**StackOverflow Assistant Chatbot Using NLP**

*Submitted in partial fulfillment of the requirements for the degree of*

Bachelor of Technology

in

**Electronics and Communication Engineering**

*by*

**Jasmine Batra**

**17BEC0125**

Under the guidance of

**Dr. Sankar Ganesh S**

**School of Electronics Engineering**

**VIT, Vellore**



May, 2021

# **DECLARATION**

I hereby declare that the thesis entitled “StackOverflow Assistant Chatbot Using NLP” submitted by me, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Prof. Sankar Ganesh S.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 23.05.2021

Jasmine Batra

**Signature of the Candidate**

# **CERTIFICATE**

This is to certify that the thesis entitled “StackOverflow Assistant Chatbot Using NLP” submitted by **Jasmine Batra (17BEC0125)**, **School of Electronics Engineering**, VIT, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering*, is a record of bonafide work carried out by her under my supervision during the period, 01.02.2021 to 23.05.2021, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the university and, in my opinion, meets the necessary standards for submission.

Place: Vellore

Date: 23.05.2021

Sankar Ganesh S

**Signature of the Guide**

**Internal Examiner External Examiner**

Dr. Prakasam P

Electronics and Communication Engineering

# **ACKNOWLEDGEMENTS**

It is a privilege to express my sincerest regards to my project coordinator, Dr. Sankar Ganesh S, for his valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of my project. I deeply express my sincere thanks to my School Dean and HOD, and the University Management for encouraging and allowing me to present the project on the topic “StackOverflow Assistant Chatbot Using NLP”.

I also take this opportunity to thank all my lecturers who have directly or indirectly helped me in my project. Furthermore, I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report.

Last but not the least I express my thanks to my friends for their cooperation and support.

**JASMINE BATRA**

# **Executive Summary**

Searching on the Stack Overflow website can sometimes be arduous and time-consuming. The thesis aims to create a conversational Chatbot to assist with Stack Overflow search that saves time.

An Intent-Classifier will determine whether the user’s question is a Stack Overflow question (programming question) or a dialogue question (non-programming question).

For a Stack Overflow question, the Programming Language (Tag)-Classifier predicts the programming language of the question so that we only search for those language questions in our database. Given the question and its programming language, cosine similarity is used to get the most similar question. The bot then replies with a Stack Overflow link to that question.

For a dialogue (chit-chat) question, we’ll use the ChatterBot python library that has a pre-trained neural network engine, to respond to a user’s queries.

The bot is integrated with Telegram messenger that serves as a medium for a user to ask questions and for the bot to respond to them. Results show that the used algorithms are in accordance with the implementation of the Chatbot approach with good test accuracies. This Chatbot will help users find answers to programming questions that they aren’t able to solve and also hold conversations with them.

**CONTENTS Page**

**No.**

**Acknowledgements** 4

[Executive Summary 5](file:///C:\Users\Admin\Desktop\Capstone%20Project\Reviews\Capstone%20Project%20Report%20Format.docx#_TOC_250004)

Table of Contents 6

[List of Figures 8](file:///C:\Users\Admin\Desktop\Capstone%20Project\Reviews\Capstone%20Project%20Report%20Format.docx#_TOC_250003)

List of Tables 9

List of Abbreviations 10

1. [INTRODUCTION 1](file:///C:\Users\Admin\Desktop\Capstone%20Project\Reviews\Capstone%20Project%20Report%20Format.docx#_TOC_250000)1
   1. Objective 11
   2. Motivation 11
   3. Background 11
2. PROJECT DESCRIPTION AND GOALS 13
3. TECHNICAL SPECIFICATION 13
4. **DESIGN APPROACH AND DETAILS** 15
   1. Design Approach 15
   2. Codes and Standards 16
   3. Constraints, Alternatives and Tradeoffs 19
5. **SCHEDULE, TASKS, AND MILESTONES** 19
6. **PROJECT DEMONSTRATION** 20
7. **RESULTS & DISCUSSIONS** 27

|  |  |  |
| --- | --- | --- |
| 8 | **SUMMARY** | 29 |
| 9 | **REFERENCES** | 29 |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
|  |  |

# **List of Figures**

**Figure No. Title Page No.**

1. A sample of tagged\_posts.tsv 14
2. A sample of dialogues.tsv 14
3. Flowchart of our StackOverflow Assistant Chatbot 15
4. Training data for Intent-Classifier 20
5. Accuracy of Intent-Classifier 21
6. Sample output of Intent-Classifier 21
7. Training data for Programming Language-Classifier 21
8. Accuracy of Programming Language-Classifier 22
9. Sample output of Programming Language-Classifier 22
10. Finding the post\_id of the most similar question 22
11. Stack Overflow thread of the returned post\_id’s question 22
12. Bot creation in Telegram 23
13. Conversation with the bot in Telegram 25
14. A dictionary about the message (question) sent by 25

the user in the terminal window

1. Intent-Classifier, Tag-Classifier, and Tfidf Vectorizer 27

models used, stored in my Google drive project repository

1. .pkl files for the ten programming languages considered 28

here, created and stored in my Google drive project

repository

17 A sample of java.pkl 28

|  |  |
| --- | --- |
|  |  |

# **List of Tables**

**Table No. Title Page No.**

1. Timeline of my project 20
2. Sample output of Chatbot 26

# **List of Abbreviations**

UI User Interface

VB Visual Basic

.pkl pickle

seq2seq sequence-to-sequence

# **1. INTRODUCTION**

## **Objective**

Building a conversational Chatbot that will assist with search on the Stack Overflow website.

