COBY: COVID-19 Telegram Chatbot by Employing Machine Learning Algorithms

Muhammad Naufaldi
Computer Science Department,
Faculty of Computing and Media,
Bina Nusantara University,
Jakarta, Indonesia 11480
muhammad.naufaldi@binus.ac.id

Sunny Jovita
Computer Science Department,
Faculty of Computing and Media,
Bina Nusantara University,
Jakarta, Indonesia 11480
sunny.jovita@binus.ac.id

Christensen M. Frans
Computer Science Department,
Faculty of Computing and Media,
Bina Nusantara University,
Jakarta, Indonesia 11480
christensen.frans@binus.ac.id

Muhammad L. I. Hanafi
Computer Science Department,
Faculty of Computing and Media,
Bina Nusantara University,
Jakarta, Indonesia 11480
muhammad.hanafi003@binus.ac.id

Nunung N. Qomariyah
Computer Science Department,
Faculty of Computing and Media,
Bina Nusantara University,
Jakarta, Indonesia 11480
nunung.qomariyah@binus.ac.id

***Abstract*—COVID-19 pandemic has been one of the biggest concerns nowadays. People always curious and ask for immediate responses regarding the current situation. The chatbot can be very useful in this kind of situation which allows the system to understand text, which means it can respond appropriately. In order to be able to return the correct responses, the chatbot needs to learn how to classify the text data input from the users. In this paper, we study three different machine learning algorithms to work on text classification problems, namely Naive Bayes, Neural Network, and Support Vector Machine (SVM). An experiment was carried out to study which machine learning algorithms produce the most accurate responses when they are implemented in the Artificial Intelligence (AI) chatbot systems. In order to make sure the tests are consistent and fair, we conducted the experiment on the same dataset, and assessed the accuracy of their respective responses. In addition, we have also successfully implemented each of these algorithms as chatbots on a social media platform, Telegram.**

***Index Terms*—AI, Chatbot, Natural Language Processing, Text Classification, Support Vector Machine, Neural Network, Naive Bayes, COVID-19.**

I. Introduction

During the pandemic, people needed a way to get information regarding the disease without giving too much effort. Ideally, people can just write questions on their device and there will be a correct answer to those question. Unfortunately, false information are everywhere. It is hard for people who are not aware of this to decide what is right or wrong on the internet. To solve this problem, a chatbot is created.

Frequently asked questions regarding the pandemic are stored in a text file which will be used to train the chatbot so it can classify the user's question and answer it accordingly [1]. All of the classified questions from the dataset will be paired with the appropriate answer from a trusted source with the help of text classification.

978-1-6654-1697-9/21/31.00

Text classification algorithms are used to assign a predefined categories to text. Its domains of application are numerous and important, and given the proliferation of documents in digital form they are bound to increase dramatically in both number and importance [2]. This text classification algorithms are one of the task of Natural Language Processing (NLP). NLP is a subset of Artificial Intelligence (AI) that helps machine understand human language. There are multiple machine learning algorithms that are used for text classification. It is hard to tell which algorithm that can perform better than other algorithm for this chatbot.

In this paper, we would compare which machine learning algorithms can perform better. The algorithms that are being compared are Naive Bayes (NB), Support Vector Machine (SVM), and Neural Network (NN). We study how well can these algorithms classify questions from the dataset.

The motivation behind this study is out of the curiosity of which of the above-mentioned algorithms are more accurate, and thus more suitable when being implemented on intelligent chatbot systems. This is because, as we know, these three different algorithms are simply alternatives to which algorithm that can be used in chatbot systems. It is thereby also important to determine which of these algorithms generate the most accurate responses for their users.

