the output document or texts according to the textual content. Inverse Term Frequency-Document Frequency (TF-IDF) can be used as input features for a machine learning model. The values in each row of the matrix correspond to the TF-IDF scores of the associated terms in the document or text represented by each row of the matrix. As with the rule-based approach, an evaluation would be conducted by analyzing evaluation metrics, and in the event of an unsatisfactory outcome, appropriate tuning would be performed.

c) Hybrid System

A hybrid system in natural language processing combines a rule-based approach and a machine system. In this method, the tag or label for text data is created first, followed by the definition of rules using a rule-based method. As described previously, a rule consists of a condition and an action. Another example:

Condition: If the texts contain the words "freekicks" and 'championship'
Action: Classify the text as 'sports'

Finally, the rules from the rule-based system are compared to the machine-based system's rules. The machine system has predetermined rules from the start. If something fails the comparison, it must be improved before being fed to the machine learning model. Since it compares both rule-based and machine-based systems, the hybrid system is more useful and accurate for obtaining good performance from machine learning models than the other two approaches.

3.3.2 *Word Embedding*

This paper formerly discussed various approaches to text classification in natural language processing. When using those methods to feed a machine learning model, it is essential to bear in mind that typical machine learning models are unable to analyze text data. This is due to the fact that text data contains words that are difficult for the algorithm to comprehend because there is no numerical representation of these words in the text. Word embedding is required because of this. Word2vec and Doc2vec are two well-liked word embedding tools.

a) Word2Vec

Each word in the text is translated into a numerical form (a vector) by Word2vec. Each word is thus assigned to a vector. A word's vector capture feature in relation to the text as a whole. Consider the following instance:

'I am studying psychology.' In the text, each word represents as vector like below:

I = [1 0 0 0]
am = [0 1 0 0]
studying = [0 0 1 0]
psychology = [0 0 0 1]

This assignment is based on position. The 'studying' position is 3 in that text, which is why it is represented as [0 0 1 0]. There are two ways that the word2vec model works, such as Continuous Bag of Word (CBOW) and Skip-Gram.

1a. Continuous Bag of Word (CBOW)

It is used for tasks in natural language processing that involve making predictions about the target word based on the word in its context. If we are to follow the preceding example, after each word has been represented as a vector, these vectors can be fed into the continuous bag of words (CBOW) model in order to learn the word embedding. In the output of the continuous bag of words (CBOW) model, a dense vector represents each word. Words that have meanings that are comparable to one another are located in close proximity to one another on a dense vector, which displays the meanings of words in a space that is not binary. An example of dense vector is following:

'I' -> [0.4, 0.1, -0.2, -0.3]
'am' -> [0.2, -0.4, 0.5, -0.1]
'studying' -> [0.3, 0.2, -0.4, 0.1]
'psychology' -> [-0.1, 0.3, -0.2, 0.1]

The representation of the original sentence, 'I am studying psychology', is *[[0.4, 0.1, -0.2, -0.3], [0.2, -0.4, 0.5, -0.1], [0.3, 0.2, -0.4, 0.1], [-0.1, 0.3, -0.2, 0.1]]*

This vector represents the semantic meaning of each word in that text ('I am studying psychology"). Now, if the target word is 'am', the input would be the context vector [0.7, 0.3, -0.6, 0.2], which is some of the context vectors ‘i’ and ‘studying’, and for the target word ’am’, the output would be [0.2, -0.4, 0.5, -0.1].

This is how it is possible to predict the vector representation of a new sentence. Simply put, context words are summed together to create a context vector to predict the target work. ‘Continuous Bag of Word (CBOW)’ has an input layer, a hidden layer, and an output layer [ref]. The context vector is provided by the input layer, the hidden layer learns the pattern in which the actual word embeddings are learned, and the output layer generates a probability distribution across the corpus’s vocabulary. Vocabulary is described as it is. How this layer works is described briefly in the algorithm section. The CBOW works well with large corpuses of text data; the more data in the corpus, the better it can predict the output word.

2a) Skip-Gram

Skip-Gram is another popular approach to word embedding. From the above discussion, it is interesting to note that the Continuous Bag of Words (CBOW) model makes predictions about the target word based on the words that are included in the context, whereas the Skip-gram method does the opposite. The input for Skip-Gram is the target words, and the output is the context words that it predicts based on those target words. The choice of using Continuous Bag of Words (CBOW) and Skip-Gram is based on datasets and specific tasks. Skip-gram is mainly used when we have large amounts of data as well as data containing less frequent words, while Continuous Bag of Words (CBOW) is best suited for small datasets with more frequent words. An example is given below to understand how skip-gram words.

Source: <u>The</u> latest pink plane flies over the famous pyramid.
(The, latest) (The, pink)

Source: The <u>latest</u> pink plane flies over the famous pyramid.
(latest, the) (latest, pink) (latest, plane)

Source: The latest <u>pink</u> plane flies over the famous pyramid.
(pink, the) (pink latest) (pink, plane) (pink flies)

Source: The latest pink <u>plane</u> flies over the famous pyramid.
(plane, latest) (plane, pink) (plane, files) (plane, flies)

For this example, assume that the window size w = 2, which means it concerns two words before and after the input words as well as the input word too. The example text is ‘The latest pink plance files over the famous pyramid’. If we take the word ‘the’ as our context word because it is the first word in the text, we make pairs of this word with words on its right and left. Since w = 2, only two words from the right and left of the word 'the' are concerned with making a pair. In this case, there is no word on the left side for the context word 'the'. Therefore, there are two pairs, such as (the latest) and (the pink). Similarly, if we take the word 'latest' as a context word, we would get three pairs such as (latest, the), (latest, pink), and (latest, plane).

Thats how, it keeps matching the context word with the target word for all possible combinations of context and target words. The next step is to apply one-hot encoding to the context word. One-hot encoding refers to the process of representing category data as a binary vector, similar to the example given in Continuous Bag of Words (CBOW). This is done in order to input it into the neural network, which ultimately predicts the degree of similarity. The approach known as 'Continuous Bag of Word (CBOW)' is composed of three layers: an input layer, a hidden layer, and an output layer. In contrast to the Continuous Bag of Words (CBOW) algorithm, in which the hidden layer represents the vector sum of the input word embedding, the hidden layers in the Skip-Gram algorithm reflect the embedding of the target words. Therefore, for Skip-Gram, the input layers contain the one-hot encoded vector that represents the target word. The context words that are most likely to appear surrounding the target words are predicted by the output layer's probability distribution.

b)Doc2Vec

Doc2Vec is another popular way to do word embedding. Doc2Vec build based on Word2Vec techniques. Only difference is that along with word vectors like Word2Vec, Doc2Vec add another vector in the input named paragraph ID. The following is the architecture of the Doc2Vec. The architect is based on the Continuous Bag of Word

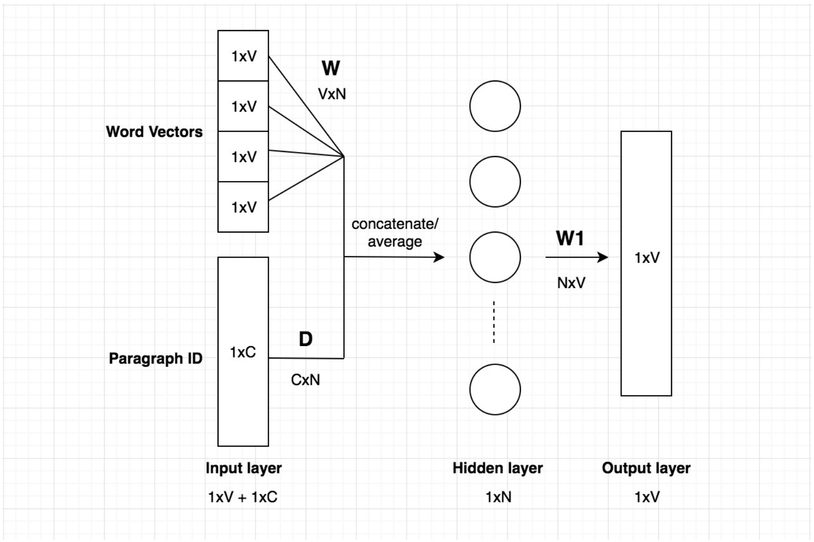


Figure 3.3: Architecture of Doc2vec model [14].

(CBOW) model; the only difference in the input layer is paragraph ID, which means when training the word vectors, the paragraph ID, which is basically a document vector, is also trained.

Take the case where we want to group together texts that are similar in terms of sports. In this case, we can use Doc2Vec to create a vector representation for each text and then use these vectors to group the texts according to how similar they are in terms of semantics. To generate the document vector, first represent each document in the datasets as a sequence of words. Then, a neural network would be used to learn how each text is represented by a vector. The network predicts the target word in the context of the document using an input of a document ID and a string of words. In order to assist the network in learning a distinct vector representation for each document, the document ID is employed as an additional input.

3.4 *Artificial Neural Network*

Massive parallelism, distributed representation, computing, and learning activity are only few of the desirable traits possessed by the human brain that are lacking in today's parallel computers [17]. The human brain also possesses many more desirable characteristics. On the basis of this, artificial neural networks were developed. Mathematical models that are based on biological neural networks serve as the basis for artificial neural networks. These networks, which are composed of neurons that process information, are inspired by biological neural networks[22].

In people, sensory processing is the way that information from the eyes, ears, tongue, skin, and other organs that sense things is sent to the brain. Sensory information is first picked up by specialized neurons in the organs of sense. This information is then sent to the brain through a network of synapses and neural paths. As sensory information comes into the brain, it is processed by the neural networks in the brain's specialized areas, such as the visual cortex or the aural cortex, which are in charge of figuring out what the information means.

In Figure 3.4, light is detected by the eye, which converts it into electrical signals that are transmitted to the brain. The electrical signal is generated after sensing the light via the eye In the process section, the brain interprets where the visual cortex processes the electrical signals to use them to identify objects. For example, we want to recognize a specific image of an animal from a group of images of an animal. The input to the network would be a specific image that we want to recognize, which would be processed by the network in a series of layers to produce an output representing the recognized image of the animal.

The first layer of the artificial neural network is comprised of input neurons that receive the image of a particular animal to be recognized.

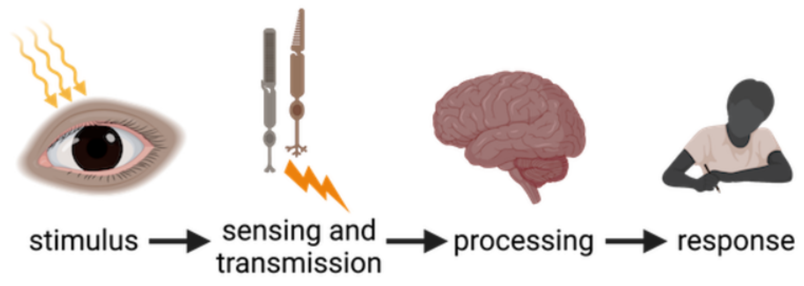


Figure 3.4: Sensory processing and brain via sensory organs [1].

The aforementioned neurons are presumed to establish connections with neurons in the subsequent layer, which would undertake the task of processing rudimentary characteristics of the image. Subsequently, these neurons would establish connections with neurons in the ensuing layer, which would be accountable for the identification of more intricate features. This is how the neural network model knows the input image is related to other images and learns to recognize similar features, and finally, the outcome of the artificial neural network would recognize the animal in that input image. Following is a graph of an artificial neural network. In artificial neural networks, there

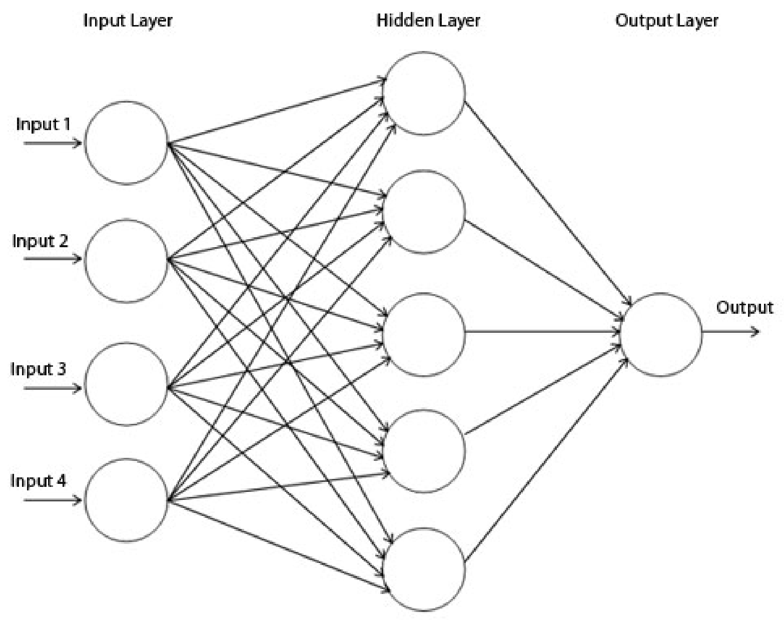


Figure 3.5: Artificial Neural Network [29].

are three layers of neurons: the input layer, the hidden layer, and the output layer. The input layer is the layer that receives data. The data that constitutes the input is received by the input layer, and it is then sent on to the following layer, which is known as the hidden layer. The information is processed in the hidden layer. There may be more than one hidden layer, and each layer of the hidden layer consists of a collection of neurons that are connected to the layer below it and the one above it. These connections can run in both directions. These connections are able to operate in either direction. The final layer is called the output layer, and it is this layer's job to produce the actual result. In addition to the input layer, the hidden layer, and the output layer, the following is a list of typical terminology that is utilized while speaking about artificial neural networks.

