# Lib-Bot: A Smart Librarian-Chatbot Assistant

**Tong-Jun Ng[1], Kok-Why Ng[2] and Su-Cheng Haw[3]**

[1,2,3]*Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Malaysia*

*E-mail address: kwng@mmu.edu.my*

**Abstract:** Library is a knowledge warehouse and various long past references can be found in it. Students, professors, kids, and adults are regularly encouraged to visit the library as it provides a conducive environment for building the habit of reading books and improving individual critical-thinking skills. As technology is getting more and more advanced nowadays, some common problems faced by the librarians can be replaced by machines. For instance, the librarians may not be available all the time at the counter; reduction of physical contact due to Covid19 infection et cetera, machines can take over the librarians' roles to handle the tasks. In this paper, an Artificial Intelligence (AI) chatbot is proposed and implemented on mobile application to answer library-related questions. Bidirectional Encoder Representations from Transformers (BERT) algorithm is employed to classify the intent of the user's messages. Besides, many existing chatbot applications support only the text input. This paper proposes a speech-to-text recognition feature to enable both text and voice input. If there are any queries that cannot be solved by the chatbot system, it will store the queries in the database and the library admins can filter the queries and upload new training data for the AI model to cover a wider range of questions.

**Keywords:** Bidirectional Encoder Representations from Transformers, Library, Chatbot, Machine Learning, Intent Classification

## 1. INTRODUCTION

It is well known to us that Reading is a common behavior that every educated person cannot avoid, and the best place for reading is the library. People also often go to the library to complete research and work. Despite the rapid development of technology, physical books cannot be replaced by devices, and libraries will continue to play a vital role in the future. However, librarians may not be available all the time to answer visitors' queries.

In this day and age, AI chatbots have been getting more attention in daily life. Developing a chatbot assistant in the library can help avoid physical contact while maintaining user satisfaction. A chatbot can free up librarians' time and provide both text and voice input options for the user. However, the chatbot may not be able to answer every question, and its purpose is to provide fast and accurate solutions to common problems. If the chatbot is not comprehensive enough, it will only worsen the situation as users still need to ask librarians for solutions.

AI chatbot is a smart or intelligent computer program that can communicate with human and solve their problems, either providing general purpose or specific domain solu-tions by using natural language and some AI techniques to enhance natural language understanding (NLU) or process-ing (NLP).

In past research, it is worth noting that the ability to understand and respond to long questions accurately and efficiently with low response time is very crucial and always be the issue for chatbot development [1]. To address the queries quickly while still maintaining accuracy, chatbot needs to first be able to identify and understand the meaning or intention of user message given as fast as possible. A person is impossible to answer a question precisely and quickly without knowing the intention of the question, the same goes for chatbot development. Therefore, many past studies and research have been done, some traditional machine learning and deep learning algorithms like Sup-port Vector Machine (SVM), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), etc. have been used in implementing and tackling the intent classification task for chatbot development [2],[3],[4],[5]. However, the intent classification done in past studies does not achieve a satisfactory result and it is crucial for a chatbot to have a high accuracy intention identification ability. Therefore, this paper proposed Lib-Bot, an AI-powered library chatbot

assistant which can classify and identify queries intention using Bidirectional Encoder Representations from Transformer (BERT). The intent classification model of Lib-Bot has achieved an outstanding accuracy result which will be helpful for other natural language tasks and chatbot development for other domains.

In short, all the issues mentioned above have triggered the interest to develop Lib-Bot with the objectives below:

- To develop a library chatbot assistant for answering queries and recommending books based on user preferences.

- To integrate speech-to-text recognition for the chatbot in the application

- To create a dashboard at the admin portal site to show the chatbot's unsolved problems for future enhancements.

This paper is organized as follows: Section 2 reviews the chatbot related work done by past researchers and the comparison on existing chatbot-related applications in the market. Section 3 describes the proposed methodology and the overall architecture design of Lib-Bot in detail. Section 4 presents the results and discussions on the evaluation of Lib-Bot. Lastly, section 5 concludes the paper with some possible future works.

## 2. RELATED WORK

### A. Machine Learning & Deep Learning Techniques in Chatbot Development

Several machine learning and deep learning approaches existed in several domains [1],[6],[7]. This section reviews some of the techniques.

### 1) Naive Bayes

An intent classification model is used to identify the user's message intention for chatbot development [2]. The Naive Bayes classifier is used for the intent classification task. After training, the model is compared to the Logistic Regression intent classification model in terms of accuracy, precision, and recall using a confusion matrix. Performance evaluation is done using 20% of the training data as the test data. The Naive Bayes model achieves an accuracy of 63.63%, while the Logistic Regression model achieves a higher accuracy of 72.72%. The author points out that the Logistic Regression algorithm will have better performance compared to the Naive Bayes algorithm in the intent classification task of chatbot development.

The Naive Bayes algorithm is used to train an intent classification model for the chatbot. The chatbot developed in [8] is designed to answer queries related to hearing impairments from users in an interactive way. The performance of the intent classification model using the Naive Bayes algorithm is evaluated in terms of accuracy, precision, and recall using a confusion matrix. The testing result shows that the model achieves an accuracy of 88.75%, precision of 98.61%, and recall of 88.75%. The performance evaluation result indicates that the Naive Bayes algorithm works well in classification tasks.

An AI chatbot is developed to help students and their parents inquire about college details. As a result, students and parents can get instant replies regarding information without reaching the administration office of the college [9]. The Multinomial Naive Bayes algorithm is used in the system to classify the class of the user input. The user input is first pre-processed with tokenization, stemming, and lemmatization processes. After pre-processing, every word in the sequence is calculated for its repetition by using the Multinomial Naive Bayes classifier over the dataset. Scores are given to the input sentence, indicating the matching rate of the sentence with the classes. According to the classified class, the chatbot will respond to the user with an appropriate answer

### 2) Support Vector Machine (SVM)

The researchers proposed to build a medical chatbot using the SVM algorithm. The SVM algorithm is used in implementing a disease classification model for the medical chatbot [10]. The disease classification model will identify the disease and the chatbot will provide appropriate suggestions to the user based on the symptoms inserted by the user. The classification of SVM is done by finding the hyper-plane which separates the data points in n-D space into two classes. After the training of the SVM model, a performance evaluation is done to compare the accuracy of SVM, KNN and Naive Bayes algorithms in classifying 200 diseases. The researchers found out that SVM performs the best among the three classifiers with an accuracy of 92.33%. On the other hand, the Naive Bayes and KNN algorithms achieve an accuracy of 81% and 87.66% respectively.