II. Related Studies

Similar study has been performed by Tamizharasi et al. [3], where Tamizharasi [3] have created a medical chatbot that predicts user's sickness and recommend treatment based on the user's information and symptoms. Tamizharasi [3] used machine learning algorithms such as SVM, NB, and KNN to

train the medical chatbot and compared which of the three algorithms has the best accuracy. Hassan et al. [4] compared NB with SVM for text categorization task. Algorithms are enriched by Wikitology which is a knowledge repository. Nicholas et al. [5] created a medical chatbot that helps user to do self-assessment of their own health through Telegram or SMS using fuzzy SVM. Papiya et al. [6] created a healthcare chatbot that identify illness and supply basic information regarding the illness before consulting to a doctor. The chatbot will take user symptoms as its inputs and clarify it with a series of questions. Papiya [6] uses multiple algorithms which is N-gram, TF-IDF, and Cosine similarity. This research will try to provide a comparison between algorithms similar to a research by Tamizharasi [3] with different algorithms, topic and a much simpler dataset. As concluded in the research conducted by Tamizharasi [3] and Hassan [4] we would like to see if SVM would also perform better than NB and NN in a short text classification task.

III. Methods

Here we propose the development of COBY chatbot. The solution design is illustrated in Figure 1. The program will start by taking user input and process those input so it can be predicted by each models that was created using the algorithms that we want to compare. The outputs of those models would later be an intent or class that was defined in the dataset. Using the intent, the program would find the appropriate function to get the output that was defined in the code.

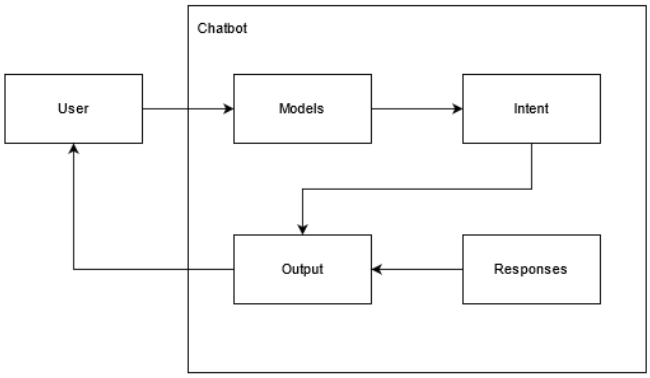

Fig. 1: Solution Design

A. Type of Chatbot

This chatbot is a retrieval based chatbot. A retrieval based chatbot is when there are predefined output for every class of input that it received. The reason why we went with retrieval based chatbot is because it is easier and we want to have a predefined response since any misinformation about COVID-19 would not be good.

B. Machine Learning Algorithms

In this current technology, chatbots could understand and respond as human brain do using human natural languange. Automated chatbots generally use Machine Learning algorithms (ML), coupled with Natural Language Proccessing (NLP) within the domain of Artificial Intelligence (AI). These technologies brought chatbot invention to a completely new advanced level. The algorithms that we chose for this chatbot are Naive bayes, Support vector machine, and Neural Network. These algorithms are a popular choice in text classification topic and that is why we chose to do these algorithms. In our case, the input to the machine learning algorithms is the text data (user questions) and the output is the intents (text class). This problem will be considered as multi class classification machine learning problem.

Naive Bayes: Naive Bayes (NB) is a machine learning algorithm that based on the Bayes' theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle; every pair of features being classified is independent of each other. Bayes' Theorem is a way of finding a probability when we know certain other probabilities.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

where:

$$\begin{aligned} P(A) &= \text{prior probability of event A} \\ P(A|B) &= \text{conditional probability of A, given B} \\ P(B|A) &= \text{conditional probability of B, given A} \\ P(B) &= \text{prior probability of event B} \end{aligned}$$

To find the probability of an event A happening when an event B occured, we use Bayes' Theorem. The equation assumes every feature is independent hence why it is called naive. Naive bayes is a popular algorithm in text classification due to how simple it is to implement allowing the program to have more efficiency. In some cases, Naive bayes are not as effective as the other machine learning algorithms because Naive bayes assumes every word in its input are independent. However, we still expect Naive Bayes to do well because the question in the dataset generally have a straightforward relationship with its intents [7].