3.4.1 *Activation Function*

Activation functions are important components of artificial neural networks because of the role they play in facilitating the comprehension of complicated, non-linear, and learning input-to-output mappings [40]. There are many activation functions, however, the ones that will be discussed here are some of the ones that are frequently employed in research that is based on neural networks in terms of this research's primary purpose:

a) Linear Activation Function

The output of a linear activation function is precisely proportional to the input values since the function starts with a weighted sum of those values and then applies a linear transformation to them [40]. The equation [40] is,

$$F(x) = ax$$

In the equation, 'a' is the user-selected constant. The input is x, and the output is F(x). The graph of linear activation function is shown below. The linear activation function is usually used in regression

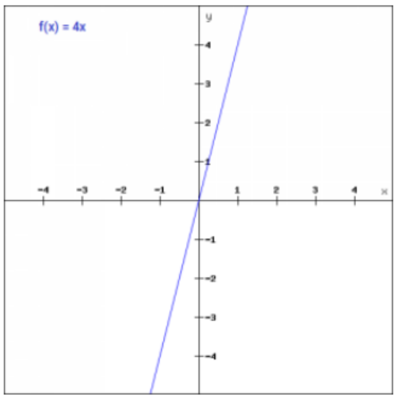


Figure 3.6: Linear Activation Function [40].

because it outputs a continuous variable. This activation function is unsuitable for classification. Thus, it is unsuitable for predicting binary outcomes like'yes' or 'no' or 1/0.

b) Non-Linear Activation Function

The employment of an activation function that is non-linear in nature allows for the introduction of non-linearity into the model. This helps the neural network model to learn complicated correlations between the variables that are input and output into the model. There are a few different varieties of it, some of which are described in the following paragraphs.

1b) Sigmoid Function

The sigmoid function is popular in binary classification problems because it generates an S-shaped curve. The sigmoid function transforms the value range to 0 to 1 [40]. For example, if x is 0.35, which is

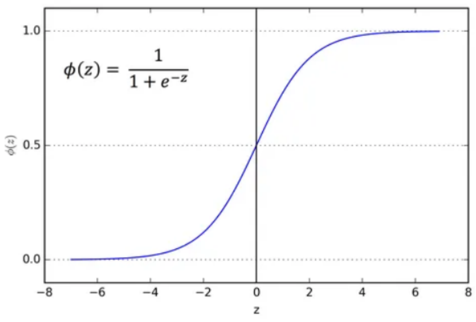


Figure 3.7: Sigmoid Function [40].

the input value, then the sigmoid (0.35) is 0.5866. This value indicates that if the given input is 0.35, then the model outputs a probability of 0.5866 for a positive outcome (in the case of 1 and 0, it would be 1).

2b) Tanh Activation function

Tanh activation function is a better version of the sigmoid function, and the range is from -1 to 1. The input that is located in the negative region will be heavily mapped with the negative value, and the value zero in the input will be mapped close to the zero value [40]. The equation of the Tanh activation [40] function is as follows :

$$f(x) = 2sigmoid(2x) - 1$$

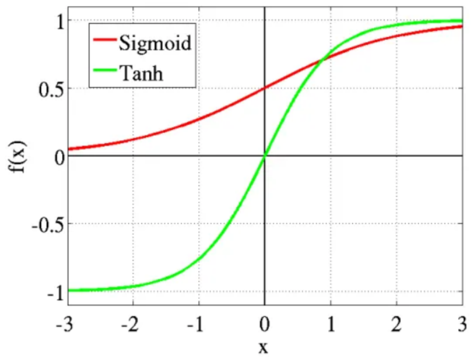


Figure 3.8: Tanh Activation Function [40].

The gradient for the sigmoid activation function is small since it is saturated to 0 or 1, which leads to a vanishing gradient problem. A gradient indicates the rate of change of a function and In the vanishing gradient problem. When the gradients of the loss function with respect to the weights in the lower layers get progressively small, it becomes increasingly difficult for the network to properly learn and update those weights. This is due to the fact that the gradients of the loss function are decreasing. On the other hand, Tanh functions saturate to -1 or 1, so it is a slightly better choice. .

3b) ReLu Activation Function

This is the most commonly used activation function in the world right now [39]. The equation is f(x) = max(0, 1) [40]. If the input value is positive, it returns the same value as input; otherwise, it returns 0. For example, if the input value is x = 3, the function returns f(x) = max (0, 3) = 3. Similarly, if x = -4, the function returns f(x) = max(0, x) = max(0, -4) = 0. Therefore, this is an issue of dying ReLu because it produces 0 if the input is negative, and because of that, a neuron cannot contribute to its learning process and becomes inactive for the remainder of the training process. To fix this issue, leaky relu is

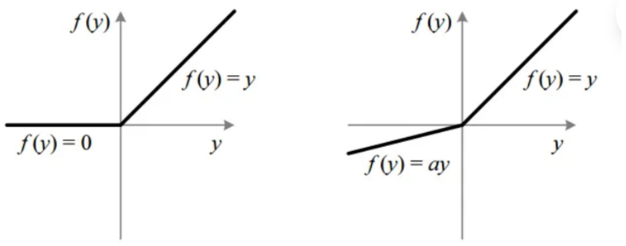


Figure 3.9: ReLu and Leaky ReLu [39].

introduced, which allows negative inputs and thus avoids the dying relu problem. The equation of Leaky ReLu [49] is,

$$f(x) = max(\alpha x, x)$$

Here is in between (0,1), which is α predefined constant that ensures that neurons can continue to learn by becoming active.

3.4.2 *Regularisation in Neural Network*

The two most common terms in machine learning are overfitting and underfitting. If the model performs poorly in the training data, such as by being unable to capture the relationship between input and output and producing very low accuracy, it is called underfitting. Overfitting describes the situation in which a model performs exceptionally well on the data it was trained on but not as well on the data it was tested or validated with. Therefore, to prevent the overfitting of the model, an early stopping approach is applied, which basically stops the training processes before the model becomes overfitting. By specifying the number of epochs, it is possible to stop the training before it becomes too tiring. Epoch is an iteration of the training dataset. For instance, Epoch = 3 indicates that the neural network model has completed two entire training dataset iterations. Another common regularization technique is known as dropout, and it is also used to prevent overfitting. In neural network approaches, sometimes a single neuron has more influence on output data than the other neuron, which causes overfitting. Dropout prevents this input from a single neuron from affecting the output data by arbitrarily eliminating a portion of the neurons during training.

3.4.3 Recurrent Neural Network: Long short-term memory

This next section is about recurrent neural networks (RNN), which are a class of artificial neural networks. This thesis is heavily based on recurrent neural networks (RNNs). Therefore, this section is more theoretically focused on recurrent neural networks and their different types of variants that are used to train the dataset.

Traditional machine learning algorithms such as support vector machines and Naive Bayes can be used in text classification. The implementation of these traditional algorithms usually necessitates expertise in the relevant field to engineer the features. In the context of text classification, it may be necessary to devise a feature that denotes the existence or non-existence of specific keywords that serve as indicators for each respective class or category in the text data. The process of designing these features necessitates specialized knowledge in the relevant field, including familiarity with the vocabulary and syntax of the language, which can be time-consuming and expensive when dealing with complex domains.

That's the reason the neural network was introduced. Neural networks automatically learn patterns from text data without the help of manual engineering. A neural network also works well in complex domains. When it comes to text classification, recurrent neural networks are popular in natural language processing tasks [28].

Long short-term memory (LSTM) is one of the types of a recurrent neural network. Long-short-term memory (LSTM) helps to overcome the issue of long-term dependencies. Long-term dependencies are common in natural language processing tasks. It is referring to the relationship that exists between words that are located quite far apart in a sentence. Consider the following example:

I was born in Asia; therefore, Asian food is my favorite.

In this example, the words 'asia' and 'favorite' are a few words apart, but the two words are semantically related in this sentence. This type of long-term dependence cannot be captured by a traditional machine learning classifier because it cannot fix the length of the input representation. Because of that, the relationship between the words 'asia' and 'food' can be lost. Long-short-term memory (LSTM) helps mitigate this issue by enabling the information to move through the network for longer periods of time. The following is the figure of the long-short-term memory (LSTM) building block. This long short-term memory (LSTM) takes three inputs. As a result, this single unit decides based on the present input, which is x_t, the prior output, h_{t-1}, and the previous memory, c_{t-1} [47]. Additionally, it creates a new output and modifies its own preceding memory, c_{t-1}. Now the input is the c_{t-1}.

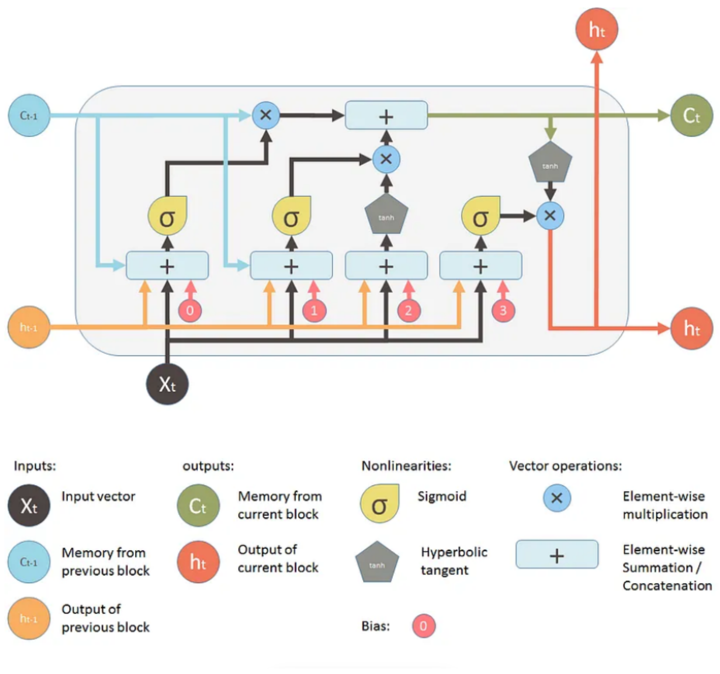


Figure 3.10: LSTM Architecture [47].

'X' operator, which works as a forget gate in the figure, helps to forget c_{t-1} or go through with c_{t-1}. '+' operator indicates the merge of c_{t-1} and c_t where c_{t1} is new memory. The information that is allowed into the memory cell by the input gate and that is allowed to remain there over time by the forget gate is stored in the memory cell. An LSTM cell's input gate is responsible for selecting which bits of information from the input are stored in the cell's memory. A sigmoid function is applied to the data, which consists of the most recent hidden state as well as the current input vector, x_t. Finally, the output gate uses the current input, x_t as well as the previously hidden state, employs a sigmoid function, and then outputs a vector that selects which bits of memory are used for the prediction.

3.4.4 *Recurrent Neural Network: Bidirectional LSTM*

Another type of recurrent neural network that is introduced in the implementation part of this document is called bidirectional LSTM (BiLSTM). Like long short-term memory (LSTM), bidirectional LSTM is also employed in natural language related problems. The difference is that LSTM works in one direction, while BiLSTM works in both directions. For example, a sentence like 'I agree with you" In the sentence, LSTM takes the vector of the word 'I' in the input layer and updates the next two layers: the hidden and output layers. The same goes for the other words in the sentence, too. LSTM takes one by one. BiLSTM works in two directions: forward and backward. In this case, the word 'I' would be taken in the forward direction and the word 'you' would be taken in the backward direction. For both forward and backward directions, BiLSTM is also updated in the same way that other words in this sentence would process an update. The LSTM model is unable to figure out certain complicated patterns in the dataset, but the BiLSTM model can, and this helps increase the performance of the text classification process. The architecture of BiLSTM looks like the following: A BiLSTM model has two LSTM,

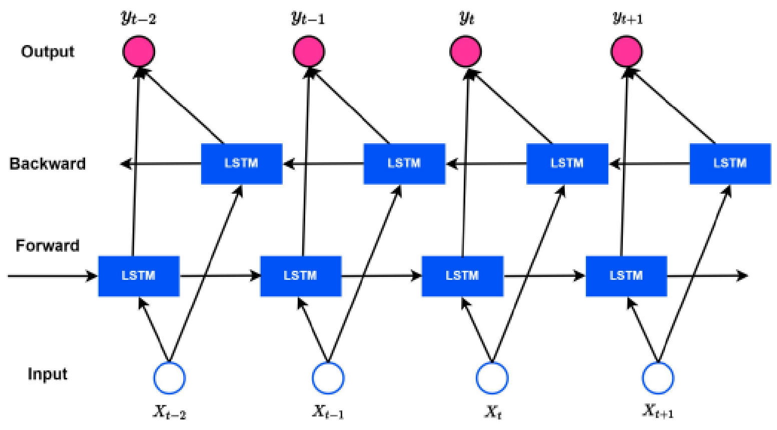


Figure 3.11: Bidirectional LSTM model [48].

where one is in the forward layer and the other is in the backward layer. If we take the above example again:

I agree with you

Here, X_{t-2} = 'I', X_{t-1} = 'agree, X_t = 'with', X_{t+1}= 'you' in the forward layer and X_{t-2} = 'you', X_{t-1} = 'with, X_t = 'agree'. X_{t+1}= 'I' in the backward layer. Both of these layers run separately with LSTM from one word to another. The output from both layers links together, which allows the final state to learn patterns or information from both $X_{t-2} \rightarrow X_{t-1} \rightarrow X_t \rightarrow X_{t+1}$ and $X_{t-2} \rightarrow X_{t-1} \rightarrow X_t \rightarrow X_{t+1}$.

In the implementation chapter of this document, it is shown that the LSTM model slightly outperforms the bidirectional LSTM model, even though bidirectional LSTM is an updated version of LSTM. Both the LSTM and bidirectional LSTM models tend to overfit the original training dataset of this thesis, but the bidirectional LSTM model is more overfitted than the LSTM model in the implementation part. In the evaluation phase of this thesis, a contrasting scenario has occurred where the evaluation performance of bidirectional LSTM is more adequate.

There is no accurate answer to why this could happen. It could happen because of the structure and quality of the dataset. It also depends on the proposed architecture of both models. The reason to implement both models is to finally know which model is more suitable to develop the chatbot.