The researchers aim to develop a chatbot to provide relevant information and medical assistance to the user according to their health status. [11] The researchers use an SVM algorithm to predict the illness based on the symptoms in the input sentence from the user. The chatbot will respond with detailed information related to that illness. Embedding is used to deal with the massive amount of data and ensure the efficiency of the training. In this paper, the performance of SVM in predicting illness is compared with KNN and Naive Bayes classifiers. The SVM algorithm achieves the highest accuracy of 93.66%, the KNN algorithm obtains the lowest accuracy which is 70% and the Naive Bayes algorithm gets an accuracy of 89.67%.

A COVID-19 chatbot is developed for citizens to solve questions related to COVID-19 and increase awareness towards the real danger of the COVID-19 outbreak. The SVM classifier is used in this paper to perform intent classification tasks [3]. The SVM model is responsible for identifying the intention of the user from the input messages and replying with relevant information. The authors mention that the SVM algorithm is chosen because it needs less

training to guarantee accuracy in classifying the intent as compared to other algorithms.

### 3) Recurrent Neural Network (RNN)

RNN is used to classify the intents of sentences in conversations [4]. The dataset used is a university guest book from the website which contains multiple kinds of visitors' comments. A total of 600 pieces of data are used for training and the data is divided into 5 different intent categories namely lecturing, admission, registration, administration, and comment. After preprocessing the data with tokenization and encoding, the data are validated by using the K-fold cross-validation (kFCV) technique. Next, the data are fed into the RNN model to train. The model successfully obtained an accuracy of 81% with the validation technique of k = 5. The author suggested adding more datasets to further improve the model.

A response-generating chatbot for the insurance industry, IntelliBot is proposed to overcome the shortcomings of existing chatbots. In [12], the author pointed out that most of the responses from the existing chatbots are not meaningful and only available for certain specific domain questions only. To quote an example, most chatbots are not able to answer questions beyond their domain such as "What is the time now?", "Are you female or male?" and so on. To tackle this issue, IntelliBot contains 4 different strategies namely template-based, knowledge-based, internet retrieval-based and generative-based to be chosen to respond to the user's queries. Suppose the template-based, knowledge-based and internet retrieval-based failed to generate an appropriate response. In that case, the user's queries will be handled by a generative-based strategy which uses a deep recurrent neural network (DRNN) based Seq2Seq model. The model also uses an attention mechanism in the encoding process to focus on key elements in the sentence during the model training. The model training is separated into three phases in which the dataset and number of training iterations are different in each phase. The dataset used consists of Cornell Movie dialogue and the credit card insurance dataset. The performance of IntelliBot is considered outstanding after calculating the F1 score and comparing it with other chatbots.

Bidirectional Recurrent Neural Network (BRNN) with attention layers is used so that the input sentence from the user with an enormous number of tokens or words can also be replied appropriately. In [13] research, with the utilisation of the TensorFlow library, a chatbot with Neural Machine Translation (NMT) model is developed. The author pointed out that an attention model can contribute to memorising long sequences. In addition, for models like BRNN, it is important to know the conversation context. Therefore, the attention mechanism is essential in solving such context problems which require both future and historical data. In this research, the dataset used to train for the chatbot is from the Reddit dataset which consists of comments in January of the year 2015. One of the most important

elements in the dataset is the score which indicates if the comment is an accurate reply to its parent comment. The training dataset contains around 3 million pairs of data. The initial perplexity, learning rate and bleu score of the model before training are 16322.15, 0.001 and 0.00 respectively. At the 23000th training step, the perplexity, learning rate and bleu score are 56.10, 0.0001 and 21.67. The sentiment-appropriate reply is the future model improvement in this research paper.

### 4) Long Short-Term Memory (LSTM)

Evebot, a fully generative sequence to sequence-based chatbot proposed to responsible for the diagnosis of negative emotions of the user and responding to some positive messages, solutions, or suggestions to reduce the impact of psychological distress such as depression, anxiety and so on to the user. Evebot [14] combined and utilised different deep learning models to deal with different tasks. The research paper discussed all the deep learning models used in the system. In the system, a Bi-directional LSTM (Bi-LSTM) based model was trained as a mood detection model to detect the negative emotion of the user. The bi-LSTM-based model has been proven to be more effective than the normal LSTM model as the LSTM algorithm can only train the sequences of the data from left to right whereas Bi-LSTM can train the neural network using both left-to-right and right-to-left sequences. The mood detection model trained will classify the input sentence as either positive or negative emotion. The sigmoid activation function is used for the model's output vector to produce a probability from 0 to 1 and those sentences having a probability smaller than 0.5 will be classified as having negative emotion. Moreover, the Bi-LSTM model is used again to train a text classification model for detecting if the input sentence is psychological-related or not. Next, the researchers also used an anti-language Seq2Seq neural network and Maximum Mutual Function (MMI) model in building the generative response model. This response model will generate answers according to the input instead of giving fixed responses. After training all the models, Evebot will first classify the input message from the user to identify if the user is in a positive or negative mood. If the input message is deemed positive, the responses will be generated from a casual chatbot. However, if the input message is deemed negative, the psychological counselling chatbot will take over and respond with some positive message or suggestion to the user. For the training result of the mood detection model, the Bi-LSTM model achieved a better precision which is 90.91% whereas LSTM achieved a precision of 87.74%. On the other hand, the Bi-LSTM model got 90.37% and the LSTM model obtained 87.56% in precision respectively for the psychological counselling-related text classification task. In short, the Bi-LSTM model has better performance as compared to the LSTM model.