Support Vector Machine: Support Vector Machine (SVM) is a supervised machine learning algorithm. SVM are commonly used in classification problems. SVM is based on hyperplane that divides the dataset into classes.

Hyperplane is a line that divides the data between classes. When it could not be divided with a simple line, the graph would be turned into 3D and the the line that divides the data would be a plane instead since it is 3D as shown in Figure 2.

We expect SVM to do better than NB. Generally in traditional text classification problems, SVM would do better than NB as mentioned in [8]. In addition, SVM are less likely to overfit [7]. SVM also does not require a good parameter tuning since they can find good parameter settings automatically which makes it easy to use [9].

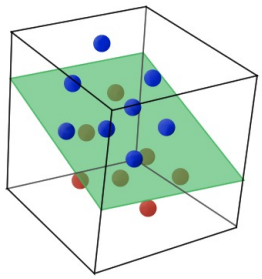

Fig. 2: Support Vector Machine Hyperplane

Neural Network: Neural network is a computational model based on how human brain process information. The base of this computational model is the neuron where it will receive inputs from other neuron or external sources. Each input have a weight which is given based on the relative importance of other inputs.

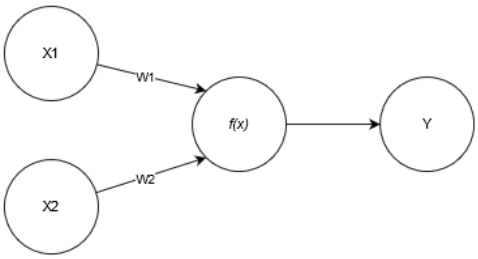

Fig. 3: Neuron

As it can be seen from Figure 3, the neuron takes inputs from $X1$ and $X2$ along with its weight ($w1$ and $w2$) and then calculate it with the defined function ($f(x)$) and it will output Y as in this figure.

The ability of Neural Network to learn complex might negatively affect the accuracy [7] since the questions from the dataset are rather straightforward and simple. However, that does not mean neural network algorithm will do horribly compared to the other algorithms. It might be compensated by exactly that ability of neural network to learn complex pattern [10] for a longer set of questions in the dataset.

C. Libraries

These are the libraries that are used in making the chatbot:

- **NLTK**
- **Scikit-learn**
- **PyTorch**
- **Selenium**
- **Beautifulsoup4**
- **Python-telegram-bot**
- **Requests**
- **Pycountry**

The libraries that are relevant to the machine learning aspect of this application are NLTK which is used to process sentence into tokens, Scikit-learn which is a powerful machine learning library that provides an automated way to create a model and split the dataset, and PyTorch which is also a machine learning library that specialize in natural language processing. We used PyTorch to implement the neural network model.

D. Dataset

The topic of our chatbot is COVID-19. By keeping this in mind, we searched for a dataset that has a lot of questions regarding COVID-19. We decided to use a dataset that was posted on IBM website named: COVID-19 Questions[1]. The sample questions and its intent is shown in Table II.

TABLE I: Dataset intents

Intents	Count
About_Anezka_WhoAreYou	8
COVID_Description	29
general_Help	6
Treatment_info	23
Quarantine_what_to_do	21
Protecting_Against_Infection	42
About_Anezka_hate	10
Case_Count	32
Symptoms	76
About_Anezka_Greeting	7
Total	254

TABLE II: Sample questions and its intents from dataset

Intents	Questions
About_Anezka_WhoAreYou	what can you help me with
About_Anezka_WhoAreYou	are you a robot?
COVID_Description	Is covid a disease?
COVID_Description	Where can I find more information about covid 19?
general_Help	Can you help me?
general_Help	what should I do
Treatment_info	Can I take ibuprofen if I have corona virus?
Treatment_info	what medicines should I take if I have coronavirus
Quarantine_what_to_do	If I have Covid how long do I have to self isolate
Quarantine_what_to_do	how long will we have to self quarantine?
Protecting_Against_Infection	Is it even possible to avoid the virus?
Protecting_Against_Infection	How often should I wash my hands?
About_Anezka_hate	everything is horrible
About_Anezka_hate	I hate these kinds of things
Case_Count	How many people are infected in Illinois?
Case_Count	are people infected in my city ?
Symptoms	what are the covid symtoms
Symptoms	I have headache and sore throat ...what should I do ?
About_Anezka_Greeting	good afternoon
About_Anezka_Greeting	What's up