3.4.5 Bidirectional Encoder Representations from Transformers (BERT)

Google Research Lab introduced a language model in 2018 named BERT, which is devoted to natural language-related tasks [10]. BERT is a pretrained model with a vast amount of unsupervised data. Therefore, when BERT is applied to any natural language dataset, it is more capable of capturing the meaning and relations between sentences in the dataset. How this Bidirectional Encoder Representations from Transformers (BERT) works is in its name. BERT uses transformers, which consist of two techniques named encoder and decoder. The input is read by the encoder, and the output is produced by the decoder. BERT is only concerned with the encoder, not the decoder. The encoder takes input such as a sentence or chain of sentences non-directionally, which means it learns the semantic meaning of a word based on its surrounding words. This is another difference between LSTM and bidirectional LSTM. The encoder takes the token

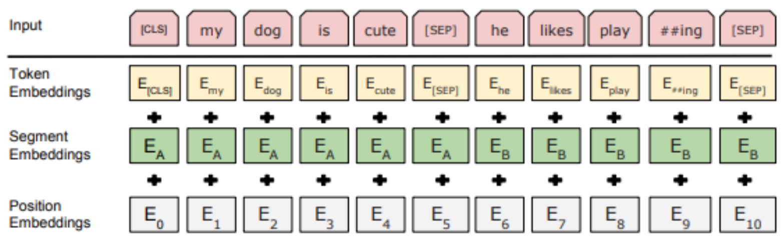

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 3.12: BERT model [10].

sequence, such as the input in the figure like ['my', 'dog',...]. That means a sentence is broken into words, then each word is again broken into its subword, such as the word 'playing', which breaks into 'play' and 'ing'. Then a classification token named 'CLS' was put into the beginning, and a separator token named 'SEP' was inserted after the end of each sequence of tokens. Token embedding helps to convert each token into a vector. This vector can be a float value too. The number of the sentence is essentially encoded into a vector, which is what segment embeddings are. And finally, positional embeddings are learned vectors that account for every conceivable position.

There are two approaches that help BERT achieve this non-directional training. The first one is masked language modeling (MLM). It helps identify missing words in a sequence of words based on their neighboring words. The approach is known as next sentence prediction (NSP). Next sentence prediction (NSP) merges two sequences of words to recognize the semantic relationship between these two sequences of words.

The BERT model can be overfitted for a smaller dataset where long-short-term memory (LSTM) would perform better. Even though BERT performs better in most cases, depending on the dataset, traditional recurrent models can perform better than BERT models too. Another issue is that BERT is computationally expensive, which means running a BERT model takes a lot of time and high computational resources. It is always better to apply different models to the same dataset to check which fits better with it.

3.5 *Evaluation Metrics*

This part describes the evaluation metrics that help evaluate the machine learning model. With the help of evaluation metrics, it is possible to assess how well a machine learning model is performing. Evaluation metrics such as accuracy, precision, recall, F1 score, and confusion matrix are common in text classification.

- True positive (TP):Correctly classified a text as belonging to a specific category or class [3].
- False positive (FP): Identifies a text as belonging to a specific category or class when it does not [3].
- True negative (TN): Correctly classified a text as not belonging to a specific category or class [3].
- False negative FN: Failed to classify a text as belonging to a specific category or class when it does [3].

a) Accuracy

The accuracy metric checks the percentage of correctly classified texts in text classification. Accuracy can also be described as the proportion of instances that are correctly classified [**?**]. For the above confusion matrix accuracy, the equation [42] would be,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

b) Precision

Precision checks the percentage of correctly classified positive texts out of all texts that are classified as positive. Correctly classified positive texts stand for texts that are actually classified in the testing phase, the same as classified in the training phrase. If a text is classified as 'football' in the testing phase, it is actually classified as 'football' in the training set too. All texts that are classified as positive are a combination of true positives and false positives. Precision [20] is,

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

c) Recall

The percentage of correctly classified positive texts out of all positive texts in the dataset for each category. For example, if there are four categories in texts and the machine learning model correctly classifies

80 percent of texts about one specific category, then the recall for that specific category would be 0.8. Recall [20] is,

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

d) F1 Score

The F1 score concerns the value of precision and recall, which range from 0 to 1. A higher score means the model performed well. F1 Score [20] is,

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

e) Confusion matrix

The confusion matrix helps to evaluate the performance of a machine learning model. The confusion matrix looks as Table 1 [31]:

Predicted / Actual	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Table 3.1: Confusion Matrix (Binary Classification)

If it is a binary classification, confusion matrices concern true positive, false positive, true negative, and false negative, but if the classification is not binary, then this is how the following should be calculated as Table 2 [31]:

Predicted / Actual	Transmission	Learning	Virtual
Transmission	a	b	c
Learning	d	e	f
Virtual	g	h	i

Table 3.2: Confusion Matrix (Multi-Class)

If we take transmission, Then, True positive = a
True negative = e + f + h + i
False positive = b + c
False negative = d + g
Therefore,

$$Precision = \frac{a}{a + b + c}$$

$$Recall = \frac{a}{a + d + g}$$

3.6 Model Poisoning

One of the main concerns for the final model would be the model poisoning attack. In model poisoning, the user deliberately introduces incorrect data to the model to corrupt its performance so that it cannot classify text in the future. How this attack is relevant to the thesis shown in 'Section 4.1'. This data could be biased and manipulated, which would undermine the accuracy of the model. There are two types of attacks [19].

First, when attackers inject so much malicious data that the boundary of the machine learning model becomes ineffective to classify, this type of attack is known as 'Availability'. Second, there is another attack known as a 'Backdoor attack'. To do this, malicious samples are added to the training data, which causes a model to respond in a particular manner when triggered by a particular pattern or keyword.

The second attack, named 'Backdoor Attack," is irrelevant to this thesis. Since it is not possible for users to add malicious samples to the training data since the chatbot is closed-ended, if someone wants, he or she can only inject malicious data in the chat response, not malicious samples in the training data. One way to mitigate an 'Availability' type attack is to apply anomaly detection algorithms to check whether any responses are significantly different from the chatbot or not. If yes, it would mark them as needing further inspection by hand or automatic. Another solution is to apply 'Feedback mechanisms', which allow for a manual report of any malicious responses.

3.7 *Programming Libraries*

Python is chosen to build the conversational agent since Python is popular in machine learning tasks. Following are the Python libraries that have been used in the implementation process.

- Pandas: This library provides data analysis tools as well as tools for data preprocessing such as cleaning, tokenizing, stop word removal, vectorization, etc.
- Scikit-learn (sklearn): It helps to use a variety of algorithms as well as help in data preprocessing such as one-hot encoding, countvectorizer, etc. It also helps in the evaluation part by helping to measure accuracy, precision, and recall.
- NLpaug: It helps to implement the data augmentation techniques that are discussed in Chapter 3. Data augmentation such as SpellAug, substitution augmentation, insertion augmentation, and synonym augmentation
- NumPy: It is used for numeral computing such as array manipulation, linear algebra, etc.
- TensorFlow: TensorFlow is introduced to build neural network models, which help handle large amounts of data. It is very common in natural language processing tasks such as text classification, language modeling, etc.

- NLTK: NLKK is a natural language toolkit that is used when working with human language data. It helps to do stemming, parsing, tagging, etc. It also provides prebuilt functionality, such as for text classification, to categorize text into predefined classes or categories.

4
Implementation

The previous chapter focuses on the theoretical details of techniques, and here it is shown how those techniques were applied to the original dataset. Initially, it starts with providing a diagram of the proposed chatbot and how it will be able to work. After that, it goes into detail on the dataset, its issues, and the coding method that was used for the dataset. The following section will discuss why data validation is necessary as well as how it has been carried out. Following that, discussions on data preprocessing and data augmentation will take place. The idea presented in Chapter 3 serves as the foundation for data augmentation. Finally, the neural network models provide a clear explanation of why the Bidirectional Encoder Representations from Transformers (BERT) model is a good fit for the original dataset and why designing the chatbot using this method is the best option where messages in the chat corpus refer to one another.

4.1 The Agent Model

The following is the model of the agent, including how it works and how it is able to classify messages in a chat corpus as referring to other messages.

1. Start.
2. User A initializes the chat by starting a discourse.
3. The agent read user A's response, classified it and saved it.
4. User B responses.
5. The agent read user B's response, classified it and saved it.
6. The agent checks whether user A's and user B's response classes are the same or not.
    - if not, the agent asks user B whether user B is sure that user B correctly refers user A.
        - if yes, the agent will change the class of user B's response in step 5 and update the neural network model.

    – else, agent asks user B to refer user A correctly, return to step 4.

  • else, return to step 2.

7. return to step 2.

8. End.

Typically every single chat system ever constructed is unable to classify each and every text perfectly. The agent model is designed with that in mind. For example, there are two users conversing with each other, and suddenly one of them, e.g., user 2, responds out of the corpus, which indicates that user 2 is unable to refer to user 1. Here comes the agent, which would ask user 2 to check whether his or her response refers to the response of user 1. If 'yes', the agent will collect the response and send it to the model to update the model. If 'no', it means user 2 is not able to refer to the response of user 1. The agent would ask user 2 to respond correctly to refer to user 1's response.

In this instance, the neural model is continuously updated with new replies, which are not classified. The main concern of this model relies on the user's responses to determine whether they are referring to each other. For example, if a user intentionally or unintentionally says 'yes," even though the user is not correctly referring to the other user's response, the neural model would update with incorrect data, which would lead to incorrect classification of data from the agent in the future. Therefore, since user honesty is not always guaranteed, the neural network would be polluted. This leads to data quality issues that can be overcome, as discussed in the "Discussion' chapter.

This also leads to a data poisoning attack if a user intentionally puts in incorrect data continuously and the model trains on that incorrectly classified data, which degrades the performance of the model. The posioning attack can be mitigated by taking some steps, which are introduced in the 'Discussion' chapter.

4.2 *About The Dataset*

The original dataset consists of conversations between students in Saarland University's education technology department. As part of an experiment designed for the educational technology department, 33 questions were posed to the students, to which they responded with their levels of comprehension, agreement, and disagreement. The questions were asked sequentially, and responses were recorded separately for each question. Each student was initially assigned an ID number. The responses from the students were collected and organized in an Excel file. In order to facilitate the conversation, several different discussion rooms were set up. The students in each room carried on the conversation by answering the questions that had been posed to them. At the conclusion of the study, each participant's response was aggregated into a single Excel document. Some sample questions and sample responses from the conversation file are as follows:

Scenario 1
Do you both agree with the following statement: Descriptive feedback contains specific information. Why?
User 111: Yes, descriptive feedback in particular offers more insight into students' performance.
User 110: Yes I agree with the statement. Descriptive feedback may contain specific strengths/weaknesses of students, as well as future suggestions for improvement.

Scenario 2
Do you both agree with the following statement: Hybrid learning includes the rotation model, flex model. Why?
User 114: Yes, rotation model (individual learning units are performed online) and flex model (virtual learning by face-to-face teacher as needed) are part of hybrid learning
User 116: Yes, the rotation and flex model are considered to be variations of hybrid learning, as well as self-blend and enriched virtual model.

Scenario 3
Do you both agree with the following statement: Augmented reality refers to, enhanced version of the physical world. Why?
User 113: yes
User 112: yes, because that's the definition of Augmented reality
User 113: because it deletiers external elements to enhance the physical world

4.2.1 *Dataset Problems and Coding*

Overall, the dataset is not large enough, and many of the responses are unrelated to the questions that were asked. These responses include things like 'hi', 'hello', 'bye', and similar phrases. The fact that many of the words are spelled incorrectly and that some of the texts are missing key information is one of the most significant issues with the dataset. It is not possible to code responses to each of the questions individually due to the fact that there is insufficient data. If the answers to each question are coded separately, we could end up with 33 unique sets of coding, which could result in a variety of issues, including overfitting, decreased model performance, and difficulty identifying patterns. Instead of individually coding each question, ten distinct clusters were created, and questions that were similar were grouped together in the same cluster. An example is as follows:

Cluster: transmission
Do you both agree with the following statement: Feedback refers to transmis-

sion of information about learner performance. Why?
Do you both agree with the following statement: Peer feedback refers to two-way communication process. Why?

Both of the preceding questions pertain to 'feedback' so we coded them and their respective responses in the 'transmission' cluster. The following is a list of cluster or code names.

- *Transmission: Feedback-related aspects of the question, such as descriptive feedback, peer feedback, evaluative feedback, etc.*
- *Augmented: Concerns about augment-related issues, including augmented reality, in educational questions and answers.*
- *Representation: Questions and answers involving graphical elements such as diagrams, maps, and other such thing.*
- *Interactivity: Discussion that focuses on the interaction between students and instructors, such as that which takes place in hybrid classrooms or on online study platforms.*
- *Learning: Involves the study of pertinent questions and the answers to those questions.*
- *Media: Multimedia-related question.*
- *Legit: Questions and answers concerning matters of law. One example from the dataset is 'Legalizing Marijuana'.*
- *Virtual: A virtual setting between the teacher and students.*
- *Technology: Involves the responses that concern technology.*
- *Generic: All of the non-topic responses. This generic section does not have any specific questions; rather, it has responses that are not relevant, such as 'hi', 'how are you', 'what time...', etc.*

Even so, the dataset is insufficient to be used as input for a machine learning model due to the fact that there is not enough data. In order to improve the quality of the dataset, several well-known data augmentation techniques were utilized, all of which are discussed in greater detail in the following section titled 'Data Augmentation'.

4.2.2 *Validation of Coding*

It is important to verify the validity of the coding assigned to datasets, as the coding of each text can vary from person to person. It is necessary for this reason because some words can have different meanings depending on the context in which they are used, and different terms can sometimes imply the same meaning [25]. To ensure the code's reliability, it is essential that different people code the same text in the same way, as briefly discussed in the section above.

1st Coder: Koushik Chowdhury, MSc Student, Saarland University, Saar-

brücken, Germany
2nd Coder: *Tajbeed Ahmed Chowdhury, PhD Student, Saarland University, Saarbrücken, Germany*

Cohen's Kappa is utilized as a tool for determining how reliable one coder is in comparison to another. The reason for utilizing Cohen Kappa is that it is the most commonly used statistical measure for assessing inter-rater reliability between two coders. The result is 0.7, which indicates that two coders have fair or moderate level of agreement [30]. Even though two coders do not completely agree on all of the data that was coded, the results indicate that there is a fair amount of consistency in the data that was coded. The coded dataset was used for further experimentation as a result, even though there is still room for improvement to get the best Cohen Kappa results. It is more difficult to code thousands of texts again, and it may take several tries to get the 'near-perfect agreement' results from Cohen Kappa. Additionally, there is a chance that even after many tries, the 'near-perfect agreement' is not achieved. The second coder is performing this as a volunteer effort and is not accountable for this thesis; therefore, it is important to keep in mind that repeating the same thing repeatedly across thousands of sentences takes time.

