A Banking Chat Bot - Conversational Bot for Customer Care using Deep Neural Networks

Raja Rathina.G[1], Syed Shenaz.S[2], Vani.R[3], Dr.V.Gowri[4]

[1,2,3]Department of Information Technology, Velammal Engineering College, Chennai, India
[4]Professor, Department of Information Technology, Velammal Engineering College, Chennai, India

---***---

Abstract - *A chatbot is a computer program which involves artificial intelligence concepts and responds like an intelligent entity when conversed with. This paper discusses the deployment of one such conversational bot intended to serve a bank in helping the customers with the procedures that should be followed while acquiring any service from a bank. Practically, it is very difficult for the common people to understand the terms and conditions involved with banks. Therefore, a user needs some helping hand to complete the procedures involved in acquiring the services provided by banks. In order to provide a customer, the necessary guidelines and a great experience, a bank requires a Business process outsourcing (BPO) unit. We could rather use a virtual interactive system such as a chatbot built with artificial intelligence to provide an experience similar to that of conversing with an actual human and at a lower cost. Chatbots can be utilized to minimize up to 60% of the cost spent on BPO and provide the same services at the customer's pace and place.*

***Key Words*: ANN(Artificial Neural Network), DNN(Deep Neural Network), Chat-Bot, NLP(Natural Language Processing).**

1. INTRODUCTION

People today prefer to talk with chatbot rather than human agents. This trend, which started slowly, is beginning to transform the retail banking industry as a whole. The transformation is majorly influenced by the advancements in chatbot technology and 24/7 availability of the bots to answer customer's routine queries, especially after business hours. Thus, with the increased implementation of chatbots in the retail banking industry, chatbot development has become more trending [1].

In this paper, we will discuss about the artificial intelligence concepts that can be used to build a chatbot for a bank. We will be using concepts of NLP (Natural Language Processing) and DNN (Deep Neural Network). To build a chatbot. Thus, the chatbot can respond to the queries in an accurate manner.

Sometimes, the computer may fail to understand the meaning of a sentence, leading to uncertain results. These problems can be overcome by the machine learning algorithms like DNN and NLP.

The way we converse with the chatbots is nonlinear, irregular and full of context. In industries where customer service is of prime importance like banking, chatbots equipped with NLP can analyze, process and communicate with users using language they understand. NLP techniques categories customer data by tagging parts of speech, correcting spelling and re-formatting data into something the machine can read [8].

A deep neural network is an artificial neural network with multiple layers between the input and output layers. It resembles the complex neural structure of a human brain. The DNN finds the correct mathematical manipulation to turn the input into the output, it is done based on probabilities of each node to be true or false. The input layer receives input data and passes the inputs to the first hidden layer. The hidden layers perform mathematical computations on our inputs to set their respective probabilities. One of the challenges in creating deep neural networks is to calculate the number of hidden layers, as well as the number of neurons for each of the hidden layer.