In order to have a proper comparison, we decided to have a file dedicated to process the data that all the models will use. The pre-processing techniques that we used to process the dataset are mentioned below:

- **Tokenization**: The first step of NLP pre-processing technique that needed to be used is tokenization. In this tokenization, each of word in a form of sentence is separated word by word and it is stored in a separate list for words.

[1]The dataset is publicly available at https://developer.ibm.com/technologies/artificial-intelligence/data/cqa/.

By splitting them into words, we can easily process each words for the next step. This tokenization technique is applied to both intents and questions. The tokens of the intents and questions will be stored in separated list. Naveen [1] stated that this pre-processing technique will help in increasing the accuracy of providing the correct intents when the tokenized word of question is compared with the tokenized word of intent.

```
(['I', 'think', 'I', 'might', 'have', 'the', 'virus', '.'], 'Symptoms')
(['I', 'have', 'a', 'burning', 'feeling', 'in', 'my', 'chest'], 'Symptoms')
(['what', 'is', 'the', 'Corona', 'Virus'], 'COVID_Description')
(['What', 'does', 'covid', 'stand', 'for', '?'], 'COVID_Description')
(['what', 'is', 'covid19'], 'COVID_Description')
(['How', 'is', 'COVID-19', 'spread', '?'], 'COVID_Description')
(['I', "'m", 'afraid', 'and', 'do', "n't", 'know', 'what', 'to', 'do'], 'general_Help')
(['Should', 'I', 'avoid', 'people', '?'], 'Protecting_Against_Infection')
(['Where', 'did', 'the', 'coronavirus', 'come', 'from', '?'], 'COVID_Description')
(['Does', 'eating', 'garlic', 'help', '?'], 'Protecting_Against_Infection')
```

Fig. 4: Tokenization into words

- **Stop Words**: Stop words are words or characters that do not have much meaning in the language, including punctuations. For example, “is”, “a”, “the”, “!”, “.”. Therefore, to make it easier for the computer to understand the human language, it is important to filter out these characters/words, as they do not contribute much to the meanings. This process can only be done after Tokenization, where all the words/punctuation within the user input phrase/sentence had already been split individually first [11].
- **Stemming**: After filtering out the stop words, we would stem each words to prevent any repeated word that has similar meaning. Stemming is a rule-based process to reduce inflected words to its root form. This stemming technique is used to minimize the count of words that are repeated in the given sentence. It reduces the count of words by finding the roots of the word. For instance, the words ”organization”, ”organize”, ”organizer”. All of them have the root word of ”organ”. Thus, they are likely to have the same meaning and those three words will be considered as one word as ”organ”. In this chatbot program, stemming helps in decreasing the number of words stored in the list and increases the processing speed.
- **Bag of words**: Finally, the words that are stemmed would be turned into vectors with the bag of words technique. The bag of words technique takes all the vocabulary that is in the dataset and turn it into a vector map, and when an input is processed it will take those words and count the occurrences from the vector map that was created.

TABLE III: Bag of words

Vocab.	how	long	self	quarantine	if	i	covid	isolation
Sent. 1	1	1	1	1	0	0	0	0
Sent. 2	0	1	1	0	1	1	1	1

For example, two sentences will be pre-processed and added into the vocabulary list. Then for each sentences, the bag of words technique will be applied. The two sentences are:

- Sentence 1: `How long will we have to self quarantine?`
- Sentence 2: `If I have Covid how long do I have to self isolate`

Each sentence will produce a vector map that contains the occurrences of each words compared to the vocabulary. With each sentence added, the size of the vector map will grow and so is the vocabulary list.