4.2.3 Word Cloud

A word cloud graphically represents word collections based on their recurrence in the source text or dataset. With the help of a word cloud, it is possible to determine important words from the source text, which helps to get a better understanding of the source text before applying any classifier. The following is the word cloud output for the dataset. In the word cloud, it is noticeable that some of the

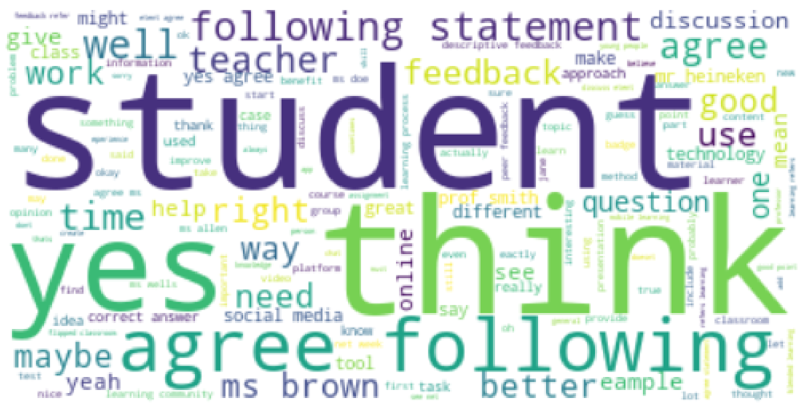


Figure 4.1: Word cloud of original dataset.

words are large and some of them are small. The size indicates the frequency of the word in the source text. This frequency also helps determine the importance of the word in the dataset. For example, the word 'student' is used more frequently than the word 'teacher' here. The color of the word has no meaning here.

4.3 Data Pre-processing

The name of the original dataset is 'ETCL-Messages Dataset'. To feed the dataset into the algorithm, proper preprocessing is required. The raw text data is often unstructured and contains irrelevant information. Irrelevant information can be of many types; for example, punctuation, stop words, and special characters, etc. create noise in the dataset that prevents the machine learning model from learning. The preprocessing of the dataset helps it be noise-free as well as structured and meaningful. The preprocessed dataset can be easily understood by a machine learning model. The following are some reasons, with a sample from the original dataset, why it is required to do proper data processing on our dataset.

Do you both agree with the following statement: Badges display accomplishments. Why?
User Id: 114: Yes, I agree, they can easily start conflicts and demotivate students. They display accomplishments, but the absence of badges could display failing.
User Id 116: I do agree with the statement. I would also say they not only display achievements but also give context on how did the student achieved it. However, a badge may become a goal for a student, the same as a grade, instead of being a feedback/evaluation device.

- These texts contain special characters (? , . : - /), which are irrelevant information as they do not add anything to the overall sense of the text. The stop words such as 'the', 'a', 'an', 'she," etc. also do not carry value in the text analysis. By removing this type of stop word and special characters, it is possible to shift the focus to the more meaningful words, which at the end help in text classification. The numbers are also part of the data noise, so they also need to be removed so that the algorithm can pivot its focus on relevant information.

- The above text contains both uppercase and lowercase characters, but for the algorithm, it is essential to have the dataset in the same format instead of multiple formats such as uppercase and lowercase.

- The text classification algorithm cannot take text by itself as input data. The preprocessing helps convert the input text data into numerical form with the help of word embedding techniques, which are briefly explained in Chapter 3.

With the help of 'pandas', which is a data analysis library in the Python programming language, text preprocessing has been done on the original dataset. The main objective is to clean the text data and prepare it for the neural network model. The following are the pre-processing steps taken for this dataset:

In the following statistics from Table 3, it is clearly shown how the size of the original data is reduced.

Details	¬ preprocessed	preprocessed
Samples	4938	4938
Characters	446362	288438
Characters per sample (avg.)	90.39	58.41
Words	77680	39981
Words per sample (avg.)	15.73	8.10
Number of unique words	5278	3066
Unique words per sample (avg.)	1.07	0.62
Number of stopwords	35605	0
Stopwords per sample (avg.)	7.21	0
Number of punctuations	13435	20
Punctuations per sample (avg.)	2.72	0.04
Average sentence length	1.55	0.97
Maximum sentence length	186	108
Mean sentence length	15.73	8.10
Median sentence length	10	5

Table 4.1: Data analysis report: preprocessed vs without preprocessed.

The total number of characters in the dataset has been reduced by 35 percent. Initially, the datasets contained 35605 stopwords; after the preprocessing, the number of stopwords was zero. The number of punctuation marks is reduced to 20 from 13435. Some of the punctuation could not be removed because those punctuation marks carry semantic meaning into the text. Therefore, if those punctuation marks were removed, some of the important information in the text would be lost. It is also very noticeable how sentence length was reduced after the preprocessing of the dataset.

4.4 Data Augmentation

One of the main implementation parts of this thesis is data augmentation. Since the original dataset is not large enough to feed into a neural network model, it is necessary to increase the size of the dataset by applying some common data augmentation techniques. Word embedding approaches such as Word2vec, GloVe, and fastText have been applied for data augmention. The question would be why there are different approaches instead of one. The reason for applying several approaches is that every approach has its own strengths and weaknesses; therefore, by applying multiple approaches, it is possible to overcome each other's limitations and get better performance in text classification. Another benefit is that each of them has their own technique to capture the semantic meaning, and by combing their embeddings, it is possible to create diverse embeddings. For each of these three approaches (Word2vec, gloVe, and fastText), the following augmentation techniques were applied in the implementation process, which are briefly described in the "Background Theory' chapter.

- *SpellingAug: Spelling augmentation that helps to substitute words for their corresponding misspelled words.*
- *Insertion Augmenation: Insert new words into a specific existing text.*

- *Substitute Augmentation: Replace the word with a word with similar meaning (not a synonym).*
- *Synonym Augmentation: Replace words with their corresponding synonyms.*

Details	before	after
Samples	148400	148400
Characters	17331232	12601232
Characters per sample	116.79	84.91
Words	2796391	1641708
Words per sample (avg.)	18.84	11.06
Unique words	227604	175778
Unique words per sample (avg.)	1.53	1.18
Stopwords	1035425	0
Stopwords per sample (avg.)	6.97	0
Punctuations	488592	39433
Punctuations per sample (avg.)	3.29	0.26
Average sentence length	1.52	0.98
Maximum sentence length	213	121
Mean sentence length	18.84	11.06
Median sentence length	14	8

Table 4.2: Data analysis of data augmentation: before preprocessing of dataset and after preprocessing of dataset.

After applying the data augmentation technique to the preprocessed data, it is noticeable that, compared to Table 3, the sample size and number of words have increased, as well as the average punctuation. The following figure illustrates the differences more clearly:

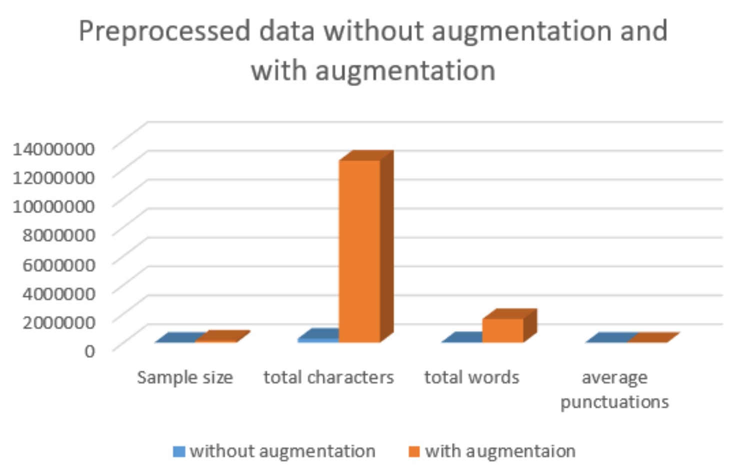


Figure 4.2: Preprocessed Data with augmentation and without augmentation.

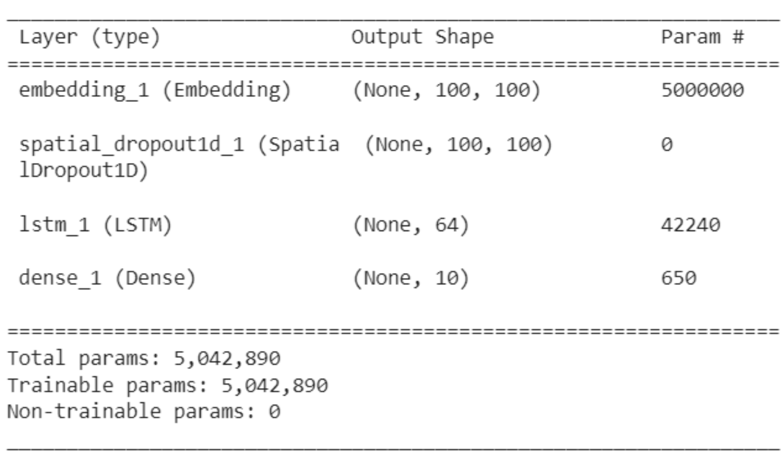
From the above chart, it is more visible how the data size increases extremely after the data augmentation techniques. The sample size is increased by almost 30 times, and both the total character number and the total word number are increased by more than 40 times. This extreme size of the dataset helps the algorithm learn patterns and correctly classify text.

4.5 Recurrent Neural Network

The proposed two recurrent network described in the 'Background Theory' chapter has been applied to both preprocessed augmented datasets and unaugmented preprocessed datasets, which helps to evaluate the outcome. Initially, the datasets were split into training sets and test sets. The training set size is 90 percent, and the test set is 10 percent. The test set assists in determining whether our neural networks are capable of classifying the text after training with the training set.

4.5.1 Long short-term memory (LSTM)

The following architecture of the LSTM model has been introduced for both augmented and unaugmented processed datasets. This model



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 100)          5000000

 spatial_dropout1d_1 (Spatia  (None, 100, 100)         0
 lDropout1D)

 lstm_1 (LSTM)               (None, 64)                42240

 dense_1 (Dense)             (None, 10)                650

=================================================================
Total params: 5,042,890
Trainable params: 5,042,890
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.3: LSTM model.

is built with the help of 'Keras', which is a Python library. This is a sequential model that uses long short-term memory (LSTM) for text classification. This model has four layers named 'Embedding', 'SpatialDroupout1D', 'LSTM', and 'Dense' layer. The parameter in this context refers to a value learned by the model through the training process. The weights and biases utilized to convert input data into output predictions are referred to as parameters in neural networks. In a simple model, the parameters may encompass the slope and intercept coefficients of the regression line that is employed to fit the training dataset.

- *Embedding: The initial layer involves transforming every term in the input sequence into a vectorized representation of a consistent dimensionality. The proposed methodology involves the processing of input sequences with a fixed length of 100. Each word index within the sequence is subsequently transformed into a dense vector representation of 100 dimensions.*

- *SpatialDropout1D: The application of dropout is implemented on the embedding layer with the aim of mitigating the issue of overfitting. The output shape of the aforementioned is (None, 100, 100). Nevertheless, the* *model lacks trainable parameters, resulting in a parameter count of zero.*

- *LSTM: The model is capable of capturing extended long-term relationships or dependencies within the input sequence. The resulting output of the model will be a singular vector with a dimension of 64 for every input sequence. The model comprises a total of 42,240 trainable parameters.*
- *Dense: The resultant outcome will generate a probability distribution encompassing ten unique classes, as the initial dataset has been encoded with ten categories. The model consists of 650 parameters that are subject to training.*

The model has been compiled utilizing the 'categorical_crossentropy' loss function, which is commonly employed in multi-class text classification. This loss function serves to quantify the disparities between the predicted and actual probability distributions of the target variable. In addition to the loss function, the model was compiled with an optimizer called 'adam' that facilitates the updating of the model's weights as well as 'accuracy' metric is applied in the compilation process. The following are the model accuracy and loss values for the augmented preprocessed datasets and the unaugmented preprocessed datasets. The loss value comes from the loss function. The

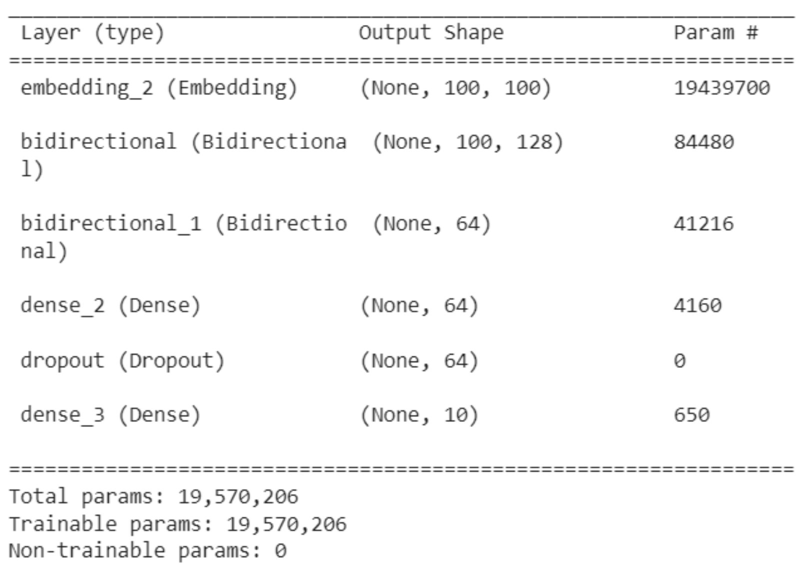
Augmented Pre-processed data	Training	Test
Accuracy	0.9058	0.8709
Loss Value	0.2958	0.624
Unaugmented Pre-processed data	Training	Test
Accuracy	0.7591	0.8101
Loss Value	0.7115	0.6473

Table 4.3: LSTM Results.

model performs well with augmented data in terms of accuracy in both the training and test sets. The loss value for the test set is higher than the train set, which is more than double, 0.624, which indicates a potential issue of overfitting for an augmented preprocessed dataset. For a preprocessed dataset without augmentation, test accuracy is larger than train accuracy, which is also an indication of overfitting. Comparing these two datasets, the augmented preprocessed dataset performs better, but it still indicates a performance issue, such as overfitting in this case.