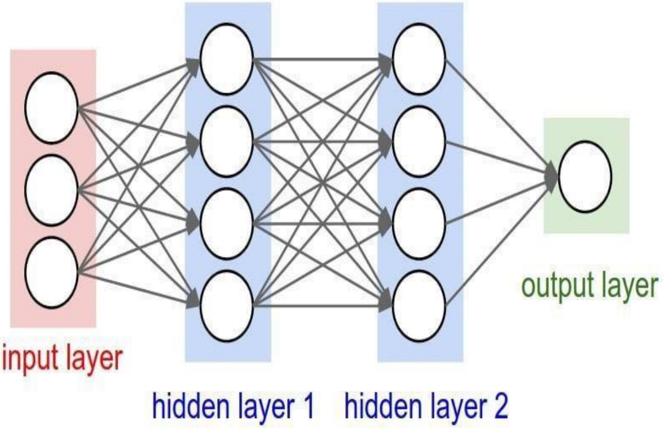


Fig-1: Basic Structure of Artificial Neural Network

2. PROPOSED MODEL

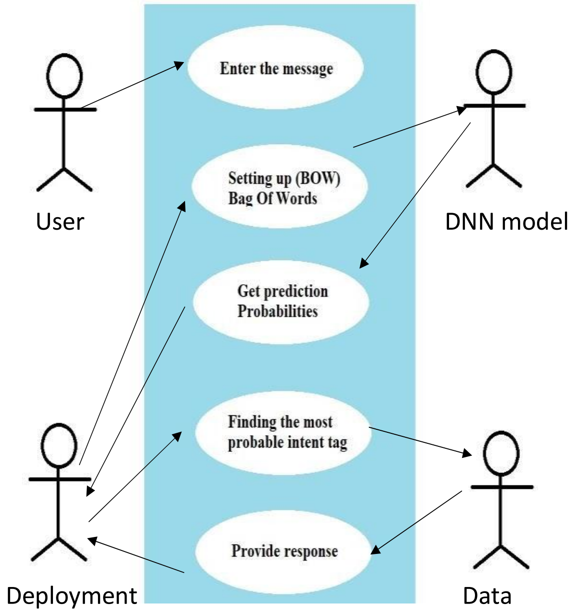

Enter the message
Setting up (BOW) Bag Of Words
Get prediction Probabilities
Finding the most probable intent tag
Provide response
User
DNN model
Deployment
Data

Fig-2: Use Case diagram of proposed solution

The above use-case diagram shows the actors in the proposed system and the actions performed by them. The user first types the query in the deployment server. The server on receiving the text input, converts it into a Bag of Words. Thus, the string is converted into digits which can be fed into the Deep Neural Network. The DNN model further processes the BOW and return a set of probabilities, which state the associativity between the input query and the response. Higher the probability better is the accuracy of response. So, the highest probability value is sent to the dataset and a response from that corresponding intent tag is returned.

2.1 DEEP NEURAL NETWORK MODEL

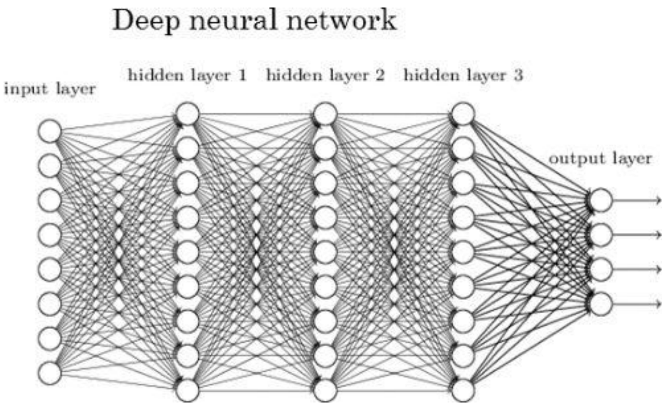


Fig-3: Proposed Deep Neural Network

The DNN Model proposed has three hidden layers to process the input. Each of these hidden layers are made of ten neurons. Softmax is a classifier and function that is being used. With Softmax, the data can be categorized by direct classifier. The number of epochs used are thousand five hundred. An epoch is one cycle through the full training dataset. Usually, training a deep neural network takes more than a few epochs [5].

2.2 DATA SET

We are using a JSON file, that contains key-value pairs of patterns and responses. What we are doing with the JSON file is creating a bunch of messages that the user is likely to type in and mapping them to a group of appropriate responses. The tag on each dictionary in the file indicates the group (the context) that each message belongs too [7]. With this data we will train a neural network to take a sentence of words and classify it as one of the tags in our file. Then a response is randomly chosen from that group and displayed to the user. The more tags, responses, and patterns are provided to the chatbot the better and more complex it will be.

2.3 DEPLOYMENT SERVER

We use Flask Web Framework to deploy the conversational bot in real time. Python requires its own web framework like Django or Flask to connect to a website/Frontend/User Interface. The Design of User Interface is being developed using HTML-5, CSS-3 and Java Script. Flask will act as a middle ware by fetching the user input and redirecting it to the python backend server [6]. Once the backend process is done, it will again redirect the output to the user interface. Thus, the developed project can be hosted and made available for public usage.

3. SYSTEM IMPLEMENTATION

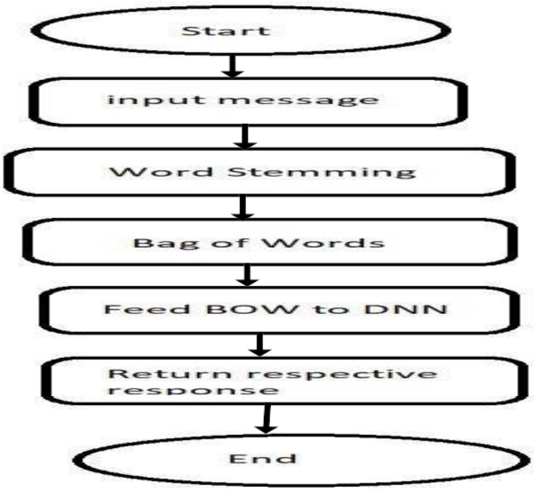
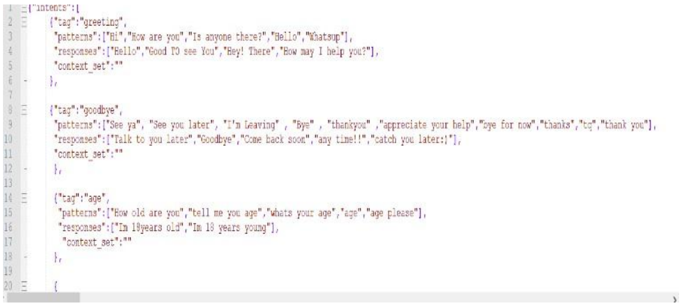