E. How the Program Works

To see and use our code please download all the resources on Github link provided in the supplementary files. Install all the libraries required and then install Telegram. The user manual is provided in the Github. The application can be added to the Telegram by search the `coby_chatbot` application name (see Figure 8).

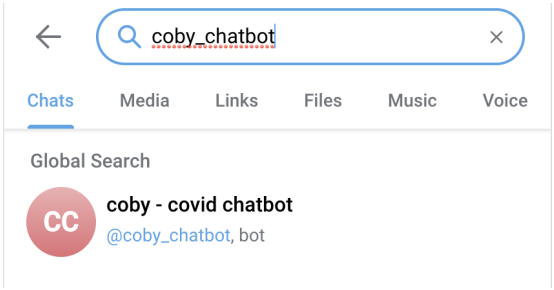

Fig. 5: COBY

The program starts by taking user input. The input will go through pre-processing techniques that was mentioned earlier. The input that has been processed will be used to predict the intent of the input. After getting the intent from the prediction, the program will search through a function that was created to handle the intent. From that function, the proper responses will be returned and sent through the bot to the user (see Figure 6).

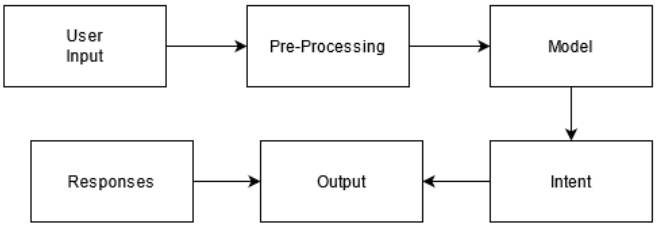

Fig. 6: Program flow

The set of responses that we have are taken from various sources. The program have three types of responses based on where it came from (see Figure 7).

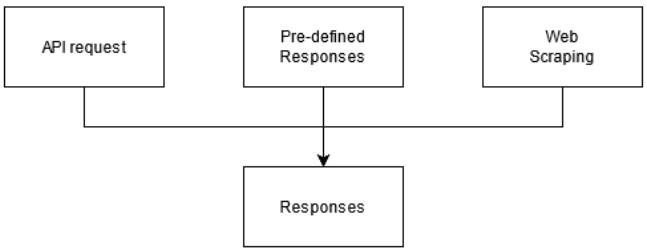

Fig. 7: Source of responses

The first type of response come from API request. This type of response handles the intent called *Case_Count* which gives the number of cases throughout the world or number of cases in a specific country. The data of cases are taken from a COVID-19 API[2].

The second type of response is the pre-defined responses. This type of response handles intents that does not require any trusted data or simply interactions. These intents are:

- About_Anezka_WhoAreYou
- About_Anezka_Greeting
- About_Anezka_hate
- general_help
- Gratitude

The third type of response come from web scraping. These responses are scraped from the CDC website[3]. These responses explains about what COVID-19 is and how to handle it as a civilian. Because of that, we decided it is better to scrape a trusted website in case there are any changes on this information. The intents that uses web scraping are:

- Treatment_info
- Quarantine_what_to_do
- COVID_Description
- Protecting_Against_Infection
- Symptoms

IV. Results and Discussion

The results of the comparison are evaluated using the standard metrics of accuracy, precision, recall, and f1-score through Scikit-learn classification report. The data are split into 66% for training dataset and 33% for testing dataset with the same seed for all models.

Performance measures generally evaluate specific aspects of classification task performance, and thus do not always present identical information [12]. Therefore, we will discuss the comparison between algorithms by each measure. The performance measures that will be discussed are:

- Precision
  The proportion of correctly predicted positives to all positives.