4.5.2 *Bidirectional Long short-term memory (BiLSTM)*

Bidirectional LSTM (BiLSTM) is another type of a recurrent neural network [27]. Similar to long-short-term memory (LSTM), bidirectional LSTM is described in the 'Background Theory'. The following model has been designed for this approach: This model introduced two bidirectional layers as well as adding another dense layer in comparison to long-short-term memory (LSTM). The first bidirectional layer executes a dual-directional analysis of the input sequence, whereas the subsequent bidirectional layer performs a dual-directional analysis of the output sequence generated by the first bidirectional layer.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 100, 100)          19439700

 bidirectional (Bidirectiona  (None, 100, 128)         84480
 l)

 bidirectional_1 (Bidirectio  (None, 64)               41216
 nal)

 dense_2 (Dense)             (None, 64)                4160

 dropout (Dropout)           (None, 64)                0

 dense_3 (Dense)             (None, 10)                650

=================================================================
Total params: 19,570,206
Trainable params: 19,570,206
Non-trainable params: 0
```

Figure 4.4: BiLSTM model.

The primary objective of the initial dense layer is to decrease the output's dimensionality from the preceding layer while also incorporating non-linearity into the model. The second dense layer serves the function of producing the ultimate output of the model, which manifests as a probability distribution encompassing the 10 feasible classes. The initial dense layer comprises 64 neurons that utilize the rectified linear unit (ReLU) activation function, while the subsequent dense layer consists of 10 neurons that employ the softmax activation function. The following is the model accuracy and loss value for the augmented preprocessed dataset and the preprocessed dataset without augmentation from the Bidirecional Long Short-Term Memory (BiLSTM) model.

Augmented Pre-processed data	Training	Test
Accuracy	0.8979	0.8771
Loss Value	0.3280	0.7169
Unaugmented Pre-processed data	Training	Test
Accuracy	0.7414	0.7859
Loss Value	0.7305	0.6771

Table 4.4: BiLSTM Results.

For the augmented preprocessed dataset, the difference between accuracy from the train set and the test set is closer to each other, but the loss value is higher for the test set than the train set, which is almost twice as much. For the preprocessed dataset without augmentation, training accuracy is lower than test accuracy, and in both cases, it provides a high loss value. The model performed slightly well with the augmented preprocessed dataset, but similar to the long-short-term memory (LSTM) model, it indicated an overfitting issue.

4.5.3 *Comparison of Recurrent Neural Network variants*

The figure indicates the difference in results between long-short-term memory (LSTM) and bidirectional LSTM (BiLSTM). For both mod-

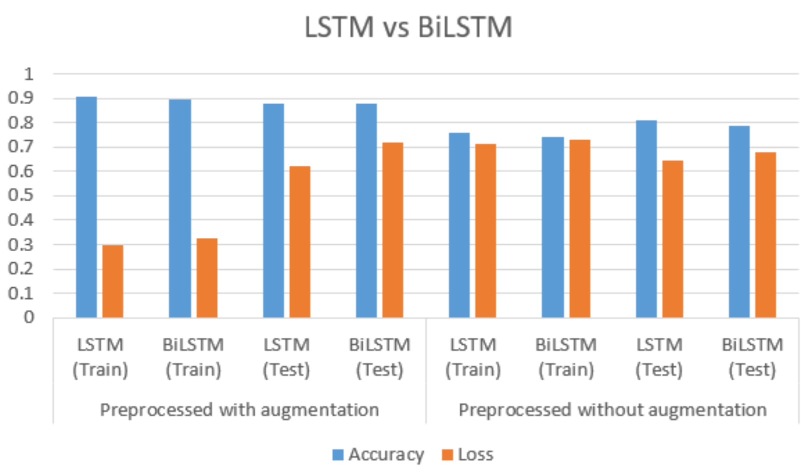
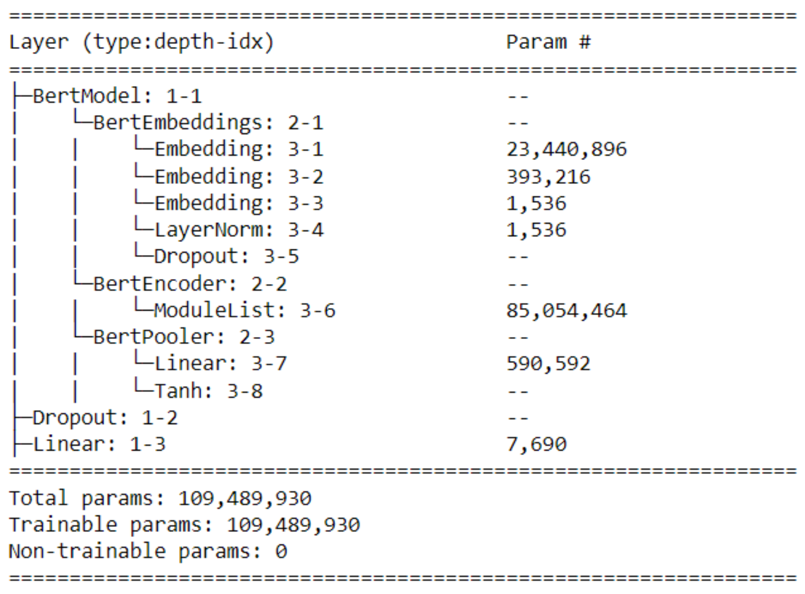


Figure 4.5: LSTM Results vs BiLSTM Results.

els, the preprocessed dataset without augmentation can be ignored because, in terms of accuracy, both models produce lower accuracy and a high loss value. Both models perform well with augmented-preprocessed data. Long short-term memory (LSTM) is slightly better than bidirectional long-term memory (BiLSTM) because, in both the training and testing sets, long-term memory (LSTM) provides better accuracy and a lower loss value. The concern here is the loss value. For both models, the loss value increases significantly in the testing set compared to the training set. This loss function could be reduced with the help of epoch by increasing the epoch value; here, epoch = 1 indicates that the process of a model involves a full iteration over the entire training dataset. The main issue is that it may reduce the loss value, but it would increase the accuracy, which is also an issue with overfitting. To overcome this issue, a new model named Bidirectional Encoder Representations from Transformers (BERT) is introduced, which is a family of neural networks but not recurrent neural networks.

4.6 *Bidirectional Encoder Representations from Transformers (BERT)*

In Chapter 3, it is clarified how this model works and what the main differences are between it and recurrent neural network variants. The main difference between BERT and recurrent neural networks is that BERT is an already trained model with a large amount of data, whereas recurrent neural networks are not. Therefore, the BERT model has the ability to capture dependencies better than a recurrent neural network when it is performed on a new dataset. The reason BERT has been applied in the implementation process is because, even though both recurrent neural networks have better accuracy, both of them tend to overfit. In addition to its benefits, one of the primary drawbacks of the BERT model is that, in comparison to recurrent neural networks, it demands a significant amount of computational resources. These resources include a significant number of graphics processing units, as well as training and inference periods of time. The inference time refers to the time required for a trained model to make a prediction or classify new data. The following architecture of this model has been used in this implementation section: From the



```
=================================================================
Layer (type:depth-idx)                   Param #
=================================================================
├─BertModel: 1-1                         --
|    └─BertEmbeddings: 2-1               --
|    |    └─Embedding: 3-1               23,440,896
|    |    └─Embedding: 3-2               393,216
|    |    └─Embedding: 3-3               1,536
|    |    └─LayerNorm: 3-4               1,536
|    |    └─Dropout: 3-5                 --
|    └─BertEncoder: 2-2                  --
|    |    └─ModuleList: 3-6              85,054,464
|    └─BertPooler: 2-3                   --
|    |    └─Linear: 3-7                  590,592
|    |    └─Tanh: 3-8                    --
├─Dropout: 1-2                           --
├─Linear: 1-3                            7,690
=================================================================
Total params: 109,489,930
Trainable params: 109,489,930
Non-trainable params: 0
=================================================================
```

Figure 4.6: BERT Model.

figure, it is noticeable that there are three layers in this model- named 'BERTModel: 1-1', 'Dropout', and 'Linear'. ' BERTModel: 1-1' has three sublayers, which are,

- *BertEmbeddings: The BertEmbedding layer is designed to take textual input and generate token embeddings. This layer also has three distinct sublayers for embedding, each of which is designated as 'Embedding: 3-1', 'Embedding: 3-2', and 'Embedding: 3-3' accordingly. These sublayers are referred to by their respective names. Token embedding, segment embedding, and position embedding are the three types of data that are handled by each of these sublayers of embedding. The input of each layer can be normalized by employing the 'LayerNorm'. The term 'dropout' is employed to stop 'overfitting'.*

- *BertEncoder: According to the information presented in Chapter 3, it is constructed using a line of transformer blocks. The BertEncoder is fed input embeddings from the BertEmbeddings layer, which are created as the layer is being trained. These embeddings are then used by the BertEncoder. The output of the transformer block that came before it in the BertEncoder is fed into the transformer block that came after it, which results in a new output being generated by the BertEncoder.*

- *BertPooler: A vector representation of the input text is generated by the*

BertPooler layer after it receives the output from the BertEncoder layer, which is the layer directly above it. This particular vector representation is of a fixed size, which indicates that it possesses a definite number of dimensions. It facilitates the ease with which it can be connected with other elements of a model.

- *Dropout: By arbitrarily removing some of the nodes from the training set at regular intervals, dropout helps to prevent overfitting.*
- *Linear: This layer is responsible for mapping the value that was calculated by the BERTPooler layer into the desired format for output. This layer is sometimes referred to as the 'output layer' in some contexts.*

The following are the BERT model accuracy and loss values for the augmented preprocessed datasets and the unaugmented preprocessed datasets.

Augmented Pre-processed data	Training	Test
Accuracy	0.94	0.93
Loss Value	0.17	0.14
Unaugmented Pre-processed data	Training	Test
Accuracy	0.88	0.90
Loss Value	0.41	0.38

Table 4.5: BERT Results.

The results indicate the BERT model provides a high level of accuracy in training and testing for both augmented and unaugmented preprocessed datasets. The accuracy results are slightly better for augmented preprocessed datasets.

The loss value for the unaugmented preprocessed dataset is higher in both the training and test sets, which led to overfitting. There is room to reduce the loss value for the unaugmented preprocessed dataset, but by removing the loss value, it would increase the model accuracy closer to 1, which is also an indication of overfitting.

For the augmented preprocess test, test loss of 0.14 is significantly lower than train loss of 0.17, which is good in terms of performance as well as indicating no overfitting. The test accuracy and train accuracy are also close to each other. A test accuracy of 0.93 tells us that the BERT model is able to classify 93 percent of test data instances. Overall, the BERT model produces strong performance for the augmented preprossed dataset.

4.7 *Recurrent Neural Network Vs BERT*

From Section 4.4.3, it is obvious that the long-short-term memory (LSTM) model performs better than bidirectional long-short-term memory (BiLSTM). Therefore, the following figure provides an illustration of how the LSTM model and the BERT model work in terms of accuracy and loss value. The figure shows the performance com-

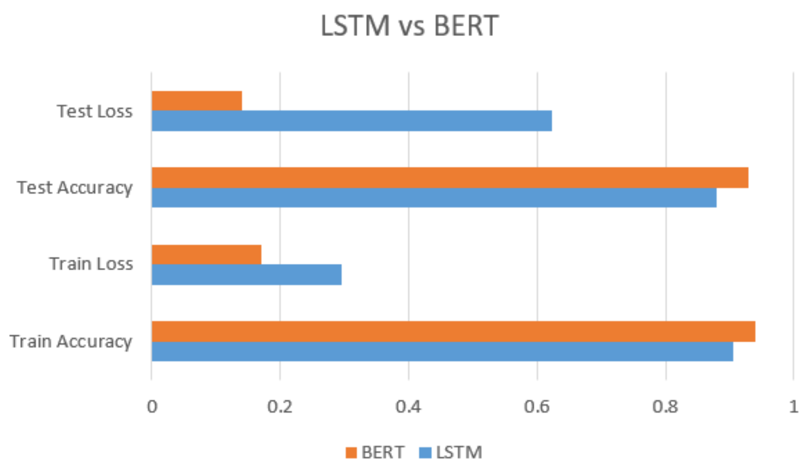


Figure 4.7: LSTM Model vs BERT Model.

parison between long-short-term memory (LSTM) and bidirectional encoder representations from transformers (BERT) models on a classification task. Both models are trained on an augmented preprocessed dataset and have been evaluated on both a training set and a test set. The training set is 90 percent of the original augmented preprocessed data, and the other 10 percent is the test set.

The results indicate the BERT model achieved a lower loss in both training and test sets compared to the LSTM model. The LSTM model produces a higher loss value in the training phase, which then almost more than doubles in the testing phase. This illustrated that the BERT model was able to minimize the error between expected and actual outputs, where the LSTM model was unable to do so.

In terms of accuracy, both LSTM and BERT performed well, but BERT provides slightly better accuracy in both the training and test phases than the LSTM model. The main concern of the LSTM model is that, even though it produces good accuracy in both the training and test set, it provides a high loss value. On the other hand, the loss value and accuracy are balanced for the BERT model in both the training and test phases.

Overall, the figure indicates that bidirectional encoder representations from the Transformers (BERT) model performed better where long-short-term memory (LSTM) tends to overfit, which results in being unable to classify texts.

5 Evaluation and Results

This chapter shows the evaluation approaches followed to evaluate the neural network model. There are two evaluation approaches that have been followed here. The first one is based on the confusion matrix, which was applied to the testing dataset. The second approach is based on a new dataset that was created with the help of the 'Open AI API' and contains the responses to 33 questions that were asked for the data collection. This new dataset is useful for evaluating the neural network models in order to determine how effectively the models identify the texts contained in the new dataset and how well they can classify those texts.

5.1 Confusion Matrix

As discussed in Chapter 3, the confusion matrix tells how many texts are correctly classified in the test set based on the trained model. Since the Bidirectional Encoder Representations from Transformers (BERT) model performed well on the dataset, the following confusion matrix report is based on BERT results on an augmented preprocessed dataset on the testing set.

$$\begin{array}{c} \\ Augmented \\ Generic \\ Interactivity \\ Learning \\ Legit \\ Media \\ Representation \\ Technology \\ Ttransmission \\ Virtual \end{array} \begin{bmatrix} A & G & I & L1 & L2 & M & R & T1 & T2 & V \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 5 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 2 & 25 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 3 & 59 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 16 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 0 \\ 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 54 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix 1: The confusion matrix report is based on BERT on the testing set.
Here, A = augmented, G = generic, I = interactivity, L1 = learning,

L2 = legit, M = media, R = representation, T1 = technology, T2 = transmission and V = virtual. This confusion matrix has been run with a small portion of data, which is 494 samples from the test case, taken randomly.

	precision	recall	f1-score	support
augmented	0.80	1.00	0.89	4
generic	0.98	0.96	0.97	311
interactivity	0.89	0.86	0.88	29
learning	0.89	0.92	0.91	64
legit	1.00	1.00	1.00	1
media	1.00	0.89	0.94	18
representation	1.00	1.00	1.00	1
technology	1.00	0.57	0.73	7
transmission	0.79	0.93	0.86	58
virtual	0.00	0.00	0.00	1
macro avg	0.84	0.81	0.82	494
weighted avg	0.94	0.94	0.94	494

Table 5.1: Classification Report of Matrix 1.

The samples were overly populated with 'generic' responses, then 'learning' and 'transmission' respectively. For example, the precision value for the 'learning' response is 0.89, which means the BERT model is 89 percent precise in identifying the learning samples. Some of the categories received precision = 1, which is because of lower samples in these categories. Even with the lower samples, the BERT model has correctly classified the label or category for those samples.

Overall, the weighted prediction is 0.94, which is good compared to the accuracy received from the BERT model for both the training and test phases. This sample contains only one response from the 'virtual' category, which is not correctly classified by the BERT model, where the model correctly classified all responses correctly related to the 'augmented', 'legit', and 'representation' categories, though the size of the responses from these categories in the sample is lower.

5.2 *New Dataset*

OpenAI is a research institute that works with artificial intelligence. OpenAI gives opportunity to every registered user to generate API key [2]. API key is an identifier that is not the same for all users. Every user can generate a unique API key. An API key from OpenAI is needed for authentication when something integrates with the OpenAI platform. This helps to access the natural language-based platform of OpenAI. The integration code to connect openAI is [2].