Fig-4 Flow chart depicting the workflow of the chatbot

3.1 TRAINING DATA

We are using a JSON file, that contains key-value pairs of patterns and responses. In the JSON file, we create a bunch of messages that the user is likely to type in and map them to a group of appropriate responses. The tag key on each dictionary in the file indicates the group that each message belongs too. With this data we will train a neural network to take a sentence of words and classify it as one of the sub-dictionaries with the tag in our file. Then a response is randomly chosen from that group and displayed to the user. The more tags, responses, and patterns are provided to the chatbot the better and more complex it will be [2].



```
{"intents":[
    {"tag":"greeting",
     "patterns":["Hi","How are you","Is anyone there?","Hello","Whatsup"],
     "responses":["Hello","Good TO see You","Hey! There","How may I help you?"],
     "context_set":""
    },

    {"tag":"goodbye",
     "patterns":["See ya", "See you later", "I'm Leaving" , "Bye" , "thankyou" ,"appreciate your help","bye for now","thanks","tq","thank you"],
     "responses":["Talk to you later","Goodbye","Come back soon","any time!!","catch you later:)"],
     "context_set":""
     },

     {"tag":"age",
      "patterns":["How old are you","tell me you age","whats your age","age","age please"],
      "responses":["Im 18years old","Im 18 years young"],
       "context_set":""
     },

     {
```

Fig-5: Dataset

3.2 EXTRACTING DATA

Now we must extract the data which we want from our JSON file. We need all the patterns and tags to which the user query belongs to. A list of all of the unique words in our patterns, are created to store these values. Then, by iterating through the JSON file, we can extract the data we want [7].

3.3 WORD STEMMING

Stemming is a process of finding the root of the word. In other words, to eliminate all the punctuations, plural forms, tenses and other such elements of a word [3]. The resulting word is the root of that word. For example, the word "that's" stem might be "that" and the word "happening" would have the stem of "happen"[3]. Process of stemming words is used to reduce the vocabulary of our model and attempt to find general meaning or context of the sentences. The stemmed words are stored as a unique list to use in the next step of our data pre-processing.

3.4 BAG OF WORDS

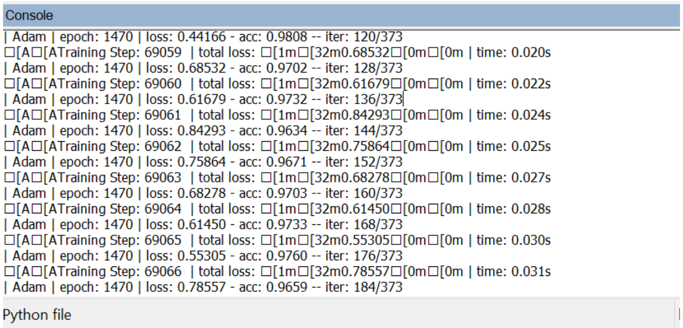
Neural networks and Machine learning algorithms require numerical input whereas, a chatbot basically receives text or string type input. Thus, to represent our sentences or string type input as numerical data, we use the concept of bag of words. We will represent each sentence with a list, the length of the amount of words in our model's vocabulary. Each position of the list will represent a word from our vocabulary. If the position in the list is a 1 then that will mean that the word exists in our sentence, if it is a 0 then the word is not present. We call this a bag of words because the order in which the words appear is not maintained. Instead only the presence or absence of a word is only determined [4].

3.5 DEVELOPING A MODEL

Now that we have pre-processed all of our data, we are ready to start creating and training a model. In this project, we have used a Deep neural network with three hidden layers and ten neurons in each layer. our network will be fed with bag of words. Thus, on further processing by our DNN, a class that the BOW belongs too (one of our tags from the JSON file) is returned. Soft-max is a classifier and function that is being used. With Soft-max, the data can be categorized by direct classifier [5]. The number of epochs used are thousand five hundred. An epoch is one cycle through the full training dataset.