To build a dialogue Chatbot that will be able to:

* Answer questions related to programming.
* Simulate dialogue and chit-chat on all non-programming related questions.

For programming questions, the Stack Overflow dataset will be used.

For non-programming questions that require a chit-chat mode, a pre-trained neural network engine available from the [ChatterBot](https://github.com/gunthercox/ChatterBot) python library will be used.

## **Motivation**

The primary motivation of this project is to build something that is useful for learning (study/work) purposes that save time.

Stack Overflow serves as a question-answering site for the programming community that features questions and answers on an extensive range of computer programming topics.

It is one of the most widely used applications by programming enthusiasts to look up answers for questions that they aren’t able to solve – by an engineering student for his studies (assignments), a software working professional for his work (projects), and tech enthusiasts for acquiring knowledge, but not everyone finds time to search for a particular question and look into the answers in Stack Overflow through search engines with ease. And even on searching, they get multiple questions/answers to examine to find the best one, making it all the more gruelling. So, I wanted to build something that would help people in searching for their doubts/questions on Stack Overflow and getting the correct answers (most similar question) and at the same time to chit-chat with the user - this Chatbot does that. The bot ‘JasmineStackBot’ is a conversational bot that interacts with the user, and whenever a user asks a programming question, it responds with the Stack Overflow link to the most similar question.

## **Background**

A Chatbot is an AI-based computer program that can talk to humans in natural language. It understands human language, processes it, and interacts back with humans while performing specific tasks [12].

Chatbots can be logically divided into the following two categories:

* Database/FAQ based — There exists a database with some questions and answers, and the bot responds to a user’s query using the database.
* Chit-Chat Based — Simulate dialogue and hold conversations with the user.

Chatbots are designed mainly using two approaches:

* In a Rule-based method, a bot responds to questions based on certain pre-programmed rules. The defined rules can range from simple to complex. The bots can deal with straightforward queries but not complicated ones.
* Self-learning bots are those that employ Machine Learning techniques and are far more efficient than rule-based bots.

Related Works

N. N. Khin and K. M. Soe [1] used the seq2seq model with Attention Mechanism based on the RNN encoder-decoder model to explore ways of communication by neural network Chatbots. This Chatbot is designed for use in the university education sector to answer frequently asked questions about the university and its related details.

B. Setiaji and F. W. Wibowo [2] focused on the machine being programmed with the ability to recognize sentences and make decisions on its own in response to a question. This work employs bigram to calculate sentence similarity, which divides the input sentence into two characters. The higher the score, the more similar the reference sentences are. Chatbot’s knowledge is stored in a database. In relational database management systems (RDBMS), the Chatbot comprises a core and an interface that accesses that core.

M. Shen and R. Huang [3] describe how data collected when users conduct conversations using WeChat social network application can be used to enhance people’s lives as well as build a customized Chatbot based on personal conversation history. This work uses a cognitive map based on the word2vec model to learn and store the relationship between each word in the chatting records. A vector in a continuous high-dimensional vector space will be used to represent each word. They used the seq2seq method on all pairs of chatting sentences to learn chatting styles.

M.Y.H. Setyawan, R.M. Awangga, and S.R.Efendi [4] propose a classification method called intent classification on the Chatbot system to determine intent rather than user input. They compare the Naive Bayes and Logistic Regression methods for classifying data and determining the degree of recall, accuracy, and precision of both methods’ evaluation results in this analysis. According to the evaluation results, the Logistic Regression model has a higher degree of recall, accuracy, and precision than the Naive Bayes model.

# **2. PROJECT DESCRIPTION AND GOALS**

Building a conversational Chatbot to help with Stack Overflow search.

After the user asks a question, an Intent-Classifier will predict if the question asked is a Stack Overflow question (programming question) or a dialogue question (non-programming question). The bot will determine the intent using Intent-Classifier and distinguish programming-related questions from general ones.

If the question asked is a Stack Overflow question, the bot will respond to the question asked by tagging it with the corresponding programming language using Programming Language (Tag)-Classifier. If the question is a Stack Overflow question, this classifier will predict which language (tag) it belongs to. This narrows our search to only those language questions in our database. Every question in the dataset is converted to an embedding (vector), and the database contains an embeddings file for every programming language individually. This file contains the vector representation (sentence embeddings) of all questions of that programming language. Ten programming languages are considered here – C/C++ (c\_cpp), C#, Java, JavaScript, PHP, Python, R, Ruby, Swift, VB. Given that we know the question and the programming language of that question, cosine similarity is used to get the most similar question, and the bot responds with the Stack Overflow Link to that question.

If the question asked is a chit-chat question, the chatterbot will handle it. For a chit-chat mode, we will use a pre-trained neural network engine available from the [ChatterBot](https://github.com/gunthercox/ChatterBot) python library.

Telegram is set up to make our Chatbot communicate with it using the Access token. The bot will be integrated with Telegram messenger so that we can now talk to this bot in Telegram.

# **3. TECHNICAL SPECIFICATION**

Resources to build our Chatbot

**Google Colab** to train the model.

* A free online cloud-based Jupyter notebook environment used to train and evaluate our model.

