NATURAL LANGUAGE PROCESSING PROJECT

TOPIC : FAQ CHATBOT USING NEURAL NETWORK.

SUBMITTED BY:

GROUP 7

PURNA BISHAS

S2463776

s.a.bishas@student.utwente.nl

SATHVIK GURU RAO

S2431114

s.gururao@student.utwente.nl

INTRODUCTION

In a world where time is money and personal are too expensive businesses have started to make more and more use of chatbots. The chatbots usually replace the temporary need for customer service by answering mundane and ordinary frequently asked questions (faq). This smart application is designed to imitate a human conversation to give the user an interactive experience when looking for the answer or information to their question. Some of the popular chatbots include Alexa from Amazon and Siri from Apple.
In our NLP project we will be creating a chatbot for a small scale computer repair shop business using a dataset. We will be creating a chatbot incorporating the Frequently Asked Questions (FAQ) as its output to the user asking for the shop questions. In the dataset there are questions present that have direct answers to some of the more common asked questions by customers who don't need to be followed up by human contact.
The user can interact with the chatbot on the website to have an automated conversation and get the answers for their questions.
This report is not a means to solve a problem but to innovate on existing methods.

RELATED RESEARCH AND WORK

Chatbots have become very common in all kinds of fields. Raghuwanshi and S.singh have created a FAQ chatbot [1] using the AIML script and query matching. This has a drawback if the bot does not find any matching query in the dataset. There are some rule based chatbots like ALICE [2] which require the data to be coded manually which takes a long time to build.
In this era, machine learning is being used in all kinds of applications. Machine learning has been used in chatbots as well. Retrieval based chatbot for ecommerce using RNN has been shown by P.Muangkammuen and K. R. Saikaew [3]. Yu Wu et al.,[4] who have an interesting approach for chatbot using the RNN model. They are using a chronological order to match the response.

What is needed is the chatbot's concept of usability: *"The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments."*
N. Radziwill and M. Benton [5] state that a conversation agent should mimic human interactions. Also when adapting new information and requests. The quality of the chatbots are needed for people to use them.
We are creating a chatbot using the artificial neural network using tensorflow and keras. The model takes the user input and classifies the input to a class which in our model is a tag, and responds to the user with a random response which has been given in the dataset for the application. The main part of this application is the classification of the input. We are using a small dataset for modelling which should be a problem for the neural network according to Salvatore and Isabella [6] who have justified that the neural network for small dataset can perform well.

DATASET

We will be using a computer repair shop FAQ dataset which we found on Kaggle (link: https://www.kaggle.com/bitsofishan/it-helpdesk-chatbot-dataset) for the project. The dataset is a .json file which contains intents.
The intents contain:

1) tags: tells us the class or label to which the conversation belongs (like greetings). The tags can be considered as the topic of the conversation.
2) patterns: contains a list of all the user questions or the user inputs according to which the chatbot has to respond.
3) responses: a list contains some responses which will be used by the bot to reply for the user.

Image 1: the contents of the dataset

PACKAGES

The project is built with python using python packages like numpy, nltk and tensorflow and keras library. NLTK is the natural language kit for python which we will use for the tokenization and stemming of words. This we will be using to preprocess the data of the dataset. Tensorflow is an end-to-end open source machine learning platform.
Tensorflow can be combined with the keras library. We will make use of these two to create the neural network model.

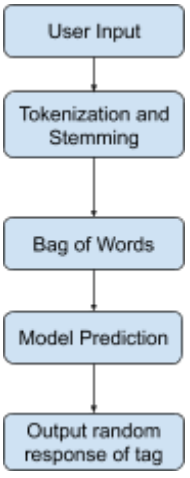

Image 2: Chatbot flowchart

DATA PREPARATION

Preparation of the dataset in consideration of the values we will be extracting from is not needed. The dataset is formatted, processed and reable for the program.
We have to extract the main features and output of the data to use it for training the model. The patterns from the data are taken as the feature input and the tags should be the output of the model. The patterns and tags are taken in a separate list. The patterns should be given as a vector to model as the neural networks only take numerical data. For this, we have to first create a vocabulary and then convert it into vector form.