3.6 TRAINING AND SAVING THE MODEL

By training out DNN, we will be able to drop a few units, thus at each iteration we arrive at a smaller neural network and thus a great accuracy output. The number of epochs we set is the amount of times that the model will see the same information while training. Thus, each of the data is compared recursively with one another to achieve a decent accuracy rate. Once the training is complete, we can save it to the file model.tflearn for use in other scripts. The final output from the neural network will be the most matching tag from our JSON file. Finally, a random response from the respective tag is provided as output. Each tag has so many different responses of the same context to provide the user with a better experience. A new answer with different words but with the same context will improve user experience [2].



```
| Adam | epoch: 1470 | loss: 0.44166 - acc: 0.9808 -- iter: 120/373
□[A□[ATraining Step: 69059  | total loss: □[1m□[32m0.68532□[0m□[0m | time: 0.020s
| Adam | epoch: 1470 | loss: 0.68532 - acc: 0.9702 -- iter: 128/373
□[A□[ATraining Step: 69060  | total loss: □[1m□[32m0.61679□[0m□[0m | time: 0.022s
| Adam | epoch: 1470 | loss: 0.61679 - acc: 0.9732 -- iter: 136/373
□[A□[ATraining Step: 69061  | total loss: □[1m□[32m0.84293□[0m□[0m | time: 0.024s
| Adam | epoch: 1470 | loss: 0.84293 - acc: 0.9634 -- iter: 144/373
□[A□[ATraining Step: 69062  | total loss: □[1m□[32m0.75864□[0m□[0m | time: 0.025s
| Adam | epoch: 1470 | loss: 0.75864 - acc: 0.9671 -- iter: 152/373
□[A□[ATraining Step: 69063  | total loss: □[1m□[32m0.68278□[0m□[0m | time: 0.027s
| Adam | epoch: 1470 | loss: 0.68278 - acc: 0.9703 -- iter: 160/373
□[A□[ATraining Step: 69064  | total loss: □[1m□[32m0.61450□[0m□[0m | time: 0.028s
| Adam | epoch: 1470 | loss: 0.61450 - acc: 0.9733 -- iter: 168/373
□[A□[ATraining Step: 69065  | total loss: □[1m□[32m0.55305□[0m□[0m | time: 0.030s
| Adam | epoch: 1470 | loss: 0.55305 - acc: 0.9760 -- iter: 176/373
□[A□[ATraining Step: 69066  | total loss: □[1m□[32m0.78557□[0m□[0m | time: 0.031s
| Adam | epoch: 1470 | loss: 0.78557 - acc: 0.9659 -- iter: 184/373
```

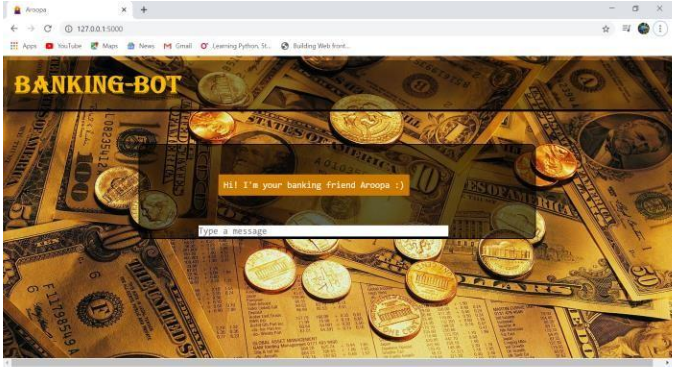
Fig-6: Epoch 1470 of training

3.7 CONNECT TO DEPLOYMENT SERVER

We use Flask Web Framework to deploy the conversational bot in real time. Python requires its own web framework like Django or Flask to connect to a website/Frontend/User Interface. Flask will act as a middle ware by fetching the user input and redirecting it to the python backend server [6]. Once the backend process is done, it will again redirect the output to the user interface. Thus, the developed project can be hosted and made available for public usage [6].

3.8 USER INTERFACE

The Design of User Interface is being developed using HTML-5, CSS-3 and Java Script. It's a simple user-friendly chatter box with a basic design.