[2]https://covid19.mathdro.id/api

[3]https://www.cdc.gov/coronavirus/2019-ncov/index.html

TABLE IV: Naive Bayes

	Prec.	Rec.	F1	Sup.
About_Anezka_WhoAreYou	0.00	0.00	0.00	2
About_Anezka_Greeting	0.00	0.00	0.00	2
About_Anezka_hate	0.00	0.00	0.00	2
Treatment_info	0.60	1.00	0.75	3
Quarantine_what_to_do	1.00	0.70	0.82	10
COVID_Description	0.75	0.27	0.40	11
Case_Count	1.00	1.00	1.00	10
Gratitude	1.00	1.00	1.00	5
Protecting_Against_Infection	0.80	0.75	0.77	16
Symptoms	0.63	1.00	0.77	27
general_Help	0.00	0.00	0.00	1
accuracy			**0.75**	89
macro avg	0.53	0.52	0.50	89
weighted avg	0.73	0.75	0.71	89

TABLE V: Neural Network

	Prec.	Rec.	F1	Sup.
About_Anezka_WhoAreYou	0.00	0.00	0.00	2
About_Anezka_Greeting	1.00	0.50	0.67	2
About_Anezka_hate	0.00	0.00	0.00	2
Treatment_info	1.00	1.00	1.00	3
Quarantine_what_to_do	1.00	0.80	0.89	10
COVID_Description	0.91	0.91	0.91	11
Case_Count	1.00	1.00	1.00	10
Gratitude	1.00	1.00	1.00	5
Protecting_Against_Infection	1.00	0.81	0.90	16
Symptoms	0.92	0.89	0.91	27
general_Help	0.25	1.00	0.40	1
accuracy			**0.84**	89
macro avg	0.73	0.72	0.70	89
weighted avg	0.91	0.84	0.87	89

TABLE VI: Support Vector Machine (SVM)

	Prec.	Rec.	F1.	Sup.
About_Anezka_WhoAreYou	0.00	0.00	0.00	2
About_Anezka_Greeting	1.00	0.50	0.67	2
About_Anezka_hate	0.00	0.00	0.00	2
Treatment_info	1.00	1.00	1.00	3
Quarantine_what_to_do	1.00	1.00	1.00	10
COVID_Description	0.82	0.82	0.82	11
Case_Count	1.00	1.00	1.00	10
Gratitude	1.00	1.00	1.00	5
Protecting_Against_Infection	1.00	0.75	0.86	16
Symptoms	0.74	0.96	0.84	27
general_Help	0.50	1.00	0.67	1
accuracy			**0.84**	89
macro avg	0.73	0.73	0.71	89
weighted avg	0.85	0.87	0.85	89

- Recall
  The ratio of correct positive predictions to the total positives examples.
- Accuracy
  The ratio of correctly predicted examples by the total examples.
- F1-score
  A way of combining the precision and recall of the model.

The accuracy of the 3 algorithms are shown in (Table IV,

Table VI, and Table V). SVM and NN ties with the accuracy of 84% while NB have a lower accuracy which is 75%. But, since the dataset is skewed, we will not be paying too much attention to the accuracy since accuracy is not a good metric for skewed datasets [13]. The precision and recall can be used in this study. Although, precision and recall can be optimised at the expense of the others despite being a valid metric on their own [14]. Therefore, we would like to use F1-Score instead. F1-Score is a better performance metric in this case since the dataset is skewed and we want to seek the balance between precision and recall. In this case NN model did slightly better in F1-Score than SVM. NN model have 87% while SVM have 85%. It is also worth mentioning that all of these algorithm are the basic models without customization and can be further improved than it is now.

V. Program Testing

In order for user to be able to interact with the COBY chatbot, user needs to add the chatbot in Telegram application with username coby_chatbot. After that user needs to open the chat window in order to chat with the chatbot. Figure 8 shows where users can ask questions to the chatbot.