**Atom** to link our model and the bot.

* A free and open-source text editor used to establish a connection between our model and the Telegram bot.

**Stack Overflow and Dialogues** dataset.

* **tagged\_posts.tsv** — Stack Overflow posts, tagged with one programming language (positive samples).

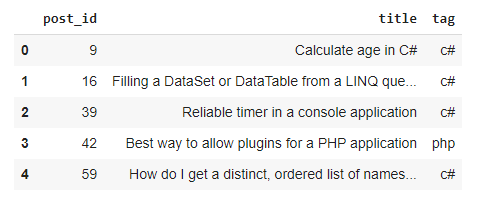


Figure 1. A sample of tagged\_posts.tsv

* **dialogues.tsv** — dialogue phrases from moves subtitles (negative samples).

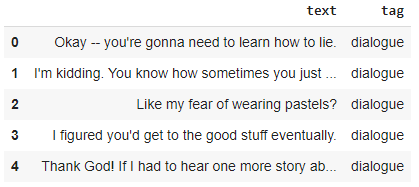


Figure 2. A sample of dialogues.tsv

To detect the intent of users’ questions (Intent-Classifier), we will use:

* dialogues.tsv
* tagged\_posts.tsv

If the question is a Stack Overflow question, to predict which language (tag) it belongs to (Programming Language-Classifier), we will use:

* tagged\_posts.tsv

**Chatterbot** library for dialogue (chit-chat) type questions.

* A Python library to enable our Chatbot to provide automated responses for chit-chat type questions.

**GoogleNews-vectors** to convert every question to an embedding.

* A [pre-trained](https://code.google.com/archive/p/word2vec/) word2vec model from Google, which was trained on a portion of the Google News dataset (about 100 billion words). In the model, 300-dimensional vectors represent three million words and phrases.

**Telegram** to instantiate the bot

* A cloud-based instant messaging software used as a medium for a user to talk to the bot by creating a Chatbot UI and connecting it to the telegram app back-end, and running our Chatbot logic.

# **4. DESIGN APPROACH AND DETAILS**

# **4.1 Design Approach**

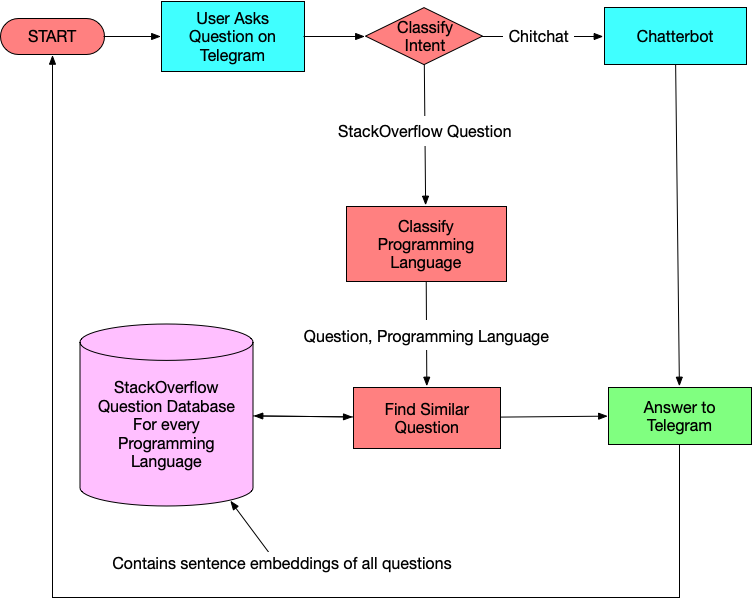


Figure 3. Flowchart of our StackOverflow Assistant Chatbot

# **4.2 Codes and Standards**

Model Creation (Model\_Creation.ipynb)

* Our model creates TF-IDF Vectorizers, Intent-Classifier model, Programming Language-Classifier model, and embeddings for each programming language considered here.
* Texts are pre-processed, TF-IDF transformations are applied on them, and the TF-IDF vectorizer is dumped.
* To create our Intent-Classifier, we first prepare the data for it, create features with a TF-IDF Vectorizer and then train a Logistic Regression Model.
* To create our Programming Language-Classifier, we first prepare the data for it, create features with a TF-IDF Vectorizer and then train a OneVsRestClassifier Logistic Regression model.
* Every question is converted to an embedding using pre-trained word vectors (word2vec model) from Google and stored that are categorized by the programming language.

Procedure

1. Import Libraries

* Import the required libraries.

1. Read the Data

* Read the dataset files and store them as a data frame.

1. Create training data for intent classifier

* Concatenate dialogue and Stack Overflow examples into one sample.
* Pre-process the texts and split the data into a training set and test set in a 9:1 ratio.

1. Create Intent-Classifier

* Transform the train set and test set into TF-IDF features.
* Do a binary classification on TF-IDF representations of texts.
* Labels will be either dialogue for general questions or stackoverflow for programming-related questions.
* Train the intent recognizer using Logistic Regression on the train set.
* Check out the accuracy on the test set to check whether everything looks good.
* Dump the TF-IDF vectorizer and the classifier with pickle to use it later in the running bot.

1. Create Programming Language-Classifier

* Prepare the data for this task and split the data into a training set and test set in an 8:2 ratio.
* Reuse the TF-IDF vectorizer that we have already created.
* Train the tag classifier using OneVsRestClassifier wrapper over LogisticRegression.
* Check out the accuracy on the test set.
* Dump the classifier to use it in the running bot.