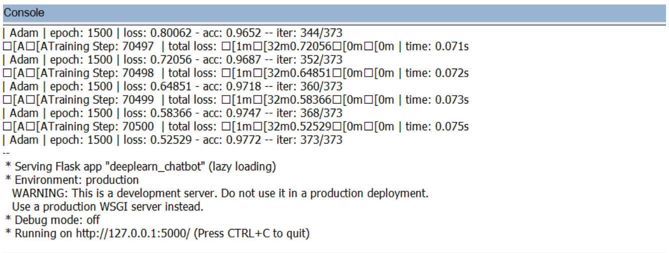
Hi! I'm your banking friend Aroopa :)

Type a message

Fig-7: User Interface of the chat-bot

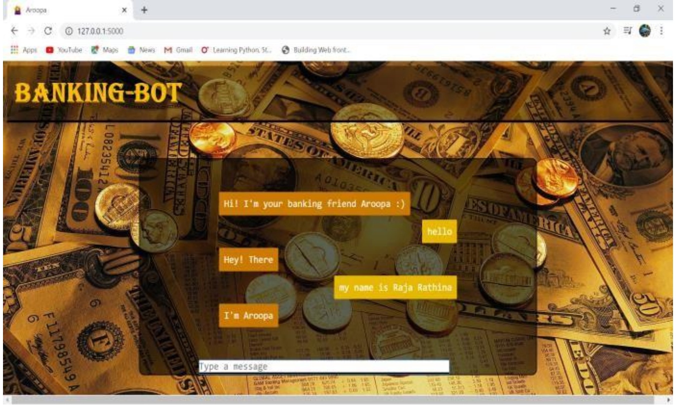
4. RESULT

This proposed approach for building a conversational bot for customer care using deep neural networks is tested on a moderately big dataset which contains nearly 500 lines of data. The training happens at rate of 1500 epoch. It takes approximately 3-4 minutes to train the entire dataset into the deep neural network. An accuracy of 97% is achieved for 60 tags of data, on deployment of the proposed neural network.



```
Console
| Adam | epoch: 1500 | loss: 0.80062 - acc: 0.9652 -- iter: 344/373
□[A□[ATraining Step: 70497  | total loss: □[1m□[32m0.72056□[0m□[0m | time: 0.071s
| Adam | epoch: 1500 | loss: 0.72056 - acc: 0.9687 -- iter: 352/373
□[A□[ATraining Step: 70498  | total loss: □[1m□[32m0.64851□[0m□[0m | time: 0.072s
| Adam | epoch: 1500 | loss: 0.64851 - acc: 0.9718 -- iter: 360/373
□[A□[ATraining Step: 70499  | total loss: □[1m□[32m0.58366□[0m□[0m | time: 0.073s
| Adam | epoch: 1500 | loss: 0.58366 - acc: 0.9747 -- iter: 368/373
□[A□[ATraining Step: 70500  | total loss: □[1m□[32m0.52529□[0m□[0m | time: 0.075s
| Adam | epoch: 1500 | loss: 0.52529 - acc: 0.9772 -- iter: 373/373
--
 * Serving Flask app "deeplearn_chatbot" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig-8: Final Accuracy of Neural Network.



Hi! I'm your banking friend Aroopa :)

hello

Hey! There

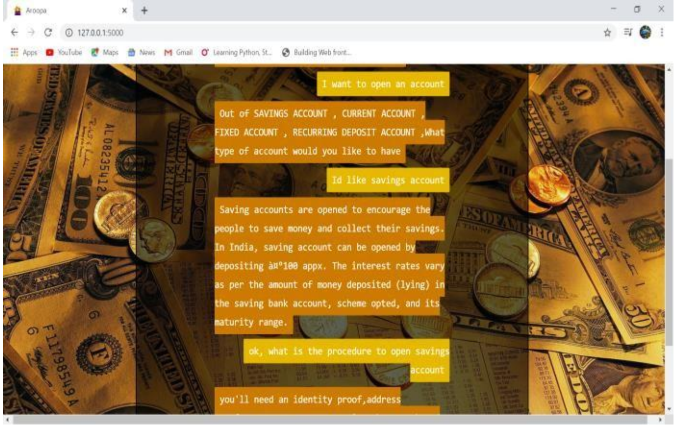
my name is Raja Rathina

I'm Aroopa

Type a message

Fig-9: Accurate replies

The above image depicts the intelligence shown by the chatbot. The user types "My name is Raja Rathina" and the bot is able to wisely answer its name in response even though the user didn't ask for its name. The bot understood that it was a message to introduce itself even though it wasn't hard coded. This shows the prediction accuracy of the chatbot.



I want to open an account

Out of SAVINGS ACCOUNT , CURRENT ACCOUNT , FIXED ACCOUNT , RECURRING DEPOSIT ACCOUNT ,What type of account would you like to have

Id like savings account

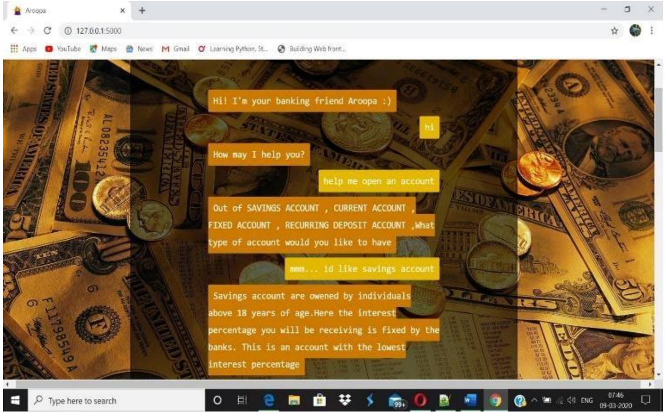
Saving accounts are opened to encourage the people to save money and collect their savings. In India, saving account can be opened by depositing à¤°100 appx. The interest rates vary as per the amount of money deposited (lying) in the saving bank account, scheme opted, and its maturity range.

ok, what is the procedure to open savings account

you'll need an identity proof,address

Fig-10: A conversation on account opening procedure.

The above image shows the conversation between the user and the bot on opening an account. The bot mentions the types of accounts. On specifying that the user needs savings account, it gives an explanation on what savings account is. Further when the user queries on the procedure to open a savings account, the bot mentions the necessary documents to be submitted to open a savings account.



Hi! I'm your banking friend Aroopa :)

hi

How may I help you?