The researchers aim to develop a canny chatbot which is dedicated to take the input message from the user and provide suggestions for them to reduce negative thoughts

or lower their pressure level. In [15], the Seq2Seq LSTM model is used in training a text classification model to classify from the input message whether the user is depressed or undepressed. The dataset is taken from Twitter and pre-processing such as removing special characters, tokenization, feature extraction, stemming, and making a word cloud and bag of words will be carried out on the dataset collected. The model is trained for 50 epochs with a learning rate of 0.1. The accuracy of the model is around 70%.

An educational chatbot is developed using the LSTM algorithm. As e-learning is becoming more popular in this era, the development of educational chatbots can help those e-learners to get instant responses to their questions instead of waiting for their tutors to answer them [5]. The LSTM algorithm is used to train a text classification model to classify the intent of the input sentence. After that, the chatbot will get the corresponding responses according to the intent classified and reply to the user. The performance of the model is identified by using a confusion matrix and compared with an existing model system on 100 test data. The result for the existing system is TN = 6, TP = 80, FN = 7 and FP = 80 whereas the proposed system is getting the result of TN = 0, TP = 100, FN = 0 and FP = 0.

An AI chatbot is developed to answer bus-related queries from the user by considering the user's current location and destination. The chatbot will provide information about the bus such as its name and number to the user so that the user can travel to their desired location safely [16]. The proposed methodology to implement the AI chatbot is by utilising Natural Language Processing (NLP) and LSTM algorithm. Natural Language Toolkit (NLTK) is used to perform some pre-processing on the dataset such as tokenization, stemming, lemmatization, stop words, chunking and so on. The dataset used is taken from the bus information database in Vijayawada city. The dataset consists of all the intents, the patterns of the intents and the responses to the intents. The deep learning model trained consists of 3 dense layers and some dropout layers are also introduced to avoid the overfitting of the model. The epoch for the training is 200 and the optimizer used is SGD. After the training, a graphical user interface is created for the chatbot. In this research paper, a performance comparison is carried out to compare the performance of the LSTM algorithm with some traditional algorithms such as Multinomial Naïve Bayes and Restricted Boltzmann algorithm. The proposed LSTM model achieved an accuracy of 95.02% whereas Multinomial Naïve Bayes and Restricted Boltzmann obtained an accuracy of 76.57% and 79.30% respectively. It shows that deep learning techniques such as LSTM perform better than traditional machine learning algorithms.

An educational chatbot with an LSTM algorithm for answering questions from the students to reduce the burden of teachers during the pandemic as all classes are conducted online during the COVID-19 pandemic is proposed. In [17], the chatbot is designed for MATLAB practical dataset. The dataset consists of the tag, which is the keyword for the question, the pattern of the questions, the response for the questions and the context related. For the data pre-processing, processes such as word tokenization, lowercase, lemmatization, word pickle and a bag of words will be carried out before the training. The first layer of the LSTM model consists of 128 neurons whereas the second layer contains 64 neurons. Some dropout layers are also added to the model to avoid overfitting of the model. The learning rate is 0.01 and the activation function used at the output layer is Relu. The performance analysis in the research paper indicates that the proposed LSTM algorithm has a better overall performance as compared to the Naive Bayes classifier and J48 classifier algorithms. The researchers developed a sentiment analysis model by using LSTM so that the chatbot can detect depression when chatting with the user. The authors [3] use LSTM instead of RNN in depression detection because RNN is difficult to train due to the exponential growth caused by repeatedly multiplying gradients. The researchers also compared the performance of LSTM with the Gated Recurrent Unit (GRU) model. The LSTM model obtains an F1 score of 80.78% and an accuracy of 92% whereas the F1 score, and accuracy achieved by the GRU model are 77.75% and 90% respectively. The result shows that LSTM has a better performance than the GRU model.

*5) Bidirectional Encoder Representations from Transformer (BERT)*

A smart AI chatbot is proposed to provide answers to COVID-19-related questions from the user [18]. The chatbot will first classify the input sentence from the user to identify which category is the sentence. After the text classification, the chatbot will respond according to the text category. This text classification task is done by using BERT as the performance of BERT in dealing with NLP tasks is outstanding as compared to other traditional machine learning and neural network algorithms. The traditional machine learning models are not able to capture the sequential context of the text input and they only perform text classification by using the keywords inside the text instead of using the question's meaning. The author also stated that training RNN and LSTM models on a very huge dataset will take a long time as compared to BERT because RNN and LSTM cannot be parallelized and only one input is accepted at a time. The accuracy of the proposed BERT text classification model is compared with the other classifiers such as K-nearest Neighbours (KNN), SVM, Naive Bayes and Decision Tree. The BERT model obtains the highest accuracy of 96% whereas the KNN model achieves the lowest accuracy of 43.33%. On the other hand, SVM, Naive Bayes and Decision Tree models get an accuracy of 72.2%, 75.5% and 85.7% respectively.

A BERT model is implemented to generate accurate answers from the user's questions by using Conversational

Question Answering (CoQA) dataset. This research paper used the CoQA dataset to fine-tune the pre-trained question-answering BERT model from the Hugging Face [19]. The author chooses the CoQA dataset because it contains both interrelated questions and passages for conversational reading comprehension. The intention of the questions asked by the user will first be classified by using the Google DialogFlow chatbot. After classifying the intention, if the answer to the question is based on a large context such as an article or a document, the BERT question-answering model will be loaded and generate suitable responses to the user.

A promising result shown when the author uses the BERT model to train on the dataset prepared. For the BERT model training, the model shows a very high accuracy which is 95% for the scenario corresponding to one question in which 10 responses are pre-selected randomly and one of the responses is the correct answer [20]. However, the performance of the model is not satisfactory when it comes to the one-on-one question-answering pair. The author points out that the BERT model can be further improved by increasing the size of tokens according to the input. Moreover, the model can also be enhanced by only providing a dataset of multiple-on-one for the question-answering pairs.