1. Store Question database Embeddings

* Load GoogleNews-vectors-negative300.bin, a [pre-trained](https://code.google.com/archive/p/word2vec/) word2vec model from Google.
* Each question is converted to an embedding and stored, so we don’t have to recalculate the embeddings for the entire dataset each time.
* Whenever a Stack Overflow question is asked, use cosine similarity to find the most similar question.
* For each programming language (tag), create a .pkl file with two data structures, which will serve as an online search index:
* tag\_post\_ids — a list of post\_ids that will be needed to show the title and link to the thread.
* tag\_vectors — a matrix where embeddings for each question are stored.

1. Given a question and tag, to retrieve the most similar question’s post\_id

* Load the question’s tag’s .pkl file.
* Convert the question to a vector and compute the minimum distance between this vector and tag\_vectors (set of vectors) to find the index of the most similar post.
* In essence, to have a function that returns the post id of the most similar question in the dataset given that we know the question and the programming language of the question.

Telegram Setup (main.py)

* Telegram makes it simple to design a Chatbot UI.
* It gives us an access token that we’ll use to connect to Telegram’s back-end using its API and run our Chatbot logic.
* Naturally, we’ll need a window to type our questions to the Chatbot, which Telegram provides for us.
* Additionally, the Chatbot is powered by Telegram, which communicates with our Chatbot logic and the models created.

Procedure

* Set up a bot by talking to the BotFather in Telegram and creating a name/user name for the bot.
* I will use main.py to make our Chatbot communicate with Telegram using the access token.
* BotHandler class implements all back-end of the bot using Telegram’s API. It has three main functions:
* get\_updates - checks for new messages sent by the user.
* get\_answer - computes the most relevant answer to a user’s question using a SimpleDialogueManager class.
* send\_message – posts the answer computed as a new message to the user.
* SimpleDialogueManager class is where we will write our bot logic that fits the pieces together to build one wholesome logic.
* All the models and TFIDF objects are instantiated (.pkl files)
* A Chatbot is instantiated using ChatterBot and trained on the [provided English corpus](https://github.com/gunthercox/chatterbot-corpus/tree/master/chatterbot_corpus/data/english) data for chit-chat type questions.
* get\_similar\_question function - given the question and the question’s programming language (tag), loads the particular tag’s post\_ids and post\_embeddings from the .pkl file, converts the question to a vector and computes the minimum distance between this vector and post\_embeddings (set of vectors) to find post id of the most similar question in the dataset.
* generate\_answer function - transforms the question using the loaded tfidf\_vectorizer and determines the intent of the question. For a dialogue question, it generates a response using ChatterBot. For a programming question, it finds the tag (programming language) and, using the get\_similar\_question function, generates the thread (Stack Overflow link) of the question that is the most similar as a response.

# **4.3 Constraints, Alternatives, and Tradeoffs**

I’ve used ChatterBot, a python library, to provide automated responses for chit-chat type questions. The code in init (in SimpleDialogueManager class in main.py) instantiates a Chatbot with ChatterBot and trains it on the English corpus data provided. The data isn’t too large. I could have done the same thing with a seq2seq model [7], used other Python libraries, or trained it on my own dataset too. But since the main objective here is to create a Chatbot to assist with Stack Overflow search and not worry too much about the responses to chit-chat type questions, I work with the ChatterBot library and train it on the English corpus data.

I will be creating a TFIDF model with Logistic regression to prepare data and train the classifiers (Intent-Classifier and Programming Language-Classifier). Other machine learning classification algorithms, such as Naive Bayes, Decision trees, or one of the deep learning models or transfer learning techniques, could have been employed instead. But since the Logistic regression model has a higher degree of recall, accuracy, and precision than the other models [4], I work based on Logistic Regression with TF-IDF features.

I’ve used GoogleNews-vectors, a pre-trained word2vec model from Google , to convert every question to an embedding [3]. I could have done better by training my embeddings using StarSpace embeddings since StarSpace embeddings are trained using supervised data, such as a set of related sentence pairings. Unfortunately, for StarSpace to be run on Windows, we’ll need to install Boost libraries (as a dependency for StarSpace), and that’s a pretty arduous task on Windows or use Docker container. Given the complications in using StarSpace embeddings and considering the good accuracy and precision when using pre-trained vectors, I chose to work with pre-trained word vectors from Google.

# **5. SCHEDULE, TASKS, AND MILESTONES**

Table 1. Timeline of my project

|  |  |  |
| --- | --- | --- |
| **Review** | **Month** | **Tasks** |
| 1 | Feb 2021 | Understanding of the project, the objective and tool requirements, and the formulation of the project plan. |
| 2 | Feb-Mar 2021 | Creation of Intent-Classifier and Programming Language-Classifier. |
| Setting up of Telegram to make the Chatbot communicate with it using BotHandler class. |
| Demonstration of a simple Chatbot using Telegram and testing for the accuracies of the classifiers created. |
| 3 | Apr-May 2021 | Instantiation of a Chatbot using ChatterBot for chit-chat type questions. |
| Storage of question database embeddings for each programming language to get the most similar question to the one the user has asked. |
| Fitting all the pieces in our SimpleDialogueManager Class in our Telegram Bot Handler that responds to the questions the user has asked. |
| Demonstration of the complete working project of the bot responding to a user’s queries via Telegram. |