VOCABULARY

The vocabulary is a list of words extracted from the data which are the main features. This list will contain only unique words, duplicates are excluded. To create a vocabulary list, first we have to tokenize the patterns of the data from the dataset and then take the stem of each word. The tokenization of words is done using the nltk word tokenizer method. We then use the Lancaster stemmer method from nltk.stem package to get the root form of the word. Stemming will reduce the amount of data by giving us the base form of the word. The list is converted to a set type which will help in removing duplicate words.
The set is converted to list format. This vocabulary is used to prepare the bag words for the model input.

BAG OF WORDS

Machine Learning algorithms cannot use text data for training, it takes only numerical data like a vector to train the model. The length of the *bag of words* (further mentioned as BOW) is the same as that of the vocabulary. The BOW will represent the patterns in a vector form. It will compare the sentence with the vocabulary list and give a “1” if the word in the sentence is present in the list else it will be given a “0”. So the vector contains zeros and ones in the place where the word does not exist and exist respectively.

Image 3: vector output of the BOW

From the above figure, we can notice a vector of zeros and ones. The ones represent the place where the word does exist in the vocabulary list. This will be used as the input to the model.
Similar to this BOW for patterns, we will create another one for the output. This is the length of the number of tags. The vocabulary for this is the list of tags. The number ‘1’ is given in the place of the tag to which the sentence belongs.

```
train_y[0]
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
```

Image 4: Training output used for the model of a random tag.

TRAINING AND TESTING DATA

Training: The training data contains both the input and the corresponding output. The data is in vector form. The bag of words for input data will be similar for the same tags.
Testing: The dataset used for training is small, so splitting a small dataset would give too little data for training which does not give good results. The test dataset has been created manually in a json file with some patterns of questions and their respective tags. This will be used for the model evaluations and finding the accuracy.

```
print("training data input:",train_x[0])
print("training data output",train_y[0])
```

```
training data input: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
training data output [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
```

Image 5: Input training data with output training data

METHODOLOGY

NEURAL NETWORK

The main task of this project is to classify the user input to a tag/class. Neural Network is a popular method used for multi-class classification. The neural network consists of an input layer, hidden layer and an output layer. Tensorflow and keras have a sequential model for modeling a neural network. The general working of the neural network is by taking a vector input.
Feedforward the output to the next layer. The weights and the bias will be added to the vector with fine tuning by the model. The outputs of the layer will be given as input to the subsequent layers. Finally the output layer will give the results.
The chatbot works mainly on classification of the user input to different tags/classes. After predicting the tag/class of the user message, a random response will be chosen from that tag and used to reply to the user. Sequential Neural network from keras has been used to create the model. The model is built by adding layers sequentially one after the other with the input layer as the first layer followed by some hidden layers and the last layer is the output layer.

INPUT LAYER

The input layer takes the input for the model in the form of vectors.The training data created will be used as input to the layer. The BOW in vector form is feeded into the network. The number of nodes in this layer is equal to the number of input variables, which is the length of the BOW list. Here only the BOW of patterns is given.

HIDDEN LAYER

The hidden layer takes the output from the input layer. There is no specific number of hidden layers that needs to be used. The model for this project works well with a single hidden layer.
An arbitrary number of nodes are used in the hidden layer. The Dense class from keras tells that the layer is fully connected. All the nodes from the input layer are connected to all the nodes in the hidden layer. The output dimension of the hidden layer is taken by default as the input dimension which is the number of nodes.

OUTPUT LAYER

The output layer will give the end result of classification in vector form. The length of the vector is given as the number of tags in the data. There will be a node for each tag in the output. The probabilities sum up to 1. An activation function is used to convert the given input to a non linear transformation. The output is a vector of probabilities of a tag to which the pattern belongs. This kind of result is obtained by using softmax activation function.
softmax function

MODEL COMPILE

After building the model, the final step is to compile the model. In the compile stage we use a loss function, which in our model is the mean squared error function. The loss function tells how good the prediction is with respect to the outputs.
In a neural network model , the weights and bias are changed to reduce the loss function. This minimizing of loss is done using an optimizer in the compile. “Adam” optimizer is used in our model. The optimizer changes the output according to the loss function.