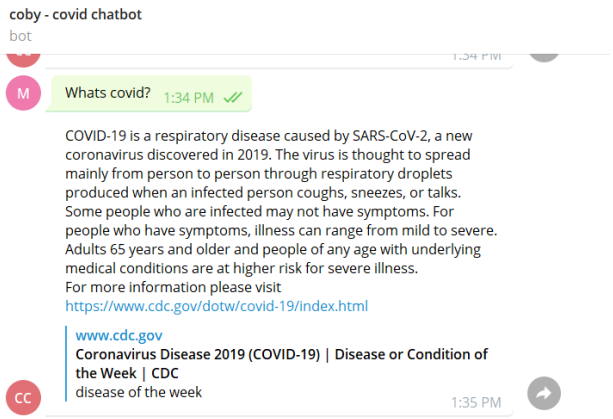

Fig. 8: Conversation between COBY and user

After giving it question, the chatbot will answer user's question immediately. A `/help` command can be used in order to list the capabilities of the chatbot. If the user wants to change the chatbot algorithms, just simply type its specific command for instance, `/SVM`, `/NB`, and `/NN` command. The chatbot algorithm will change automatically based on the user's choice. Other examples of the conversation can be seen in Figure 9 and Figure 10.

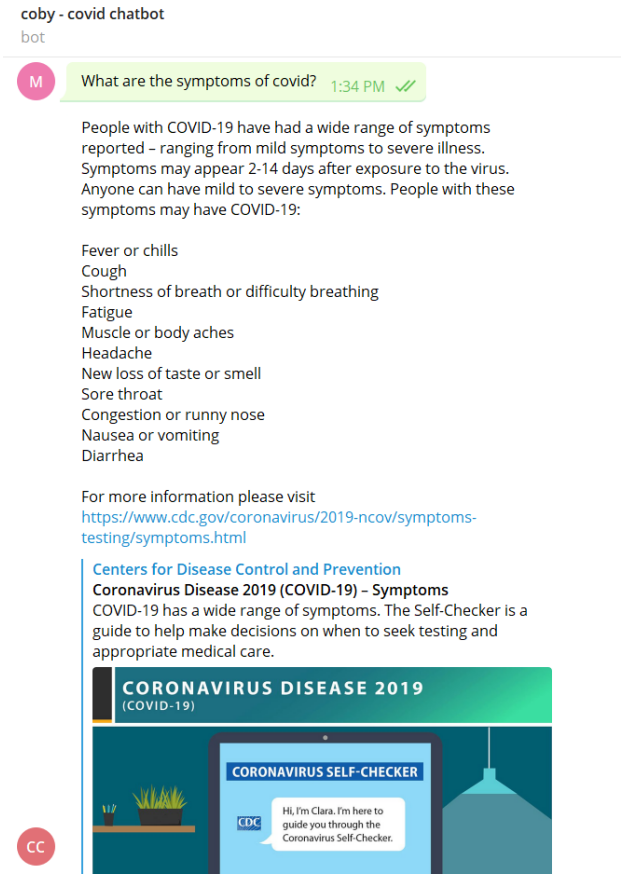

Fig. 9: Conversation between COBY and user

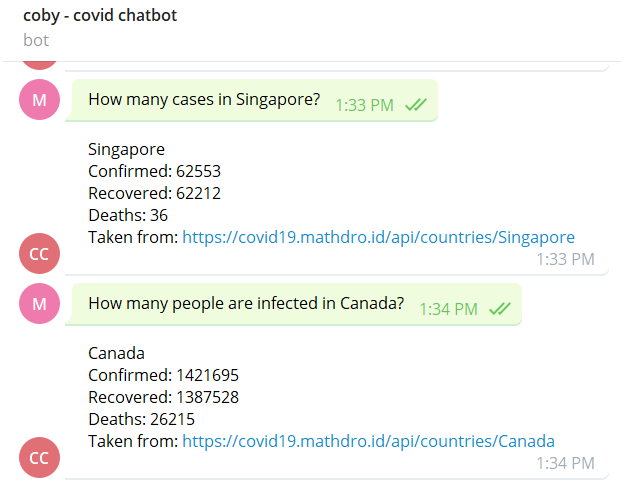

Fig. 10: Conversation between COBY and user

VI. Conclusion

Text classification is important in many applications. The problem that occurred with this study is the lack of dataset. Moreover, the dataset distribution are very imbalanced. With that in mind, by using the dataset that we have and make models with naive bayes, neural network, and SVM, we were able to compare the performance of the models. We created a retrieval-based chatbot using the models that we created during this comparison that can provide users with answers to some of the most asked questions regarding COVID-19. The output of this chatbot are mostly scraped from the CDC website and can be trusted. The results of the comparison that we have is overall SVM have a better overall accuracy while neural network have a better F1-Score all around. However, since none of the algorithms give 100% accuracy, it is possible

for the chatbot to sometimes give wrong output, which can lead the users obtain inaccurate information. Thus, maintaining such accuracy and user satisfaction is a limitation of this study.

Supplementary Files

The code is available at https://github.com/sunnyjovita/coby-covid-chatbot. The video of the application can be found on https://youtu.be/aiUFY9W1SgA. The application on Telegram is available to be accessed via https://t.me/coby_chatbot.

References

[1] N. S, "Automated chatbot implemented using natural language processing," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, p. 3, August 2020. [Online]. Available: https://irjmets.com/rootaccess/forms/uploads/automated-chatbot-implemented-using-natural-language-processing.pdf