The researchers present the development of a financial service chatbot with the name AVA. In [21], the BERT model is used to develop the chatbot so that it can understand and answer customer queries with high accuracy. AVA can classify 381 intents after the training. Performance comparison on the intent classification between the BERT model and other popular model architectures is also evaluated. The large version BERT model with Google embeddings achieves the highest accuracy which is 95.4% whereas Naive Bayes is getting the lowest accuracy of 66.1%. The performance of the other models such as LSTM, Logistic Regression and XGBoost are all lower than the BERT model. The authors also apply sentence completion in the chatbot development to maintain the performance of the intent classification model even though the input has some spelling errors. AVA is also able to provide answers for queries that were not present in the training dataset, which is an indication of the chatbot's ability to generalise to new scenarios.

The researchers proposed a methodology to train a generative-based chatbot by using a small dataset which contains the question and answers pairs. The dataset used is created manually and it is domain-specific in that it consists of 567 question-and-answer pairs about the products, prices, languages, and technologies used in the company [22]. The performance of different encoder-decoder architectures such as stacked LSTM, and BiLSTM and different embedding types such as one-hot, fastText and BERT are investigated and discussed in the paper. The authors find out that stacked LSTM and BiLSTM encoder architectures and BERT embedding vectorization perform the best as compared to the

other methods when the dataset is small.

*B. Existing Chatbot-Related Applications*

Woebot [23] is a therapy chatbot developed by Alison Darcy and it was implemented into an IOS application in 2018. The objective of developing Woebot is to help those people who think that they are having some sort of anxiety and depression symptoms. Woebot can respond to consumers with predefined responses by using cognitive-behavioral therapy. Moreover, Woebot also suggests some solutions to deal with the depression emotion of the user according to their problems. However, Woebot does have some limitations. Woebot cannot fully understand what the user is trying to say. In other words, Woebot can only recognise some fixed inputs and respond to the inputs with certain predefined responses and flows. After picking one of the chat topics provided, the user needs to follow the fixed flow and the reply will be provided with a few options instead of typing by the user.

HelloFresh [24] provides creative solutions for those who do not have time to learn how to cook. They provide recipes to cook some healthy foods from scratch and answer messages related to the recipe from their customers. Freddy is responsible for automatically sending messages to those customers who commented on HelloFresh's posts. The user just needs to type in certain ingredients or a recipe for Freddy to find relevant information on HelloFresh's blog website. Moreover, Freddy also offers some entertainment activities such as quizzes or riddles related to food. The development of Freddy has successfully increased overall user satisfaction and decreased 76

One of the famous restaurant sales chatbots in Malaysia is Domino's chatbot [25]. It is implemented on Facebook Messenger to perform pizza ordering from any location. The consumer can also personalise their ordered pizza with the chatbot. Domino's chatbot makes the ordering and delivery processes more efficient. However, Domino's chatbot is more likely to be a rule-based chatbot as the consumers are required to choose the fixed reply options provided by the chatbot instead of typing by themselves for specific questions. The pizza ordering process will have a fixed flow for the customer to follow.

The next chatbot to be introduced is Wysa which was created in 2016 [26]. Same as Woebot, Wysa also applies evidence-based cognitive-behavioural techniques (CBT) in aiding mental therapy. To capture the user's current emotions and provide the most suitable assistance, Wysa will get the user's current feelings before going into the chat. Furthermore, Wysa will often ask users about their current emotions and opinion to the response of the chatbot during the chat. This is to ensure that the chatting topics and responses from Wysa can truly help the user improve their moods. Unlike Woebot, Wysa does not require the user to follow the fixed flow when chatting on certain topics.

Amy is a virtual assistant chatbot of HSBC bank Hong

Kong [27]. It is available on the website of HSBC bank for their customers to get instant answers to some common questions related to the bank. The queries that can be answered by Amy are mainly on the products or services provided by HSBC bank. Amy is also embedded with customer feedback functionality on its responses to allow it to learn more knowledge over time so that it can answer a wider range of queries. Amy is not only able to recognise English input, but it can also recognise Traditional and Simplified Chinese as well. Some significant actions in the bank such as transferring money are not possible to perform by just chatting with Amy.

The last chatbot-related application to be introduced is Andy which is a very popular educational chatbot application [28]. Andy is a goal-oriented chatbot developed by Andrey Pyankov in 2016. Andy is implemented to be an English teacher to help the users in learning or improving their English skills. Andy is an English-speaking bot that can communicate with users by following their proficiency in English after the assessment. The user can learn new vocabulary from Andy during the chat. The most attractive feature of the application which makes Andy to be outstanding among other similar applications is the text-to-speech functionality provided. All the text responses from Andy can be converted to audio. As a result, the users are not only able to improve their English vocabulary but also enhance their English pronunciation skills by listening to the responses.

Table I below summarizes the comparison of features and functionalities provided by the six existing chatbot-related applications in the market with Lib-Bot. Lib-Bot does not support multiple languages currently as the training data used are all in English.

### 3. Methodology

#### A. Lib-Bot Application Development

The training of the intent classification model is done by using JupyterLab with version 3.2.1 in Anaconda. The version of the TensorFlow library used is 2.5.0. Since Lib-Bot is targeted to develop on both iOS and Android platforms, Flutter would be the go-to option for application development. This is because Flutter only requires the developer to write and maintain a single codebase for the two mobile platforms. It is undoubtedly true that writing with a single codebase can help to save the cost, effort, and time needed for development.

For the data storage of the system, MySQL, an open-source relational database management system will be used. The Lib-Bot application cannot communicate with the database directly. Instead, it will access the database through the backend server by making API calls.

In order to generate chatbot responses, the intent classification model and question-answering model are required to be loaded and some pre-processing on the input sentence will be carried out at the backend site. Hence, the program-ming language chosen for the backend site is Python. A Python flask application will be developed as the admin portal website for the application. The Flutter application will need to make API calls to the backend functions to get data from the database or generate chatbot responses.