MODEL FIT

The model is fully completed when we fit the model with the training dataset. In keras, the fitting of the model is done using the model.fit() method which takes the training input and training output. The other parameters which are used in this method are, epochs which is the number of times the data is used to train the model and the batch size which tells the number of data to use at one single iteration. Another parameter that has been used is the call back which is used to plot the accuracy and the loss of the training data. In image 7 we

can see that the accuracy has increased with each epoch and simultaneously the loss has been reduced. The accuracy given in this cannot be used for the evaluation of the model as it is the same data that is being used again. The evaluation is done in the next section.

```
#sequential neural network model

model = tf.keras.Sequential([tf.keras.layers.Dense(16, input_shape = (None,len(train_x[0]))), # input layer
                             tf.keras.layers.Dense(10), #hidden layer

                             tf.keras.layers.Dense(len(train_y[0]),activation="softmax"),]) #output layer
model.compile(optimizer = "adam",loss = "mean_squared_error",metrics = ['accuracy'])
model.fit(train_x, train_y, epochs=100, batch_size=8,callbacks=[plot_losses])
model.save('model.keras')
```

Image 6: screenshot of the Model.

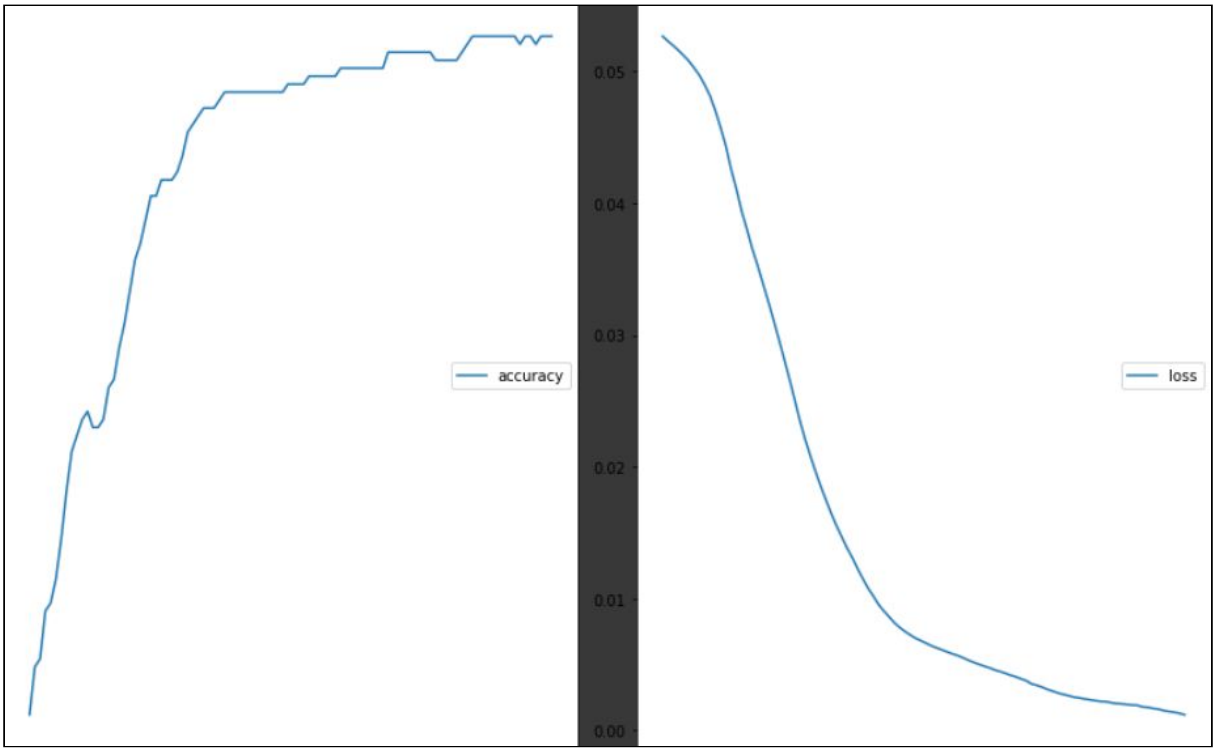

Image 7: Accuracy and loss plot

RESULTS

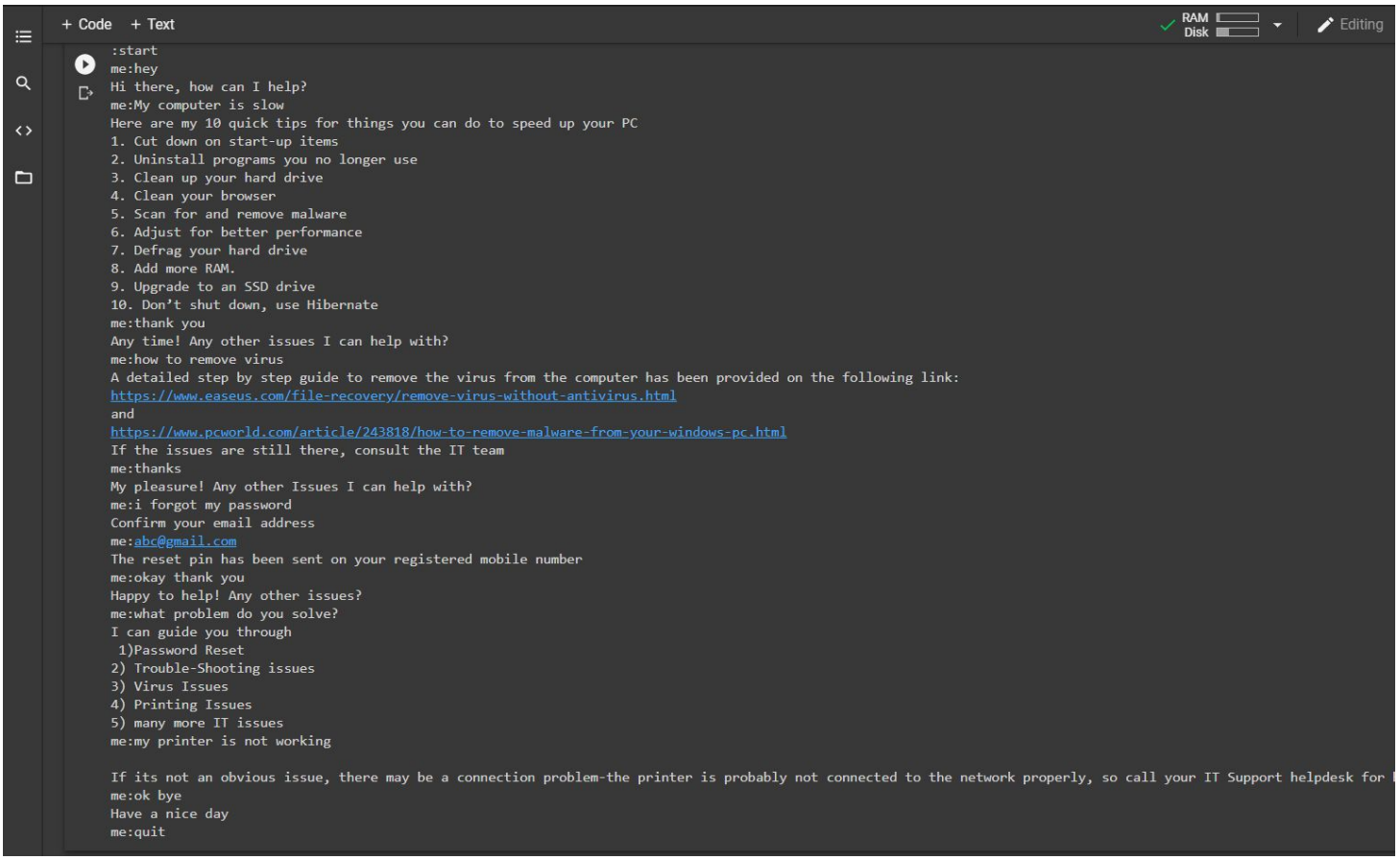

Image 8 : Sample of the chatbot conversation

EVALUATION

The model after training with the training dataset is used to evaluate with a test dataset. The test dataset is manually created by taking some users' questions and adding it to a json file. The data is converted to vector form using the same method as for the training data. The evaluation of the model is done using the model.evaluate() keras method. This method takes the test data and their tags. The loss and the accuracy has been shown in the figure below.