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, p. 1–47, Mar. 2002. [Online]. Available: https://doi.org/10.1145/505282.505283

[3] B. Tamizharasi, L. J. Livingston, and S. Rajkumar, "Building a medical chatbot using support vector machine learning algorithm," *Journal of Physics: Conference Series*, vol. 1716, p. 012059, dec 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1716/1/012059

[4] S. Hassan, M. Rafi, and M. S. Shaikh, "Comparing SVM and naive bayes classifiers for text categorization with wikitology as knowledge enrichment," *CoRR*, vol. abs/1202.4063, 2012. [Online]. Available: http://arxiv.org/abs/1202.4063

[5] N. A. I. Omoregbe, I. O. Ndaman, S. Misra, O. O. Abayomi-Alli, and R. Damaševičius, "Text messaging-based medical diagnosis using natural language processing and fuzzy logic," *Journal of Healthcare Engineering*, Sep 2020. [Online]. Available: https://doi.org/10.1155/2020/8839524

[6] P. Mahajan, R. Wankhade, A. Jawade, P. Dange, and A. Bhoge, "Healthcare chatbot using natural language processing," 2020. [Online]. Available: https://www.irjet.net/archives/V7/i11/IRJET-V7I11283.pdf

[7] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167811618300545

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *CoRR*, vol. cs.CL/0205070, 2002. [Online]. Available: https://arxiv.org/abs/cs/0205070

[9] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Machine Learning: ECML-98*, C. Nédellec and C. Rouveirol, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 137–142. [Online]. Available: https://doi.org/10.1007/BFb0026683

[10] J. V. Raj, J. V. Anton, and J. P. D. Raj, "Detection of recovery of covid-19 cases using machine learning," *Int J Cur Res Rev— Vol*, vol. 13, no. 06, p. 59, 2021. [Online]. Available: http://ijcrr.com/uploads/3565_pdf.pdf

[11] S. Ting, W. Ip, A. H. Tsang *et al.*, "Is naive bayes a good classifier for document classification," *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37–46, 2011.

[12] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2078-2489/10/4/150

[13] E. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques," *WSEAS transactions on computers*, vol. 4, pp. 966–974, 08 2005.

[14] D. Xhemali, C. J. Hinde, and R. G. Stone, "Naïve bayes vs. decision trees vs. neural networks in the classification of training web pages," *IJCSI*, vol. 4, 2009. [Online]. Available: http://cogprints.org/6708/1/4-1-16-23.pdf