#### B. Intent Classification Model

#### 1) Bidirectional Encoder Representations from Transformer (BERT)

As BERT is decided to be used in classifying the intent of the messages from users and generating answers to the users, it is a must to further study about BERT. BERT is a deep-learning language model from Google that makes use of the Transformer model. The transformer model applied an attention mechanism which connects every output element with all the input elements and evaluates the weightings between the elements according to their connections. Generally, all the language models in the past were directional models which were only able to deal with the text input sequentially from left to its right or from right to the left. Nevertheless, the introduction of BERT has started up a new generation in Natural Language Processing (NLP). BERT applied bidirectional training of the Transformer model in which the transformer encoders can deal with the text input by reading the entire sequence of words of the input at once, meaning that the context of the words before or after a given word is taken into consideration instead of just considering of the word itself. As BERT is primarily for language modelling, the decoder is not important whereas the encoder is necessary to read the text input.

BERT from Google has been trained on a very large text dataset from Wikipedia as well as Book Corpus which contains a huge number of words to overcome the tasks in NLP. Named Entity Recognition (NER), text translation, question answering, and text classification are some of the NLP problems that can be solved by applying the BERT algorithm. BERT is pre-trained by applying two training strategies which are Masked LM (MLM) and Next Sentence Prediction (NSP).

MLM training is to hide 15% of the words in the input word sequences by replacing those words with the [MASK] token. The model is required to predict what is the original value of those masked words according to the context of the hidden words in the sequences. On the other hand, NSP training is to predict if the second sentence is the subsequent sentence of the first sentence after receiving pairs of statements as the input. For NSP training, half of the inputs are pairs of sentences in which the second sentence is the subsequent sentence of the first statement whereas the two sentences have no relationship at all for the remaining half of inputs.

As shown in Figure 1, the "[CLS]" token should be added in front of every input sentence while "[SEP]" is the separator token inserted at the end of each input sentence. The input sequence of tokens will then be embedded into

TABLE I. Comparison of existing chatbot-apps with Lib-Bot

| Features and Functions | Woebot | Freddy | Domino's Chatbot | Wysa | Amy | Andy | Lib-Bot |
|---|---|---|---|---|---|---|---|
| Free of charge | Y | Y | Y | Y | Y | Y | Y |
| Text-to-speech Recognition | N | N | N | N | N | Y | Y |
| Speech-to-text Recognition | N | N | N | N | N | N | Y |
| Provide quick recommended replies option for the user | Y | Y | Y | Y | N | Y | Y |
| Attractive user interface | Y | Y | Y | Y | N | N | Y |
| Having limit on the length of input messages | N | N | N | N | Y | N | Y |
| Allow typing all the time even though certain chat topic is started | N | Y | Y | N | Y | Y | Y |
| Support multiple languages | N | N | N | N | Y | N | N |
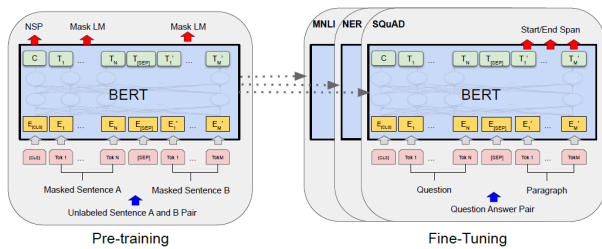


Figure 1. Overview of the BERT Model [16]



Figure 2. BERT of base and large version [29]



Figure 3. Head of the training dataset

vectors and processed in the neural network. It shows the overall procedures of BERT's pre-training and fine-tuning including the training of MLM as well as NSP. The architecture used in both pre-training and fine-tuning models is the same except for the output layers. Some text classification tasks such as sentiment classification can be solved by adding a classification layer on top of the transformer's output layer.

Model size of BERT (see Figure 2) can be determined by looking at the encoder blocks in the stack, its hidden units and attention heads. For instance, the base version of BERT has 12 encoder blocks while the large version has 24 encoder layers. BERT also has 768 and 1024 hidden units as well as 12 and 16 attention heads for the base and large versions respectively.

*2) Dataset*

The dataset used to train the BERT intent classification model is obtained manually as no standard or conventional data was available on the internet. The dataset consists of two columns namely text and intent. The text column contains the sentence, and the intent column consists of the intent of that sentence. Figure 2 shows the head of the training dataset. Before feeding the data into the model, some pre-processing on the data needed to be carried out. First, the text in the data will be tokenized into a sequence of tokens. Next, "[CLS]" and "[SEP]" will be added at the beginning and the end of the sequence of tokens respectively as they need to be inputted into the BERT model later. CLS stands for classification which is dedicated to making the pooling scheme work for classification tasks. The sequence of tokens will then be converted into numbers which represent the words. As all input sequence data will need to be the same length, the maximum length of the sequence in the data will first be identified. Padding will be given for those sequences with lengths shorter than the maximum length of the sequence in the dataset. Extra 0 will be added to the sequences to make all sequences have the same length as the maximum sequence's length.