# **6. PROJECT DEMONSTRATION**

**Intent-Classifier**

Training data

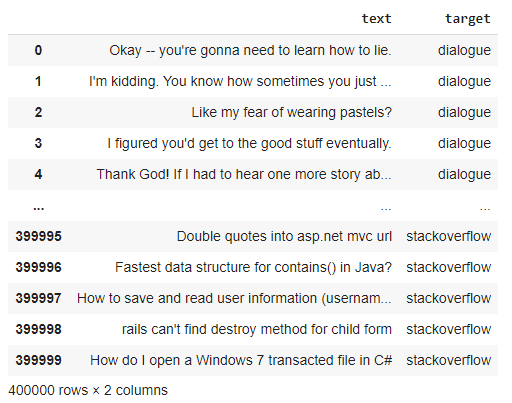


Figure 4. Training data for Intent-Classifier

Testing Accuracy

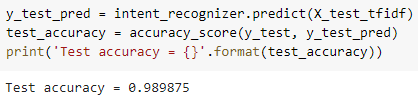


Figure 5. Accuracy of Intent-Classifier

Sample Output

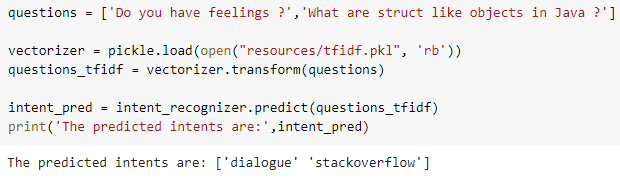


Figure 6. Sample output of Intent-Classifier

**Programming Language-Classifier**

Training data

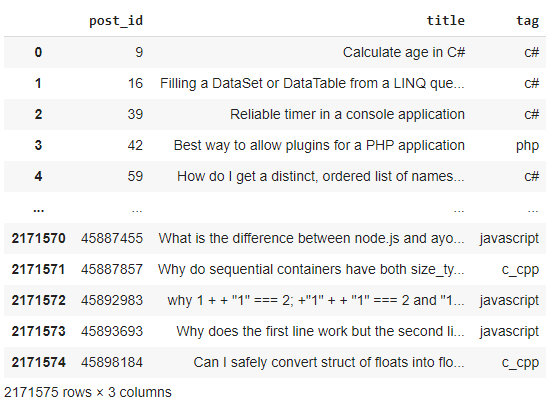


Figure 7. Training data for Programming Language-Classifier

Testing Accuracy

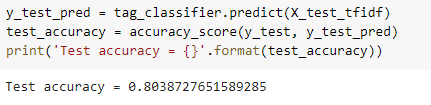


Figure 8. Accuracy of Programming Language-Classifier

Sample Output

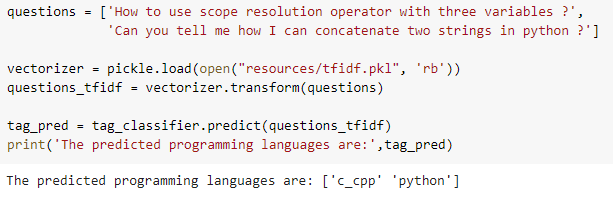


Figure 9. Sample output of Programming Language-Classifier

**Given a question and tag, to retrieve the most similar question’s post\_id**

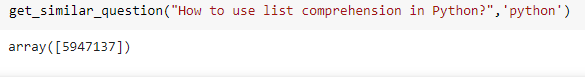


Figure 10. Finding the post\_id of the most similar question

You can find this question at:

<https://stackoverflow.com/questions/5947137>

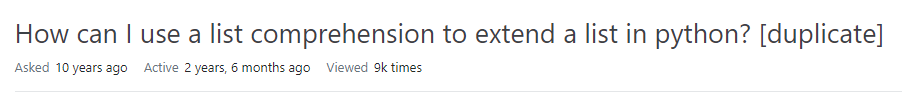


Figure 11. Stack Overflow thread of the returned post\_id’s question

**Telegram Setup (Integration of the bot with Telegram)**

Bot creation

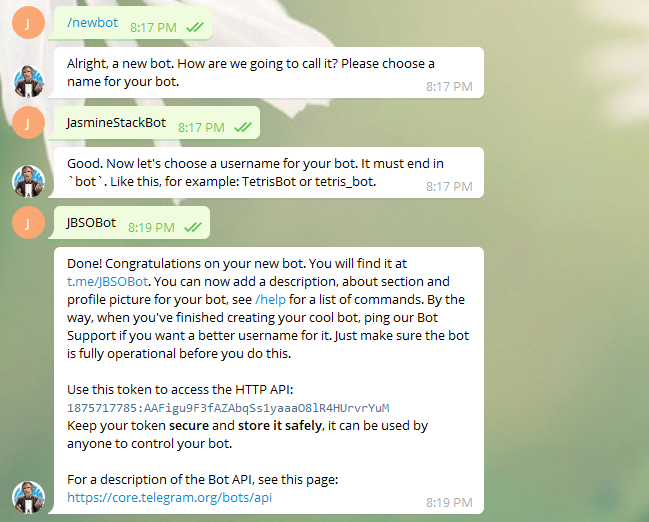
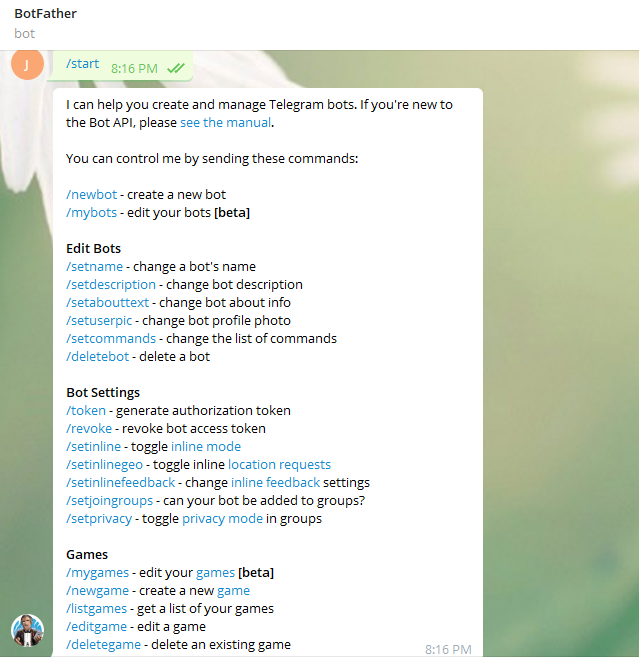
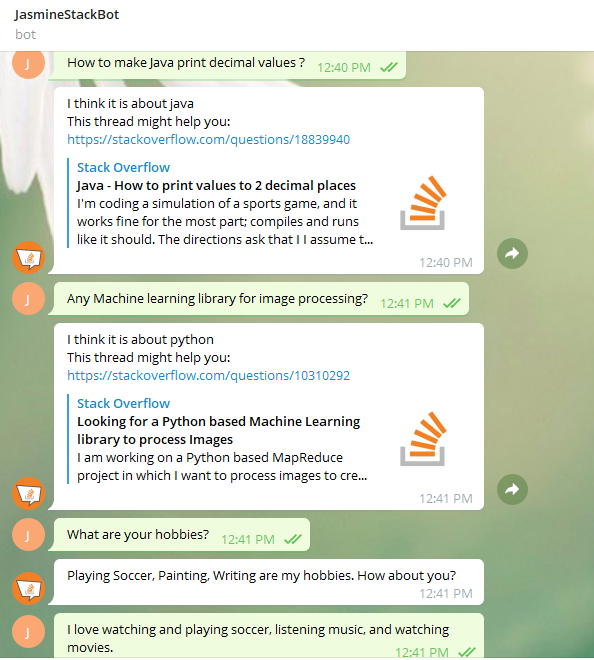
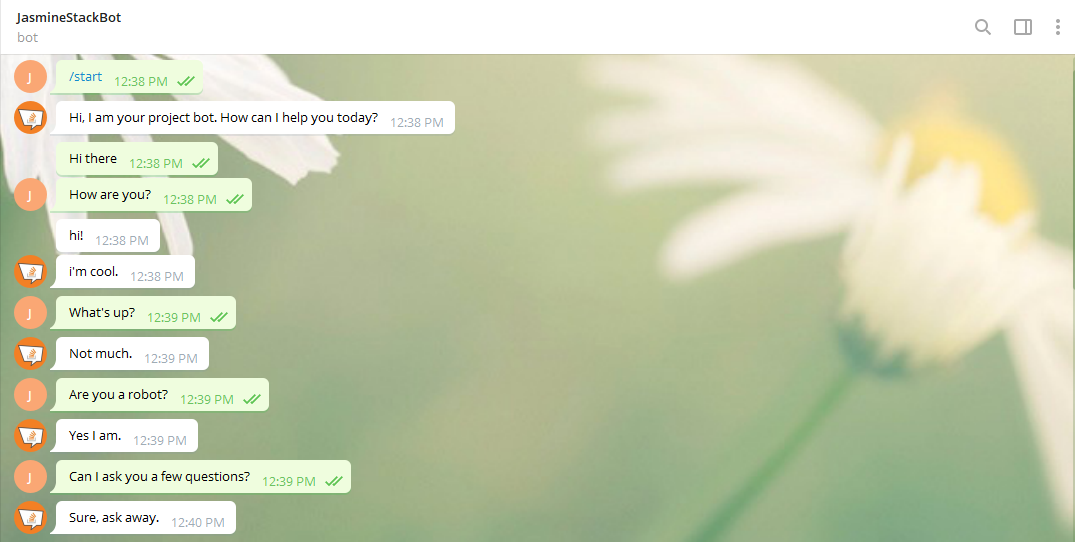