```
openai.organization = "unique_individual_id"
openai.api_key = os.getenv("generated_unique_key")
```

For evaluation purposes, this API key is integrated with Python to create a new dataset based on the 33 questions that were asked in the data collection process. The main difference between the original dataset and this new dataset is that the original dataset contains the responses to those 33 questions and other non-related texts, and overall, the original dataset has many complications such as noise, irrelevant punctuation, sudden stops of texts, etc., whereas the new dataset points to the questions without any irrelevant texts. The second difference is that the new dataset is an artificially intelligently generated dataset made with the help of Python via the OpenAI API, while the original dataset is human-generated and was done in an experiment in the educational technology department at Saarland University.

The main question could be asked: why this new dataset when we already have the testing data from the original augmented preprocessed dataset? Testing data and training data are both part of the original dataset; therefore, when well-trained models are applied to the testing data, it is obvious that models are able to classify most of the texts in the testing set because models feed the training data, which is from the same source as the testing data.

Users in real life would not type texts in the chatbot based on the testing dataset because they had no idea what was in the testing dataset or training dataset. It is important for this thesis that messages in the chat corpus refer to each other; therefore, it is important that the final model be able to classify the messages. The new dataset is similar to the user's responses, only it is machine-based instead of real humans. To test the final result with a group of humans takes time, and sometimes it is expensive too because it requires a good amount of money to hire humans for the experiment to test the final result that helps in the evaluation of this study.
Number of responses: **1005**
Number of unique classes: **9**

Class or Label	Percentage of responses
learning	30.96
transmission	22.59
interactivity	20.50
media	8.96
augmented	4.48
representation	4.39
technology	3.88
legit	2.79
virtual	1.49

Table 5.2: Percentage of responses per category in the new dataset.

That's why these machine-based responses, which similar to human responses but generated by machine. The only difference is that humans can be error-prone and biased; machines overcome this limitation by staying strict and to the point. Table 5.2 shows some statistics for this new dataset.

From the statistics, it is clearly shown that the label 'learning' has the most responses here, as well as 'transmission' and 'interactivity', which provide good percentages of responses from the overall dataset. The response from the 'virtual' class is lower than all of the others. The following is the word cloud of this new dataset: The words such

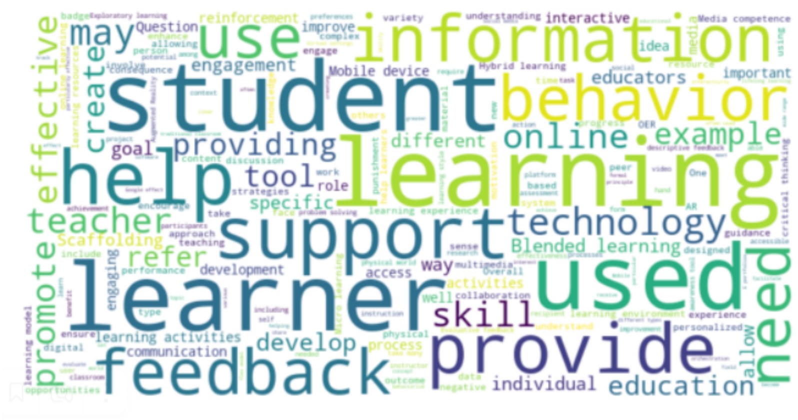


Figure 5.1: Word cloud of new dataset.

as 'student', 'learner', and 'learning' are most frequent in the overall new dataset, and the words such as 'way', 'well', etc. have less importance than those most frequent words.

5.3 *Evaluation of Neural Network Models*

From Chapter 4, it is clearly visible that Bidirectional Encoder Representations from Transformers (BERT) outperform both recurrent neural networks that were proposed for the implementation part based on accuracy and loss value. In this section, it is evaluated whether the BERT model is really able to outperform the LSTM model and the BiLSTM model or not.

5.3.1 *Long short-term memory (LSTM)*

The new dataset is tested with the LSTM models (that have already been trained in implementation) to check how many responses this model is able to classify. Both the augmented preprocessed and unaugmented preprocessed trained LSTM models are considered here. The result is as follows:

Model	Correctly classified	Incorrectly classified
LSTM1	142	863
LSTM2	148	857

Table 5.3: Trained LSTM models performance on a new dataset.

Here, LSTM1 = LSTM model that trained with the unaugmented preprocessed dataset and LSTM2 = LSTM model that trained with the augmented preprocessed dataset.

Out of 1005 responses, the LSTM model has been able to classify 142 responses when the Long short-term memory (LSTM) model is trained with the original unaugmented preprocessed dataset. The result is improved for the Long short-term memory (LSTM) that is trained with the original augmented preprocessed dataset, which is 148, but still only 14.7 percent of the overall new dataset. In the implementation part, it is discussed how the LSTM model tends to overfit. In this evaluation part, it is visible that even though the LSTM model performed very well with the training dataset, it still has negative effects on the new dataset.

5.3.2 *Bidirectional LSTM (BiLSTM)*

Similarly to LSTM, the new dataset has been experimented with the bidirectional LSTM (BiLSTM) models. The results tell:

Model	Correctly classified	Incorrectly classified
BiLSTM1	98	907
BiLSTM2	196	809

Table 5.4: Trained BiLSTM models performance on a new dataset.

Here, BiLSTM1 = BiLSTM model that trained with the unaugmented preprocessed dataset and BiLSTM2 = BiLSTM model that trained with the augmented preprocessed dataset.

The BiLSTM model that was trained with an unaugmented preprocessed dataset only classified 98 responses correctly out of 1005 responses from the new dataset. The result has been improved almost double with the BiLSTM model that was trained with the original augmented preprocessed dataset, but still only 20 percent of the new dataset.

5.3.3 *Evaluation: LSTM vs BiLSTM*

The most interesting finding of this evaluation part is that the bidirectional LSTM (BiLSTM) model performs slightly better than the long short-term memory (LSTM) model, but in Chapter 4, it is shown how LSTM outperforms BiLSTM. One of the reasons this happened would be the different distribution of the new dataset compared to the original dataset. Also, it is observed that accuracy cannot be the only metric to evaluate the model's results. Although BiLSTM performs better here, the difference is not that large when both the LSTM model and the BiLSTM model have been able to classify only 14.7 percent and 20 percent, respectively. Both models failed to classify half of the responses in the new dataset. Therefore, if one of these recurrent neural network models is applied to build the final application, it is evident that most of the time both models cannot be used to assess whether messages in the chat corpus refer to each other or not.

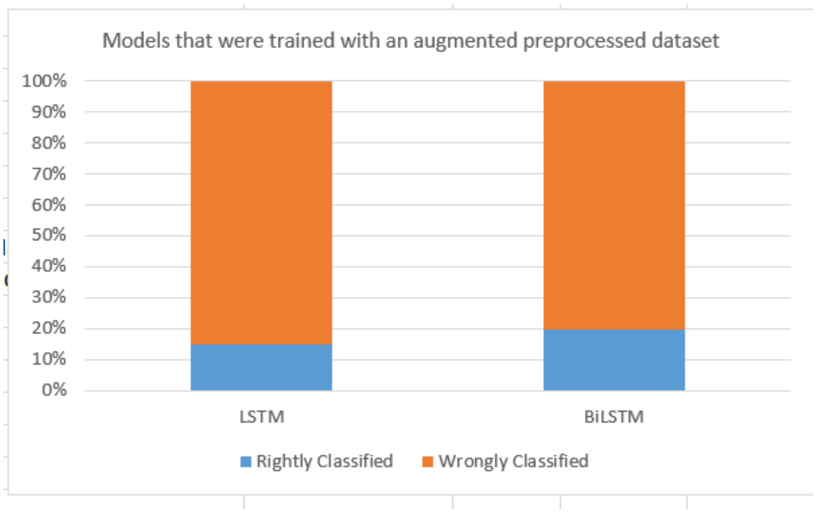


Figure 5.2: Evaluation: LSTM vs BiLSTM: The percentage of correctly classified and incorrectly classified.

5.4 Bidirectional Encoder Representations from Transformers (BERT)

Since this model highly outperforms both recurrent neural network models in terms of accuracy and loss value matrices, this part of the section takes a more in-depth look at the performance to evaluate the BERT model result, which was trained using both augmented and unaugmented preprocess datasets.

Model	Correctly classified	Incorrectly classified
BERT1	720	285
BERT2	960	45

Table 5.5: Trained BERT models performance on a new dataset.

In the implementation part, the BERT model was applied to both unaugmented and augmented preprocessed datasets. We can name them, such as BERT1 as an unaugmented preprocessed dataset and BERT2 as an augmented preprocessed dataset.

When the BERT model was trained with an unaugmented dataset, it classified 720 responses correctly out of 1005. The result is improved with the BERT model that was trained with an augmented preprocessed dataset. Almost 95.52 percent of the time, the model classified the response correctly. This evaluation result is significantly higher than both recurrent neural network models. There could be several reasons behind this remarkable improvement. The first reason is that the BERT model is already trained with a large amount of unsupervised data; therefore, when it is applied to the original dataset of this study, it easily understands the insight meaning of the original dataset. Second, the BERT model is very new to the field, with an improved architecture released by Google in 2018 and dedicated to the natural language processing task [10].

$$\begin{array}{c|cccccccccc|}
 & A & G & I & L1 & L2 & M & R & T1 & T2 & V \\
Augmented & 25 & 2 & 3 & 6 & 0 & 0 & 2 & 7 & 0 & 0 \\
Generic & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Interactivity & 1 & 9 & 140 & 37 & 0 & 9 & 0 & 0 & 10 & 0 \\
Learning & 0 & 22 & 15 & 239 & 0 & 3 & 1 & 0 & 31 & 0 \\
Legit & 4 & 11 & 0 & 3 & 4 & 0 & 0 & 2 & 4 & 0 \\
Media & 0 & 5 & 1 & 5 & 0 & 72 & 2 & 0 & 5 & 0 \\
Representation & 0 & 10 & 1 & 3 & 0 & 0 & 26 & 0 & 4 & 0 \\
Technology & 0 & 3 & 5 & 2 & 0 & 1 & 0 & 24 & 4 & 0 \\
Ttransmission & 0 & 18 & 3 & 15 & 0 & 0 & 1 & 0 & 190 & 0 \\
Virtual & 3 & 0 & 2 & 9 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}$$

Matrix 2: Confusion Matrix when BERT1 tests with new dataset)

The following report is the classification report for BERT1 model. The reason for all 0 values in the 'generic' class is that in the new dataset, no generic responses such as 'hi', 'hello', 'bye', etc. are considered. As stated earlier, responses in the new dataset are strict and to the point. From the confusion matrix following, it is observed that when BERT1 tests the new dataset, it provides 72 percent accuracy, which means that 72 out of 100 times, it correctly classified the responses from the new dataset.

Accuracy: 0.72 For example, there are 311 responses labeled 'learning'

	precision	recall	f1-score	support
augmented	0.76	0.56	0.64	45
generic	0.00	0.00	0.00	0
interactivity	0.82	0.68	0.74	206
learning	0.75	0.77	0.76	311
legit	1.00	0.14	0.25	28
media	0.84	0.80	0.82	90
representation	0.81	0.59	0.68	44
technology	0.73	0.62	0.67	39
transmission	0.77	0.84	0.80	227
virtual	0.00	0.00	0.00	15
macro avg	0.65	0.50	0.54	1005
weighted avg	0.77	0.72	0.73	1005

Table 5.6: Classification report of Matrix 2.

in the new dataset. BERT1 was able to correctly classify 239 of them. BERT1 performs very well with the 'media' label responses; 72 out of 90 times it was able to be classified correctly. BERT1 faced the most difficulty classifying 'technology' label responses, which provided the lowest precision value of 0.73

Now consider the BERT2 model, which is an improved version of

BERT1, because BERT2 was trained with an augmented dataset, which greatly increases the volume of the dataset. For that reason, the BERT2 model was able to capture the semantic meaning of words in the dataset.

$$\begin{array}{r|cccccccccc|}
 & A & G & I & L1 & L2 & M & R & T1 & T2 & V \\
\hline
Augmented & 41 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
Generic & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Interactivity & 1 & 1 & 191 & 3 & 1 & 4 & 2 & 2 & 1 & 0 \\
Learning & 1 & 0 & 0 & 303 & 2 & 1 & 1 & 0 & 1 & 2 \\
Legit & 0 & 0 & 0 & 0 & 28 & 0 & 0 & 2 & 4 & 0 \\
Media & 0 & 3 & 1 & 0 & 0 & 85 & 0 & 0 & 0 & 1 \\
Representation & 0 & 1 & 0 & 0 & 1 & 1 & 41 & 0 & 0 & 0 \\
Technology & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 35 & 1 & 1 \\
Ttransmission & 1 & 0 & 2 & 1 & 1 & 0 & 0 & 1 & 221 & 0 \\
Virtual & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15
\end{array}$$

Matrix 2: Confusion Matrix when BERT2 tests with new dataset)

When BERT2 was tested with the new dataset, it was observed that 96 out of 100 times, the BERT2 model was able to classify the responses correctly. For example, in 303 out of 311 cases, BERT2 correctly classified the 'learning' label response, where BERT1 was only able to classify 239 cases, which shows the precision value of the 'learning' label was improved from 0.75 to 0.98. Also, it has improved the precision of the 'technology' label-based response, where BERT1 resulted in 0.73 but BERT provided 0.92. BERT2 overall provided the best precision, recall, and f1-score for almost all labels.

Accuracy: 0.955

Table 5.7: Classification report of Matrix 3.

	precision	recall	f1-score	support
augmented	0.93	0.91	0.92	45
generic	0.00	0.00	0.00	0
interactivity	0.98	0.93	0.96	206
learning	0.98	0.97	0.98	311
legit	0.85	1.00	0.92	28
media	0.91	0.94	0.93	90
representation	0.91	0.93	0.92	44
technology	0.92	0.90	0.91	39
transmission	0.99	0.97	0.98	227
virtual	0.79	1.00	0.88	15
macro avg	0.83	0.86	0.84	1005
weighted avg	0.96	0.96	0.96	1005

Overall, the BERT2 model, which is trained with an augmented preprocessed dataset, performs very well on the new dataset. The evaluation report indicates the necessity of augmentation and why this BERT2 model is better than proposed recurrent neural network models. Also, BERT2 is the final model for this thesis, and as per the evaluation report, it outperforms all other models in all cases, such as preprocessing, augmentation, etc.

6 Discussion

6.1 Implementation part in terms of the Background Theory chapter and Future work

The background theory chapter covers a few topics that are necessary for understanding how task classification in natural language processing works. Back translation, one of the augmentation processes, is briefly discussed in Section 3.2.4 but not introduced in the implementation section. Back translation is computationally costly, which means it requires a considerable amount of time to translate into one language and then re-translate into the original. Since employing the other mentioned data augmentation techniques on the dataset, the number of samples has increased to approximately 148k from 4.9k. To improve the quality of the retranslation also requires good amount of time. Future work would involve applying the back translation technique to the dataset and evaluating the quality of the re-translation using established metrics.

In the background theory chapter, three different types of text classification were discussed, while the implementation part only focused on one approach, the rule-based approach, where approaches such as hybrid systems are more useful. In the hybrid approach, rule-based and machine-based methods are combined to enhance the input for machine learning models by comparing them to one another. The reason it is not introduced in the implementation part is that to implement it, both rule-based and machine-based approaches should be implemented first, which is time-consuming. Also, only implementing a machine-based approach instead of a rule-based approach is not appropriate for the dataset because the initial dataset was small and carried responses to 33 questions. Therefore, if responses related to 33 questions were tagged as pre-labeled, then there would be 33 tags or labels for 4.9k sample-based datasets. The problem would be that each tag may not have enough data. The future work would be to collect more data on those responses, whether they were generated by humans or APIs, and then apply a hybrid system that is more accurate than the other two approaches.

Another future work would be to do Doc2Vec for embedding approaches and compare it to the Word2Vec results since the implementation focuses on Word2Vec.

6.2 *Evaluation approaches in terms of the Implementation part*

From the implementation part of this report, it is observed that both recurrent neural models such as long-short-term memory (LSTM) and BiLSTM provide significant accuracy in both the training and test phases. Both recurrent neural network models perform well in the training phase, but in the testing phase, both provide a high loss value, which cannot be improved because training every new cycle of the dataset increases the accuracy value close to the boundary, even though the loss value could not decrease that much. The loss values for both long-short-term memory (LSTM) and BiLSTM models are 0.624 and 0.712, respectively, for the testing phase, where accuracy is 0.8709 and 0.8771, respectively. This large value indicates that both models would not be able to classify the maximum text correctly.

In the evaluation part of this report, it is demonstrated that both of these recurrent neural network models failed to classify most of the texts. The LSTM model has failed to classify approximately 85 percent of the texts, whereas BiLSTM failed to classify almost 80 percent of the texts. The introduced evaluation approach with a new dataset successfully evaluated the results of both recurrent neural network models because both models indicated an overfitting issue in the implementation part. Thus, both models are not suitable for the chatbot because both face difficulties in checking whether messages refer to each other or not in the chat corpus.

This problem can be overcome with the help of Bidirectional Encoder Representations from Transformers (BERT). For augmented preprocessed datasets, the BERT model provides very high accuracy in terms of low loss values for both the training and testing sets. Both sets provide loss values less than 0.20. In the evaluation part of this report, it was discussed how the BERT model classified the maximum number of texts from the new dataset. There are two BERT models with the same structure introduced in this report. BERT1, which was trained on an unaugmented preprocessed dataset, and BERT2, which was trained on an augmented preprocessed dataset. In the implementation part, it has been found that the BERT2 model performs better than BERT1 by increasing accuracy by 6 percent and decreasing the loss value by 24 percent for the training set. When these two models are tested on a new dataset, the BERT2 model correctly classifies 24 percent more texts than BERT1. The BERT2 model, which was trained using an augmented preprocessed dataset, performs significantly better than the BERT1 model in both the implementation and evaluation phases. In terms of the evaluation report, the BERT2 model performs significantly better than any other model that is covered in this document. As a result, BERT2 is an appropriate choice for the final model when developing a chatbot.

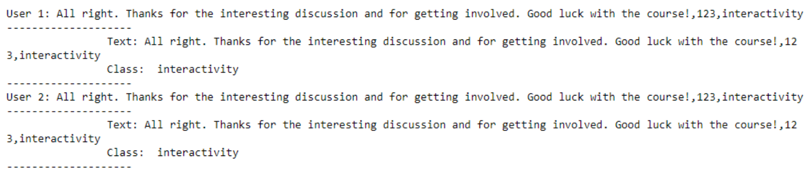
6.3 Model poisoning

Whenever messages do not refer to each other, chatbots show a warning to the user to let them know if they are correctly referring to each other or not. According to the learner's response, the chatbot will collect that message to improve the model. There could be an issue with model poisoning since the chatbot would collect data that was not classified correctly to update the model. If someone gave a biased or irrelevant response by saying that was the correct class, then the model would be trained with the wrong label or class. The model could be polluted by frequently feeding this type of incorrect class.

This issue can be overcome with the help of an anomaly detection algorithm using clustering. Clustering is an unsupervised technique, but it can still be applied to classification approaches. For example, based on the label of the task, it is possible to cluster responses. Text on each label serves as a cluster. To begin, each text or document is transformed into a vector with the assistance of the bag of words methodology. After that, it is not difficult to compute the distance that separates each data point from the center of its own cluster. In the end, a threshold is determined using the mean and standard deviation of the computed distance. Because of this, if a new message is sent to the model with data points that are bigger than the threshold, it will remove the data points. The final neural network model of this thesis can be protected against model poisoning in this way.

6.4 Jupyter interface and chatbot testing

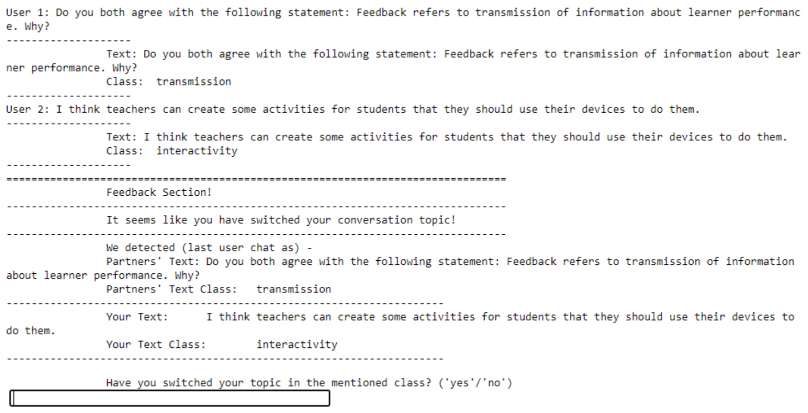
Jupyter notebook was used to generate a basic mockup to examine the basic chatbot functionality. In test case 1, messages sent by both users made reference to one another. Chatbots were able to classify those messages. It is also noticeable that both users are on the same topic.