*3) Question Answering Model*

Question answering (QA) model is to retrieve the answer from a given text or document. After classifying the intent

```
text = "The operating hour of the library is from 8 am to 6 pm."
question ="what is the closing time of the library?"

input_text = question + " [SEP] " + text
input_ids = tokenizer.encode(input_text)

print(input_text)
print(input_ids)
print(tokenizer.decode(input_ids))

questionContext = [0 if i <= input_ids.index(102) else 1 for i in range(len(input_ids))]

print(questionContext)

What is the closing time of the library? [SEP] The operating hour of the library is from 8 am to 6 pm.
[101, 2054, 2003, 1996, 5494, 2051, 1997, 1996, 3075, 1029, 102, 1996, 4082, 3178, 1997, 1996, 3075, 2003, 2013, 1022, 2572, 2000, 1020, 7610, 1012, 102]
[CLS] what is the closing time of the library? [SEP] the operating hour of the library is from 8 am to 6 pm. [SEP]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Figure 4. Input processing

of the message, if the intent classified does not require performing any database query, the message and context of the intent will be inputted into the QA model and generate the response. If no response is found, the default response of the intent will be replied to the user.

For this project, a pre-trained uncased BERT question-answering model from Hugging Face is used. This model is slightly different from other BERT models in that it is pre-trained by using the Whole Word Masking method. This means that instead of just masking the words randomly, all the tokens relevant to the masked word will be masked together as well. However, the masking rate will remain the same, which is 15%. After pre-training, it is fine-tuned on the SQUAD dataset.

The question-answering model and its tokenizer from Hugging Face will first be loaded into the application. Next, the pre-processing of the input of context and question will be carried out. The question and context will be concatenated together with the symbol "[SEP]" placed between them. The combined text will then be tokenized and encoded. The "[CLS]" and "[SEP]" symbols are added to the beginni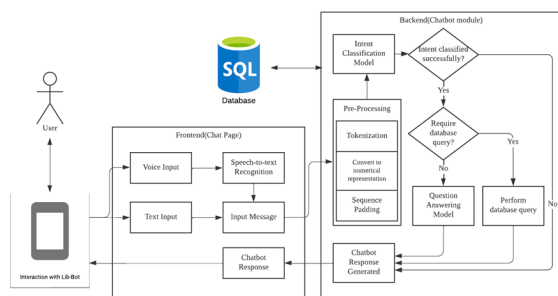ng and end of the text respectively. The number 101 represents "[CLS]" whereas 102 represents "[SEP]". Lastly, the question text which is before number 102 will be replaced with 0 and the context will be replaced with 1. The pre-processing on the input can be seen in Figure 4.

The input is then passed into the model to perform question answering. Figure 5 shows the output from the model. The start and end indices of the answer can be extracted from the output. The highest score in the output NumPy array indicates the index of the answer. If the start and end indices are both negative or the end index is larger than the start index, the response generated will not be accepted and the default response will be taken as the response for the user question. Figure 6 displays the result of answering the question "What is the closing time of the library?" based on the context of "The operating hours of the library are from 8 am to 6 pm.". The answer from the QA model is 6 pm. Getting an answer from the QA model will avoid the responses to be fixed and uninteresting.

*4) Proposed Chatbot's Architecture*

Two models will be used in the chatbot algorithm which is the Intent Classification Model and Question Answering Model. Bidirectional Encoder Representations from Transformers (BERT) will be used for both the intent

```
input = tf.constant(input_ids)[None, :]

answer=model(input, token_type_ids = tf.convert_to_tensor([questionContext]))

print(answer)

startScores, endScores = answer.start_logits, answer.end_logits

TFQuestionAnsweringModelOutput(loss=None, start_logits=<tf.Tensor: shape=(1, 26), dtype=float32, numpy=
array([[-5.703325 , -2.323728 , -6.413571 , -5.4685645 , -5.3917212 ,
        -7.8254085 , -7.481828 , -7.9239974 , -8.385733 , -9.143939 ,
        -5.703163 , -0.85186297, -2.2231588 , -3.9094248 , -6.0990434 ,
        -4.286521 , -4.8347282 , -4.7061877 , -1.4148403 , 1.5943834 ,
        -3.119761 , -3.0019138 , 6.483099 , 1.0461985 , -5.7026663 ,
        -5.703153 ]], dtype=float32)>, end_logits=<tf.Tensor: shape=(1, 26), dtype=float32, numpy=
array([[-0.64340675, -1.521875 , -4.417428 , -5.950923 , -5.037623 ,
        -4.2009077 , -6.0414004 , -6.9365425 , -6.043441 , -5.7205544 ,
        -0.6431502 , -5.996426 , -4.863099 , -2.8912356 , -5.925269 ,
        -6.004697 , -4.6548595 , -6.023759 , -6.139757 , -3.1029146 ,
        -0.51494735, -4.7417955 , 3.250334 , 9.355533 , -0.6449355 ,
        -0.64373124]], dtype=float32)>, hidden_states=None, attentions=None)
```

Figure 5. Output get from QA model

```
input_tokens = tokenizer.convert_ids_to_tokens(input_ids)

startIdx = tf.math.argmax(startScores[0],0).numpy()
endIdx = tf.math.argmax(endScores[0],0).numpy()+1

if (startScores[0][startIdx] < 0 and endScores[0][endIdx] < 0) or endIdx <= startIdx:
    print("no answer")
else:
    print(" ".join(input_tokens[startIdx:endIdx]))