Figure 12. Bot creation in Telegram

Conversation with the bot



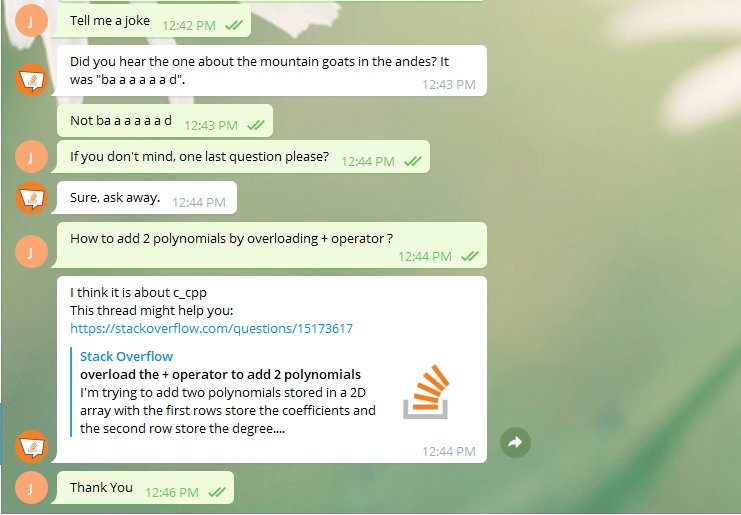


Figure 13. Conversation with the bot in Telegram

A sample of terminal window output

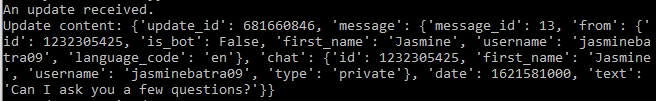


Figure 14. A dictionary about the message (question) sent by the user in the terminal window

Table 2. Sample output of Chatbot

|  |  |  |  |
| --- | --- | --- | --- |
| **User Text** | **Intent-Classifier** | **Programming Language-Classifier** | **Response** |
| Where are you from? | dialogue | - | I am from where all software programs are from; a galaxy far, far away. |
| Can I replace plain URLs with links? | stackoverflow | javascript | I think it is about javascript  This thread might help you: <https://stackoverflow.com/questions/19547008>  (How to replace plain URLs with links, with example?) |
| Are you sentient? | dialogue | - | Sort of. |
| How to create a static class in cpp? | stackoverflow | c\_cpp | I think it is about c\_cpp  This thread might help you: <https://stackoverflow.com/questions/9321>  (How do you create a static class in C++?) |
| How are you doing? | dialogue | - | I am doing well. |
| Sorting a dictionary in python | stackoverflow | python | I think it is about python  This thread might help you: <https://stackoverflow.com/questions/7546170>  (Sorting a dictionary in python) |
| What’s your favourite subject? | dialogue | - | My favorite subjects include robotics, computer science, and natural language processing. |
| How to sort a 2d array using linq? | stackoverflow | c# | I think it is about c#  This thread might help you:  (<https://stackoverflow.com/questions/7918080>)  (How can I sort a 2d array using Linq?) |
| Suggest me a movie to watch | dialogue | - | What kind of movies do you like? |
| Explain Multithreading in Java | stackoverflow | java | I think it is about java  This thread might help you: <https://stackoverflow.com/questions/5731239>  (Multithreading in Java) |

Access all the project resources here:

<https://drive.google.com/drive/folders/1k0_XatdlWiyyPbjo3tNalfZH7LGriCjv?usp=sharing>

# **7. RESULTS & DISCUSSIONS**

Results

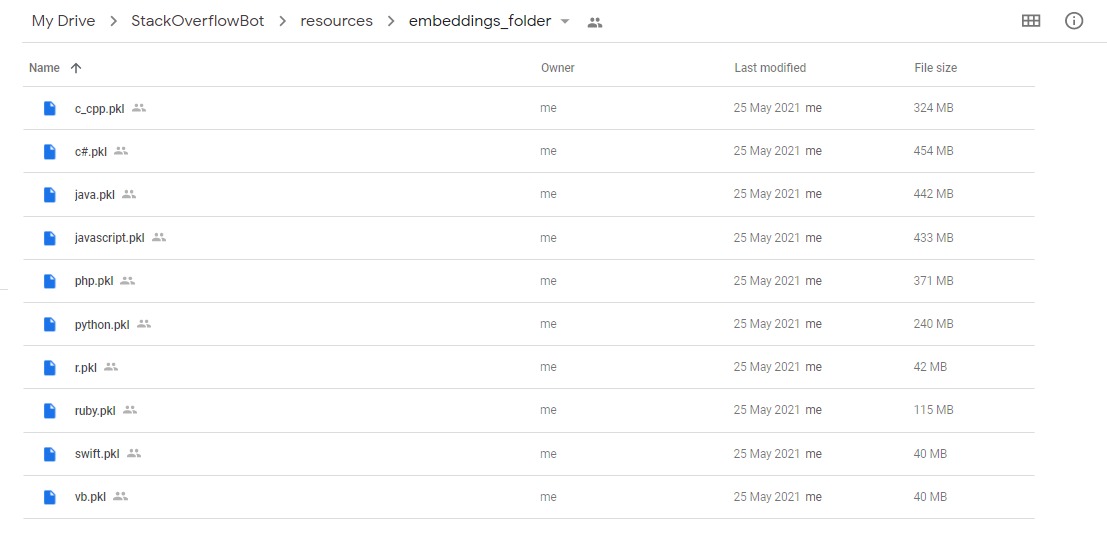
The bot responds to a programming question with a Stack Overflow link for the question asked and simulates dialogue for a non-programming question.TFIDF vectorizers have been created and saved as tfidf.pkl in my project repository (resources). Two classifiers have been created:

1. Intent-Classifier that will predict if a question is a dialogue question or a Stack Overflow question with a test accuracy of 98.98%. It is saved as intent\_clf.pkl in my project repository (resources).

2. Programming Language (Tag)-Classifier that will predict the language of a Stack Overflow question with a test accuracy of 80.38%. It is saved as tag\_clf.pkl in my project repository (resources).