```
User 1: All right. Thanks for the interesting discussion and for getting involved. Good luck with the course!,123,interactivity
--------------------
                Text: All right. Thanks for the interesting discussion and for getting involved. Good luck with the course!,12
3,interactivity
                Class:  interactivity
--------------------
User 2: All right. Thanks for the interesting discussion and for getting involved. Good luck with the course!,123,interactivity
--------------------
                Text: All right. Thanks for the interesting discussion and for getting involved. Good luck with the course!,12
3,interactivity
                Class:  interactivity
--------------------
```

Figure 6.1: Test Case 1

In test case 2, user 2 responds on a different topic to the question of user 1. Chatbot notices that user 1 and user 2 responses are not in the same class. Therefore, it has shown warnings such as 'Have you switched your topic in the mentioned class?'

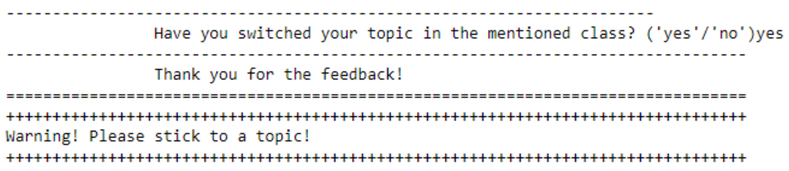


```
User 1: Do you both agree with the following statement: Feedback refers to transmission of information about learner performanc
e. Why?
--------------------
                Text: Do you both agree with the following statement: Feedback refers to transmission of information about lear
ner performance. Why?
                Class:  transmission
--------------------
User 2: I think teachers can create some activities for students that they should use their devices to do them.
--------------------
                Text: I think teachers can create some activities for students that they should use their devices to do them.
                Class:  interactivity
--------------------
================================================================================
                Feedback Section!
--------------------------------------------------------------------------------
                It seems like you have switched your conversation topic!
--------------------------------------------------------------------------------
                We detected (last user chat as) -
                Partners' Text: Do you both agree with the following statement: Feedback refers to transmission of information
about learner performance. Why?
                Partners' Text Class:   transmission
----------------------------------------------------------------------
                Your Text:      I think teachers can create some activities for students that they should use their devices to
do them.
                Your Text Class:        interactivity
----------------------------------------------------------------------

                Have you switched your topic in the mentioned class? ('yes'/'no')
```

Figure 6.2: Test Case 2

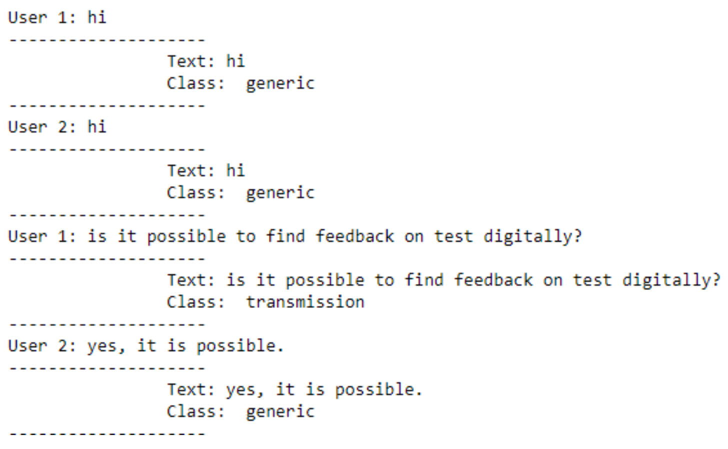
User 2, who confirmed, in response to the warning, that user 2 typed 'yes', which means user 2 changed the topic and had not correctly referred to the message sent by user 1. Then the chatbot tells user 2 to stay on topic. If user 2 responded 'no', the chatbot collected the user 2 response, which user 2 gave to user 1. Then the chatbot examines the threshold value that was derived based on the clustering results to determine whether or not the model should be updated.



```
----------------------------------------------------------------------
                Have you switched your topic in the mentioned class? ('yes'/'no')yes
--------------------------------------------------------------------------------
                Thank you for the feedback!
================================================================================
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Warning! Please stick to a topic!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 6.3: Test Case 3

Now, if user 2 responds with generic words to the question of user 1, the chatbot does not intervene because some of the questions can be answered generically, as in the example following, but the chatbot does inform user 2 that the response of user 2 is generic.



```
User 1: hi
--------------------
                Text: hi
                Class:  generic
--------------------
User 2: hi
--------------------
                Text: hi
                Class:  generic
--------------------
User 1: is it possible to find feedback on test digitally?
--------------------
                Text: is it possible to find feedback on test digitally?
                Class:  transmission
--------------------
User 2: yes, it is possible.
--------------------
                Text: yes, it is possible.
                Class:  generic
--------------------
```

Figure 6.4: Test Case 4

6.5 Final Thoughts on Result

Overall, It is notable that the final version of the BERT model can correctly categorize the vast majority of texts. The initial dataset comprises a large number of irrelevant or generic responses, and it is also insufficient in size. Despite the use of data augmentation techniques, the quality of the dataset has not improved. Because of this, both types of recurrent neural networks have a tendency to overfit. The BERT model, on the other hand, is a pretrained model that has around 3.3 billion tokens and 110 million parameters [18]. Therefore, due to the fact that the BERT model had previously been trained with a significant amount of data, it performed exceptionally well when the original augmented data was fed to it. Therefore, future work would be to collect more data that is straight to the point and try traditional neural networks instead of a pretrained model to see how it works with only original data.

7
Conclusion

This document ends with answering the research questions, limitations of this research, what changes have been made from the original proposal, and an ending note.

7.1 Answers to the research questions

A. How can the classification technique be used to classify responses from user to check whether he or she refers to another user or not?