help me open an account

Out of SAVINGS ACCOUNT , CURRENT ACCOUNT , FIXED ACCOUNT , RECURRING DEPOSIT ACCOUNT ,What type of account would you like to have

mmm... id like savings account

Savings account are owened by individuals above 18 years of age.Here the interest percentage you will be receiving is fixed by the banks. This is an account with the lowest interest percentage

Fig.11: The bot explains the terms of savings account in a different way.

The above image shows the terms of savings account different from the previous image. This depicts the random choice of responses, so that the user is provided with the same information in different vocabulary just like humans do.

5. FUTURE ENHANCEMENTS

We experimented this approach on limited set of data. In future, we would like to extend this work to a massive dataset extracted from real time conversations of banks. Further, it can be deployed to any particular bank's website. Features like transaction history and other confidential information can be made available to users by deploying private secured login ID to each customer. Thus, banking comes with ease to the globe.

6. CONCLUSIONS

In this project, we present a conversational bot that responds to user queries on the norms and procedures of a bank using Deep Neural Networks, in order to achievehigher accuracy in the response. In addition, Natural Language Processing is used to preprocess the text, so that the data fed into the neural network is of appropriate type. As a higher accuracy rate is achieved, user gets the correct answer most of the time. Instead of visiting the bank or contacting a customer care executive, user can get the information at their place and at their pace. Further, the bank can save upto 60% of the money being spent on Business Process Outsourcing. The major disadvantage of this project is that it cannot be used without an internet connection.

REFERENCES

[1] https://expertsystem.com/chatbot/

[2] P. Lukáč and P. Tarábek, "Improving DNN Solution using Repeated Training," 2019 International Conference on Information and Digital Technologies (IDT), Zilina, Slovakia, 2019, pp. 311-315, doi: 10.1109/DT.2019.8813418.

[3] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," in IEEE Access, vol. 7, pp. 32664-32671,2019,doi: 10.1109/ACCESS.2019.2903331.

[4] W. A. Qader, M. M. Ameen and B. I. Ahmed, "An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges," 2019 International Engineering Conference (IEC), Erbil, Iraq, 2019, pp. 200-204, doi: 10.1109/IEC47844.2019.8950616.

[5] F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 755-758, doi: 10.1109/UBMK.2017.8093521.

[6] https://flask.palletsprojects.com/en/1.1.x/

[7] E. Wahyudi, S. Sfenrianto, M. J. Hakim, R. Subandi, O. R. Sulaeman and R. Setiyawan, "Information Retrieval System for Searching JSON Files with Vector Space Model Method," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 260-265, doi: 10.1109/ICAIIT.2019.8834457.

[8] https://medium.com/towards-artificial-intelligence/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0