6 pm
```

Figure 6. Output get from QA model

classification model and the question-answering model. Figure 7 depicts the architecture diagram.

The input messages from the user can either be through voice or text. The message text will then be passed to the backend server by making API calls. The first task of the chatbot is to identify the intent of the user. After passing the message text, the intent classification model will be loaded at the backend site. The message text will be preprocessed before putting into the model for detecting the intent. After detecting the intent of the message, the result will return to the application. When the user first enters the chat page, the application will obtain all intent data from the Intent entity in the database. All the details of the intent such as the type of intent, the context of intent, and default responses of intent will be retrieved. Next, the application will determine the next action based on the type of intent. If the model is not able to identify the intent of the message, the chatbot will respond to the user that it does not understand the question.

There are two types of intents in which one requires a database query and another one does not require a database query. If the intent of the message requires a database query, the application will make API calls to the backend site to perform the query. The result returned will be formulated and displayed to the user as the response. On the other hand, if the intent does not require any database query, the context of the intent and the message text will be passed again to the backend site and the Question Answering Model will be loaded at the backend site. The context and message will be input into the Question Answering Model after preprocessing. The message text from the user acts as the question and the Question Answering Model will find the answer to the question from the context and return the

Figure 7. Overall design of Lib-Bot



Figure 8. Graph of Model Accuracy

result back to the application. If the model can answer the question, the result will be displayed as the response from the chatbot. However, if the model is not able to get an answer from the context, the response will be taken from the default response of the intent in the database.

For those questions that could not be answered by the chatbot, the Lib-Bot application will store the unsolved query in the database. These queries can be viewed in the admin portal to be further analyzed by the library admin. According to those unsolved queries, the admin can insert new appropriate intent into the database so that the chatbot is able to answer such queries. In the admin portal, the admin can also upload a CSV file which contains new training data for the new intent. The existing Intent Classification Model can be trained again so that it is able to identify the new intent. By doing this, the chatbot can further improve and answer different types of queries.

## 4. RESULTS AND DISCUSSIONS

The intent classification model consists of 12 encoder layers, 768 hidden units and 12 attention heads which is the base version of the BERT model. Two dropout layers are added to the model to avoid overfitting of the model. The activation function used is SoftMax to classify the intent. The optimizer used is Adam and the loss function is categorical cross-entropy. A validation split of 10%, batch size of 32 and an early stopping with a patience of 3 is also applied. Figure 8 and Figure 9 display the model accuracy and loss in graphical representation. The BERT intent classification model achieved an accuracy of 99.14% over 10 epochs.

The BERT intent classification model built is working fine after implementing into the backend application site. As a matter of fact, adding more intents and data will help in assessing the user in much better ways. Therefore, the admin can review those queries that are unsolvable by the chatbot and add in more intents and data so that the chatbot can classify intents with higher accuracy and precision. As Lib-Bot is generating the answer through the question answering model. Hence, questions with the same intention may not be getting the same output from the chatbot. The responses will seem more specific and flexible. Figure 10
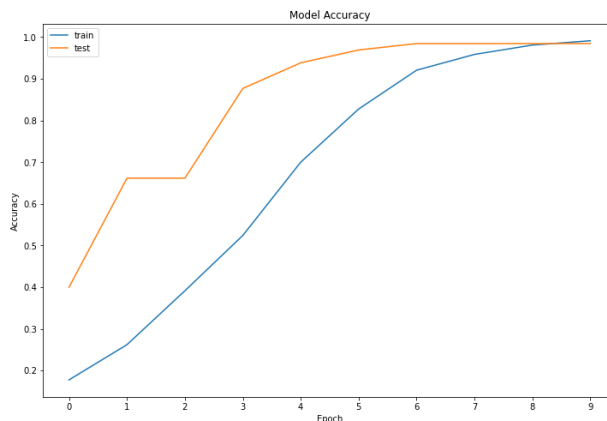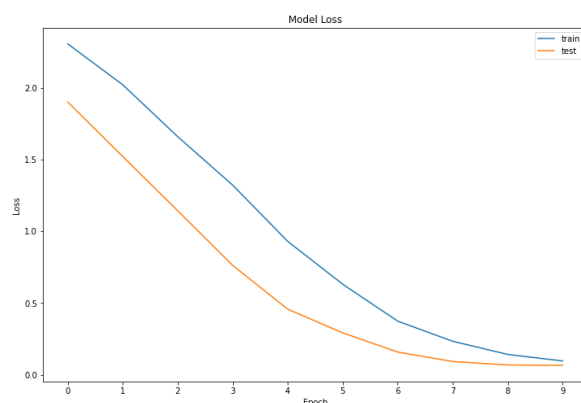


Figure 9. Graph of Model Loss

shows the example of communication between Lib-Bot with the user.

## 5. CONCLUSION AND FUTURE WORK

Chatbots is a new trend in how people communicate. It is undoubtedly true that chatbots have been widely applied in various domains nowadays. This is because applying AI chatbots in our daily lives can help us to get instant information without waiting for the real person to take over and answer. To achieve the objectives stated in this paper, various machine learning and deep learning algorithms have been surveyed and explored to select the best fit algorithm for the project. Furthermore, this paper also introduces multiple chatbot-related applications available nowadays in several fields. The overview and usage of these applications are being discussed as well.

An AI chatbot, Lib-Bot, is implemented in this project by using the BERT algorithm for both the intent classification model and the question-answering model in the chatbot architecture. A Flutter library mobile application has also been developed and integrated with the chatbot to achieve the objectives and solve the problems mentioned in this report. There are many ways that a chatbot can be
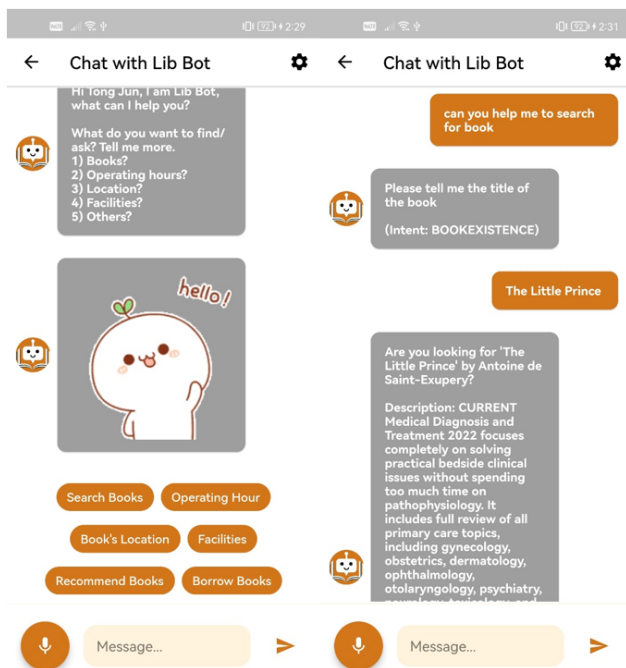
Figure 10. Communication with Lib-Bot

implemented and a lot of research and analysis are required to be done to find out the most suitable solution. The final accuracy achieved by the BERT intent classification model is 99.14%. The result indicates the outstanding performance of BERT algorithm in text classification task.

Throughout this paper, several limitations have been discovered. Firstly, the library-related dataset for training the model is difficult to find on the Internet. With a larger dataset, the performance of the intent classification can be further improved. In addition, the chatbot does not have a context memory which means that the chatbot cannot relate the pronouns in the current context with the previous messages. Hence, a reinforcement learning approach which can enable the chatbot to have contextual memory in the conversation can be a possible future work to further enhance the chatbot.