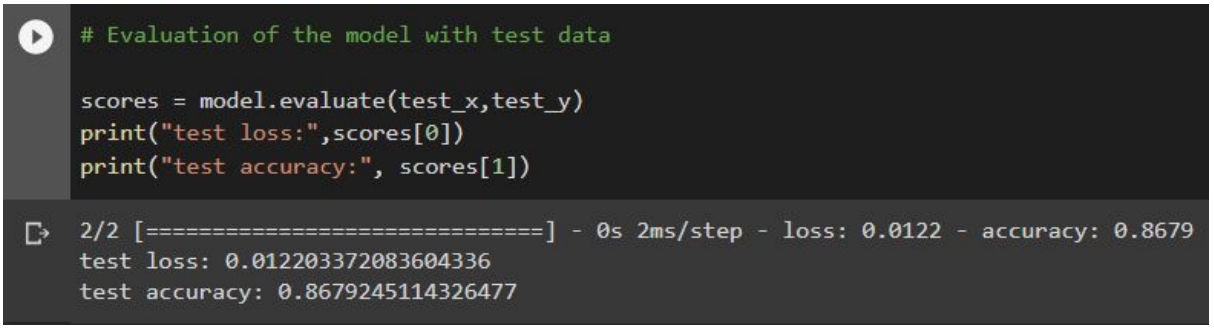

Image 9: accuracy and loss of the model when evaluated using test dataset

CLASSIFICATION

```
classify("forgot password")

[1.4710680e-04 1.1983623e-03 2.2408967e-03 5.9094907e-07 9.3428785e-01
 1.6450421e-04 1.5790934e-05 1.3018567e-04 1.5223528e-04 1.3648273e-04
 3.3498459e-02 4.4527891e-04 6.4592442e-04 8.2082804e-03 7.2296001e-03
 1.0200992e-02 9.0412839e-05 1.2070575e-03]
[('Password Reset', 0.93428785)]
```

Image 10: classification of the user input using model.predict()

The classify function takes the user input and classifies the sentence to its class using the models prediction. The model prediction gives an array of probabilities of all the classes. The class with maximum probability is considered as the best prediction. From the above figure we can see that the sentence “forgot password” has a 93% probability that it belongs to the “password reset ” class.

DISCUSSION

The chatbot works fine with frequent asked questions about certain topics. But there surely is a lot of scope to include more topics to the dataset. The chatbot only processes the terms known to the program. When asked a question unfamiliar to the chatbot it will not respond to the user by answering the question but giving options to choose from which are familiar to the chatbot. This leaves room for improvement on error logging. Right now the program does not respond clearly to the user that the chatbot does not understand the input. We should give that back as feedback to the user.

```
:start
me:how to fix internet
Happy to help
```

Image 11: example how the code has mistaken for unknown input.

CONCLUSION

The chatbot application created works very well for a small scale computer repair shop. This can be used for customer services. The chatbot will reply with high accuracy if the users input is regarding the topics which are mentioned in the dataset. If the dataset is added with more information then this application can be used broader than its current ability.

REFERENCES

[1] Ranoliya, Bhavika & Raghuwanshi, Nidhi & Singh, Sanjay. (2017). Chatbot for university related FAQs. 1525-1530. 10.1109/ICACCI.2017.8126057.
[2] Joseph Weizenbaum, "ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine", *Association for Computing Machinery (ACM)*, 1966.
[3] P. Muangkammuen, N. Intiruk and K. R. Saikaew, "Automated Thai-FAQ Chatbot using RNN-LSTM," 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2018, pp. 1-4, doi: 10.1109/ICSEC.2018.8712781.
[4] Wu, Yu & Wu, Wei & Xing, Chen & Zhou, Ming & Li, Zhoujun. (2017). Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. 496-505. 10.18653/v1/P17-1046.
[5] Evaluating Quality of Chatbots and Intelligent Conversational Agents Nicole Radziwill and Morgan Benton
[6] Ingrassia, Salvatore & Morlini, Isabella. (2005). Neural Network Modeling for Small Datasets. Technometrics. 47. 297-311. 10.1198/004017005000000058.