 Fig 15. Intent-Classifier, Tag-Classifier, and Tfidf Vectorizer models used, stored in my Google drive project repository

A .pkl file for every programming language (tag) that contains the tag’s post IDs and the embeddings for each question of that tag are stored in my project repository (resources/embeddings\_folder).

Fig 16. .pkl files for the ten programming languages considered here, created and stored in my Google drive project repository

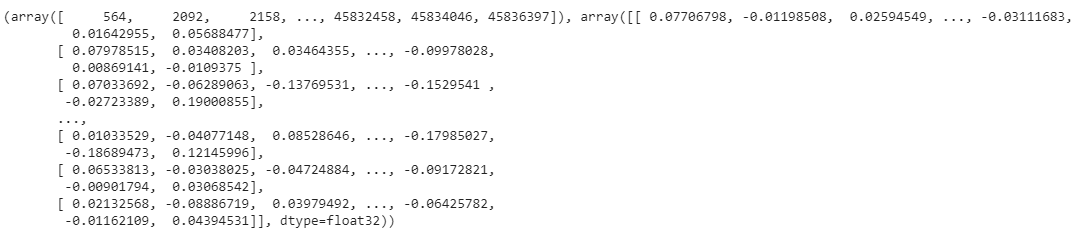


Fig 17. A sample of java.pkl

Telegram has been set up to show how our Chatbot responds to users’ queries. In the terminal window, we also get a dictionary about the message sent by the user (question) that contains a Unique Chat ID, Chat Text, User Information, etc., which we can use as per our requirements later.

Discussions

We can increase the accuracy of the classifier, handle edge cases, make it reply faster, or add more logic to handle more use cases to improve on this Chatbot. For a chit-chat mode, I have used a pre-trained neural network engine available from ChatterBot. We can also use seq2seq models or train our own models to create such bots [9]. I’ve used GoogleNews-vectors to convert every question to a vector. We can also use StarSpace embeddings for the same. In the near future, I plan to extend my work on a large-scale study to answer questions from all domains, i.e., open-domain question answering [6].

# **8. SUMMARY**

In this project, I’ve proposed an approach for designing and building an interactive Chatbot that does question-answering. The proposed approach includes different classifiers, stored question database embeddings, telegram bot handler and their implementations. Experimental results show that the selected algorithms are in accordance with the implementation of the Chatbot approach with good test accuracies. Telegram is used as a frontend medium to ask questions to the bot, which then responds back using the trained models in its back-end. The Chatbot will assist people in searching for solutions to programming questions that they would need (at work or study) and also hold conversations with the user.

# **9. REFERENCES**

[1] N. N. Khin and K. M. Soe, “Question Answering based University Chatbot using Sequence to Sequence Model,” 2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), Yangon, Myanmar, 2020, pp. 55-59.

[2] B. Setiaji and F. W. Wibowo, “Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling,” 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Bangkok, 2016, pp. 72-77.

[3] Shen, M. and Huang, R., 2018, July. A personal conversation assistant based on Seq2seq with Word2vec cognitive map. In 2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI) (pp. 649-654). IEEE.

[4] Setyawan, M.Y.H., Awangga, R.M. and Efendi, S.R., 2018, October. Comparison of multinomial naive bayes algorithm and logistic regression for intent classification in Chatbot. In 2018 International Conference on Applied Engineering (ICAE) (pp. 1-5). IEEE.

[5] L. T. Hien, L. Tran Thi Ly, C. Pham-Nguyen, T. Le Dinh, H. Tiet Gia and L. N. Hoai Nam, “Towards Chatbot-based Interactive What- and How-Question Answering Systems: the Adobot Approach,” 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh, Vietnam, 2020, pp. 1-3, doi: 10.1109/RIVF48685.2020.9140742.

[6] Quarteroni, S. and Manandhar, S., 2007. A chatbot-based interactive question answering system. *Decalog 2007*, *83*.

[7] Mutiwokuziva, M.T., Chanda, M.W., Kadebu, P., Mukwazvure, A. and Gotora, T.T., 2017, October. A neural-network based chat bot. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)* (pp. 212-217). IEEE.

[8] Ranoliya, B.R., Raghuwanshi, N. and Singh, S., 2017, September. Chatbot for university related FAQs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1525-1530). IEEE.

[9] Palasundram, K., Sharef, N.M., Nasharuddin, N., Kasmiran, K. and Azman, A., 2019. Sequence to sequence model performance for education chatbot. *International Journal of Emerging Technologies in Learning (iJET)*, *14*(24), pp.56-68.

[10] El Zini, J., Rizk, Y., Awad, M. and Antoun, J., 2019, July. Towards a deep learning question-answering specialized chatbot for objective structured clinical examinations. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-9). IEEE.

[11] Sreelakshmi, A.S., Abhinaya, S.B., Nair, A. and Nirmala, S.J., 2019, November. A Question Answering and Quiz Generation Chatbot for Education. In *2019 Grace Hopper Celebration India (GHCI)* (pp. 1-6). IEEE.

[12] Akhtar, M., Neidhardt, J. and Werthner, H., 2019, July. The potential of chatbots: analysis of chatbot conversations. In *2019 IEEE 21st Conference on Business Informatics (CBI)* (Vol. 1, pp. 397-404). IEEE.