   Initially, the dataset was coded based on the coding approach. Later, a validation of the coding approach is documented. After that, data preprocessing and augmentation were done. Finally, how rule-based models were applied to the dataset to classify messages is documented in this report.

B. How would the different tactics used by the different neural network models lead to different results?

   A few neural network models were applied to the same dataset, which is discussed in the implementation chapter, and how this algorithm works is also discussed in the background theory chapter. Because of the different strategies followed by these neural network approaches, differences in the results are observed, where Bidirectional Encoder Representations from Transformers (BERT) perform highly compared to other neural networks.

C. Why is it necessary to apply data augmentation techniques?

   It is shown how data augmentation techniques improve model accuracy and decrease the loss value. Without data augmentation, all models provide high loss values, which are improved with data augmentation techniques. Neural network models such as long-short-term memory (LSTM) and bidirectional (LSTM) perform very poorly by providing high loss values because both of these models are not pretrained.

D. How well does the evaluation strategy work in assessing the results?

The evaluation approach works well in relation to the implementation part. In the implementation, it has been observed that long-short-term memory (LSTM) and bidirectional LSTM (BiLSTM) tend to overfit. In the evaluation process, it has been shown that these two models performed poorly with the responses and were unable to classify most of the texts or responses. On the other hand, the Bidirectional Encoder Representations from Transformers (BERT) model performs well in both the implementation and evaluation parts.

7.2 Limitation and changes

The main limitation is the dataset. The dataset is overall complex because some of the texts have no proper meaning and there are so many generic responses than expected such as 'hi', 'hello', and 'bye', etc. The quality of the dataset is so poor that even data augmentation techniques cannot solve this problem for recurrent neural network models. That's why both recurrent neural network models tend to overfit, and less than 80 percent of the time they incorrectly classify texts.

Because of the limitation stated, a pretrained model named Bidirectional Encoder Representations from Transformers (BERT) is introduced, which is not in the proposal of this thesis, but it makes the overall implementation more time-consuming in an average graphical processing unit system. Therefore, updating the model with new unclassified responses would take time.

7.3 Ending Thoughts

Chapter one stated the context, motivation, research goal, research questions, and contributions. Chapter two described all the technical approaches that have been used in the implementation and evaluation parts theoretically so that the reader can understand the theoretical background before going into the implementation section. In the implementation part, a brief overview of the applied practical approach is documented. The evaluation part evaluates the neural network model results that were implemented in the implementation part. Finally, in the discussion chapter, future work as well as a thread on the neural network models have been documented.

This thesis has brought to attention how the BERT model outperforms other approaches to correctly classifying the texts in the original dataset. Apart from applying neural network algorithms, this document also reports how datasets have been validated before applying machine learning steps. At the end of the report, in the discussion section, a simple interface is demonstrated to show how messages refer to each other in the same class. When all of the effort is taken into consideration, it is possible to conclude that the solution that was built is satisfactory. One of the future tasks could be to import the neural network model into chat systems such as MS Teams, Discord, etc. There is also an opportunity for improvement. Instead of applying a pretrained model, it is best to use a traditional neural network model so that every update of the model for an incorrect classifier would take less time, but for that, a good amount of data is required.

However, the collaborative learning domain is huge. The dataset only covers some responses from some fields. It is possible to build a concrete chatbot in a higher domain with rich data, where collaborative learners can really benefit. When we see chatgpt trained with 570 gigabytes of text data, the dataset this research went through was a very tiny amount [41]. AI, like Chatgpt, introduces how text can be generated with a high percentage of accuracy. Therefore, in the future, if collaborative learning applications where it is possible to encourage learners by regularizing their discourse run with not gigabytes of data but a fair amount of data with the maximum number of topics in class, then it is really possible that collaborative learning can really impact the learner's efficiency.

Bibliography

[1] Sensory processing and the brain (article). Khan Academy., https://shorturl.at/csJO1.

[2] OpenAI API. API Reference. OpenAI API, https://platform.openai.com/docs/api-reference.

[3] M. Bramer. 2007. Principles of data mining. (Vol. 180, p. 2). London: Springer, https://link.springer.com/book/10.1007/978-1-4471-7493-6.

[4] P. Brians. 2013. Common errors in English usage. Franklin, Beedle Associates, Inc., https://shorturl.at/uvKQW.

[5] Carlander-Reuterfelt, Daniel, Carlos A. Iglesias Álvaro Carrera, Óscar Araque, Juan Fernando Sánchez Rada, and Sergio Muñoz. 2020. JAICOB: A data science chatbot. IEEE Access 8, https://ieeexplore.ieee.org/abstract/document/9200315/.

[6] Saad M. Khan David Edwards Chopade, Pravin and Alina von Davier. 2018. Machine learning for efficient assessment and prediction of human performance in collaborative learning environments. IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-6. IEEE, https://doi.org/10.1109/THS.2018.8574203.

[7] Boles W. Trouton L. Cunningham-Nelson, S. and E. Margerison. 2019. A review of chatbots in education: practical steps forward. In 30th annual conference for the australasian association for engineering education (AAEE 2019): educators becoming agents of change: innovate, integrate, motivate (pp. 299-306). Engineers Australia, https://eprints.qut.edu.au/134323/.

[8] Reddit Data. 2015. Publicly available Reddit comment for research. r/Datasets, https://shorturl.at/zGJP8.

[9] Karakostas A. Tsiatsos T. Caballé S. Dimitriadis Y. Weinberger A. Papadopoulos P.M. Palaigeorgiou G. Tsimpanis C. Demetriadis, S. and M. Hodges. 2018. Towards integrating conversational agents and learning analytics in MOOCs. The 6th International Conference on Emerging Internet, Data Web Technologies (EIDWT-2018) (pp. 1061-1072). Springer International Pub-

lishing, https://link.springer.com/chapter/10.1007/978-3-319-75928-9_98.

[10] Chang M.W. Lee K. Devlin, J. and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Google AI Research, https://arxiv.org/abs/1810.04805.

[11] Manyu Dhyani and Rajiv Kumar. 2021. An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. Mater Today Proc. 34, https://doi.org/10.1016/j.matpr.2020.05.450.

[12] Zijian Ding, Jiawen Kang, Tinky Oi Ting Ho, Ka Ho Wong, Helene H. Fung, Helen Meng, and Xiaojuan Ma. 2022. TalkTive: A Conversational Agent Using Backchannels to Engage Older Adults in Neurocognitive Disorders Screening. CHI Conference on Human Factors in Computing Systems, https://doi.org/10.1145/3491102.3502005.

[13] Dejana Diziol, Erin Walker, Nikol Rummel, and Kenneth R. Koedinge. 2010. Using intelligent tutor technology to implement adaptive support for student collaboration. Educational Psychology Review 22, no. 1: 89-102, https://link.springer.com/article/10.1007/s10648-009-9116-9.

[14] S. Fan. 2018. Understanding Word2Vec and Doc2Vec. https://shuzhanfan.github.io/2018/08/understanding-word2vec-and-doc2vec/.

[15] G.C. Feng. 2014. Intercoder reliability indices: disuse, misuse, and abuse. Quality Quantity, 48, pp.1803-1815, https://link.springer.com/article/10.1007/s11135-013-9956-8.

[16] P. Galdi and R. Tagliaferri. 2018. Data mining: accuracy and error measures for classification and prediction. Encyclopedia of Bioinformatics and Computational Biology, 1, pp.431-436, https://doi.org/10.1016/B978-0-12-809633-8.20474-3.

[17] N. Gupta. 2013. Artificial neural network. Network and Complex Systems, 3(1), pp.24-28, https://shorturl.at/bjLPU.

[18] C. Haley. 2020. This is a BERT. Now there are several of them. Can they generalize to novel words? In Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (pp. 333-341), https://aclanthology.org/2020.blackboxnlp-1.31/5.

[19] ilmoi. Jul 15, 2019. Poisoning attacks on Machine Learning. Towards Data Science, https://towardsdatascience.com/poisoning-attacks-on-machine-learning-1ff247c254db.

[20] Kim M. Harerimana G. Kang S.U. Jang, B. and J.W. Kim. 2020. Bi-LSTM model to increase accuracy in text classification:

Combining Word2vec CNN and attention mechanism. Applied Sciences, 10(17), p.5841., https://www.mdpi.com/2076-3417/10/17/5841/pdf.

[21] Simen Johnsrud and Silje Christensen. 2017. Exploring Cells and Context Approaches for RNN Based Conversational Agents. Master's thesis, NTNU, https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2451313.

[22] A. Krogh. 2008. What are artificial neural networks? Nature biotechnology, 26(2), pp.195-197, https://www.nature.com/articles/nbt1386.

[23] Alturki N. Alramlawi S. Kuhail, M.A. and K. Alhejori. 2023. Interacting with educational chatbots: A systematic review. Education and Information Technologies, 28(1), pp.973-1018, https://link.springer.com/article/10.1007/s10639-022-11177-3.

[24] M. Laal and M. Laal. 2012. Collaborative learning: what is it? Procedia-Social and Behavioral Sciences, 31, pp.491-495, doi:10.1016/j.sbspro.2011.12.092.

[25] Feng J.H. Lazar, J. and H. Hochheiser. 2017. Research methods in human-computer interaction. Morgan Kaufmann, eBook ISBN: 9780128093436.

[26] S. Lee and H.J. Kim. 2008. September. News keyword extraction for topic tracking. fourth international conference on networked computing and advanced information management (Vol. 2, pp. 554-559). IEEE, https://shorturl.at/iHTZ1.

[27] Livando N. Chandra W. Phan G. Lin, A. and A.M. Husein. 2023. Sentiment Analysis Of Hotel Reviews On Tripadvisor With LSTM And ELECTRA. Sinkron: jurnal dan penelitian teknik informatika, 8(2), pp.733-740, https://doi.org/10.33395/sinkron.v8i2.12234.

[28] Qiu X. Liu, P. and X. Huang. 2016. Recurrent neural network for text classification with multi-task learning. arXiv preprint: arXiv:1605.05101, https://arxiv.org/pdf/1605.05101.pdf.

[29] Nicola Manzini. 2017. Single hidden layer neural network. from: https://www.nicolamanzini.com/single-hidden-layer-neural-network/.

[30] M.L McHugh. 2012. Interrater reliability: the kappa statistic. Biochemia medica, 22(3), pp.276-282, https://hrcak.srce.hr/file/132393.

[31] J. Mohajon. 2021. Confusion Matrix for Your Multi-Class Machine Learning Model. Towards Data Science, https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

[32] Monalisa Dey Dipankar Das Sachit Nagpal Mondal, Anupam and Kevin Garda. 2018. Chatbot: An automated conversation system for the educational domain. International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), pp. 1-5, IEEE, https://ieeexplore.ieee.org/abstract/document/8692927.

[33] Antonio Justiniano Moraes Neto and Márcia Aparecida Fernandes. 2019. Chatbot and conversational analysis to promote collaborative learning in distance education. IEEE 19th International Conference on Advanced Learning Technologies (ICALT). Vol. 2161, https://ieeexplore.ieee.org/document/8820823.

[34] Charuta Pande, Hans Friedrich Witschel, Andreas Martin, and Devid Montecchiari. 2021. Hybrid Conversational AI for Intelligent Tutoring Systems. AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, https://proceedings.aaai-make.info/AAAI-MAKE-PROCEEDINGS-2021/paper23.pdf.

[35] Florian Peters. 2018. Design and implementation of a chatbot in the context of customer support. MS Thesis: University of Liège - Faculty of Applied Sciences, https://matheo.uliege.be/handle/2268.2/4625.

[36] Anika Radkowitsch, Freydis Vogel, and Frank Fischer. 2020. Good for learning, bad for motivation? A meta-analysis on the effects of computer-supported collaboration scripts. International Journal of Computer-Supported Collaborative Learning 15.1, https://link.springer.com/article/10.1007/s11412-020-09316-4.

[37] Md Moshiur Rahman, Ruhul Amin, Md Nazmul Khan Liton, and Nahid Hossain. 2019. Disha: An implementation of machine learning based Bangla healthcare Chatbot. 22nd International Conference on Computer and Information Technology (ICCIT), pp. 1-6. IEEE, https://shorturl.at/ruQZ7.

[38] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709, https://arxiv.org/abs/1511.06709.

[39] S. Sharma. 2022. Activation Functions in Neural Networks. Towards Data Science, https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[40] Sharma S. Sharma, S. and A Athaiya. 2017. Activation functions in neural networks. Towards Data Sci, 6(12), pp.310-316, https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf.

[41] Heacock L. Elias J. Hentel K.D. Reig B. Shih G. Shen, Y. and L. Moy. 2023. ChatGPT and other large language

models are double-edged swords. Radiology, 307(2), p.e230163, https://pubs.rsna.org/doi/10.1148/radiol.230163.

[42] Japkowicz N. Sokolova, M. and S. Szpakowicz. 2006. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings 19 (pp. 1015-1021). Springer Berlin Heidelberg, https://link.springer.com/chapter/10.1007/11941439_14.

[43] Raimo Streefkerk. April 18, 2019. Qualitative vs. Quantitative Research | Differences, Examples Methods. Scribbr, https://www.scribbr.com/methodology/qualitative-quantitative-research/.

[44] Timo Strohmann, Dominik Siemon, and Susanne Robra-Bissantz. 2017. brAInstorm: intelligent assistance in group idea generation. International Conference on Design Science Research in Information System and Technology. Springer, Cham, https://link.springer.com/chapter/10.1007/978-3-319-59144-5_31.

[45] Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196, https://arxiv.org/abs/1901.11196.

[46] Joseph Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM 9.1, https://dl.acm.org/doi/pdf/10.1145/365153.365168.

[47] S. Yan. Mar 13, 2016. Understanding LSTM and its diagrams. Medium, https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714.

[48] Moon J. Yang S. Oh H. Lee S. Kim Y. Yang, M. and J. Jeong. 2022. Design and implementation of an explainable bidirectional lstm model based on transition system approach for cooperative ai-workers. Applied Sciences, 12(13), p.6390, https://www.mdpi.com/2076-3417/12/13/6390.

[49] Zou Y. Zhang, X. and W. Shi. August, 2017. Dilated convolution neural network with LeakyReLU for environmental sound classification. 22nd international conference on digital signal processing (DSP) (pp. 1-5). IEEE, https://shorturl.at/cpGY9.