## REFERENCES

[1] P. Naveen, S. C. Haw, D. Nadthan, and S. K. Ramamoorthy, "Improving chatbot performance using hybrid deep learning approach," *Journal of System and Management Sciences*, vol. 13, no. 3, pp. 505–516, 2023.

[2] M. Y. H. Setyawan, R. M. Awangga, and S. R. Efendi, "Comparison of multinomial naive bayes algorithm and logistic regression for intent classification in chatbot," in *2018 International Conference on Applied Engineering (ICAE)*. IEEE, 2018, pp. 1–5.

[3] N. Ouerhani, A. Maalel, H. B. Ghézala, and S. Chouri, "Smart ubiquitous chatbot for covid-19 assistance with deep learning sentiment analysis model during and after quarantine," 2020.

[4] A. Prasetyo and H. A. Santoso, "Intents categorization for chatbot development using recurrent neural network (rnn) learning," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2021, pp. 51–55.

[5] E. Kasthuri and S. Balaji, "Natural language processing and deep learning chatbot using long short term memory algorithm," *Materials Today: Proceedings*, vol. 81, pp. 690–693, 2023.

[6] S. T. Lim, J. Y. Yuan, K. W. Khaw, and X. Chew, "Predicting travel insurance purchases in an insurance firm through machine learning methods after covid-19," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 43–58, 2023.

[7] Z. Yujiao, L. W. Ang, S. Shaomin, and S. Palaniappan, "Dropout prediction model for college students in moocs based on weighted multi-feature and svm," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 29–42, 2023.

[8] M. Anggraeni, M. Syafrullah, and H. A. Damanik, "Literation hearing impairment (i-chat bot): Natural language processing (nlp) and naïve bayes method," in *Journal of Physics: Conference Series*, vol. 1201, no. 1. IOP Publishing, 2019, p. 012057.

[9] G. S. S. Vikas, I. D. Kumar, S. A. Shareef, B. R. Roy, and G. Geetha, "Information chatbot for college management system using multinomial naive bayes," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2021, pp. 1149–1153.

[10] B. Tamizharasi, L. J. Livingston, and S. Rajkumar, "Building a medical chatbot using support vector machine learning algorithm," in *Journal of Physics: Conference Series*, vol. 1716, no. 1. IOP Publishing, 2020, p. 012059.

[11] R. Reshma *et al.*, "An improved chatbot for medical assistance using machine learning," in *2022 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2022, pp. 70–75.

[12] M. Nuruzzaman and O. K. Hussain, "Intellibot: A dialogue-based chatbot for the insurance industry," *Knowledge-Based Systems*, vol. 196, p. 105810, 2020.

[13] M. Dharani, J. Jyostna, E. Sucharitha, R. Likitha, and S. Manne, "Interactive transport enquiry with ai chatbot," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2020, pp. 1271–1276.

[14] J. Yin, Z. Chen, K. Zhou, and C. Yu, "A deep learning based chatbot for campus psychological therapy," *arXiv preprint arXiv:1910.06707*, 2019.

[15] D. S. Chauhan, A. Rastogi, and K. Singh, "Depression chatbot using deep learning," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022, pp. 1294–1297.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] E. Kasthuri and S. Balaji, "A chatbot for changing lifestyle in education," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. IEEE, 2021, pp. 1317–1322.

[18] E. Amer, A. Hazem, O. Farouk, A. Louca, Y. Mohamed, and

M. Ashraf, "A proposed chatbot framework for covid-19," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 2021, pp. 263–268.

[19] N. Kanodia, K. Ahmed, and Y. Miao, "Question answering model based conversational chatbot using bert model and google dialogflow," in *2021 31st International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2021, pp. 19–22.

[20] Q. Xie, D. Tan, T. Zhu, Q. Zhang, S. Xiao, J. Wang, B. Li, L. Sun, and P. Yi, "Chatbot application on cryptocurrency," in *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. IEEE, 2019, pp. 1–8.

[21] S. Yu, Y. Chen, and H. Zaidi, "Ava: A financial service chatbot based on deep bidirectional transformers," *Frontiers in Applied Mathematics and Statistics*, vol. 7, p. 604842, 2021.

[22] J. Kapočiūtė-Dzikienė, "A domain-specific generative chatbot trained from little data," *Applied Sciences*, vol. 10, no. 7, p. 2221, 2020.

[23] WoebotHealth. [Online]. Available: https://woebothealth.com/

[24] Chatfuel, "Hellofresh creates customer service chatbot with chatfuel: Chatfuel blog." [Online]. Available: https://chatfuel.com/blog/hellofresh-reduces-support-wait-times-with-chatfuel-messenger-bot

[25] Dominos.com, "Domino's home page - domino's pizza, order pizza online for delivery." [Online]. Available: https://www.dominos.com.my/

[26] M. Dhyani and R. Kumar, "An intelligent chatbot using deep learning with bidirectional rnn and attention model," *Materials today: proceedings*, vol. 34, pp. 817–824, 2021.

[27] HSBCHongKong, "Ask amy." [Online]. Available: https://www.askus.business.hsbc.com.hk/english/bot.htm?isJSEnabled=1&businessArea=Root&channel=Root

[28] AndyEnglishBot, "Chat and learn english with robot. chat and practice english with andy." [Online]. Available: https://andychatbot.com/

[29] GeeksforGeeks, "Explanation of bert model - nlp. geeksforgeeks." [Online]. Available: https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/

**Tong-Jun Ng** is a Software Engineering student at Faculty of Computing and Informatics, Multimedia University. He completed his bachelor's degree of Computer Science and Foundation in Information Technology in year 2023. He is interested in exploring and keeping himself up to date with the state-of-the-art technologies.

**Kok-Why Ng** is an Associate Professor at Faculty of Computing and Informatics, Multimedia University. His research interests include Chatbot, Recommender System, Computer Graphics, Computer Vision, Image Processing etc. He leads several funded projects on Modelling and Rendering, Recommender Systems and Microscopic Image Processing. He can be contacted at email: kwng@mmu.edu.my.

**Su-Cheng Haw** is a Professor at Faculty of Computing and Informatics, Multimedia University, where she leads several funded projects on the XML databases. Her research interests include XML databases, query optimization, data modeling, semantic web, and recommender system. She is also the chairperson for Center for Web Engineering (CWE) and chief editor for Journal of Informatics and Web Engineering (JIWE). She can be contacted at email: sucheng@mmu.edu.my.