1. INTRODUCTION

1.1 Objective

To create a dialogue Chatbot, that will be able to:

- Answer programming-related questions (using StackOverflow dataset).
- Chit-Chat and simulate dialogue on all non-programming related questions.

For a chit-chat mode we will use a pre-trained neural network engine available from ChatterBot python library.

1.2 Motivation

The main motivation of this project is to build something that is useful for learning (study/work) purposes that saves times.

Stack Overflow is a question and answer site for professional and enthusiast programmers that features questions and answers on a wide range of topics in computer programming.

It is one of the most used applications by a software student for his studies, a working professional for his work and tech enthusiasts for their learning, but not everyone find time to search for a particular question in stack overflow through search engines. And even on searching, they get multiple resources to refer from which all the more makes it gruelling. So, I wanted to build something that would help people in searching for their doubts/questions on stack overflow and getting the right answers (most similar question) and at the same time chit-chats 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.

1.3 Background

A chatbot is a computer program that can talk to humans in natural language. It can understand human language, process it and interact back with humans while performing specific tasks.

We can logically divide of chatbots in the following two categories.

- Database/FAQ based — We have a database with some questions and answers, and we would like that a user can query the database using Natural Language.
- Chit-Chat Based — Simulate dialogue with the user.

There are mainly two approaches used to design the chatbots:

- In a Rule-based approach, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex. The bots can handle simple queries but fail to manage complex ones.
- Self-learning bots are the ones that use some Machine Learning-based approaches and are definitely more efficient than rule-based bots. These bots can be further classified in two types: Retrieval Based or Generative.

Related Works

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 Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), Yangon, Myanmar, 2020, pp. 55-59.

- In the paper, the authors explored the ways of communication through neural network chatbot by using the Sequence to Sequence model with Attention Mechanism based on RNN encoder decoder model.
- This chatbot is intended to be used in university education sector for frequently asked questions about the university and its related information.

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.

- In this paper the machine has been embedded knowledge to identify the sentences and make a decision itself as response to answer a question.
- This work uses bigram for sentence similarity calculation which divides input sentence as two letters of input sentence. The higher the score obtained the more is the similarity of reference sentences.
- The knowledge of chatbot is stored in the database. The chatbot consists of core and interface that is accessing that core in relational database management systems (RDBMS).

2. PROJECT DESCRIPTION AND GOALS

The bot will be able to:

- Answer programming-related questions (using StackOverflow dataset)
- Chit-Chat and simulate dialogue on all non-programming related questions

Have created two classifiers and saved them as .pkl files:

1. Intent-Classifier: This classifier will predict if a question is a Stack-Overflow question or not. If it is not a Stack-overflow question, the Chatterbot will handle it.

2. Programming-Language(Tag) Classifier: This classifier will predict which language(tag) a question belongs to if the question is a Stack-Overflow question. By doing this we only search for those language questions in our database.

The bot will determine the intent using Intent-Classifier and distinguish programming related questions from general ones.

The bot will respond to the programming question asked by tagging it with the corresponding programming language using Programming-Language(Tag) Classifier and find the most relevant Stack-Overflow Link.

For general questions, the chatterbot will handle it.For a chit-chat mode we will use a pre-trained neural network engine available from ChatterBot python library.

The bot will be integrated with Telegram messenger so that we can now talk to this bot in Telegram and it'll also be hosted on AWS.

Telegram been setup to make our Chatbot communicate with it using the Access token.

3. TECHNICAL SPECIFICATION

Dataset

To detect intent of users questions we will use:

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

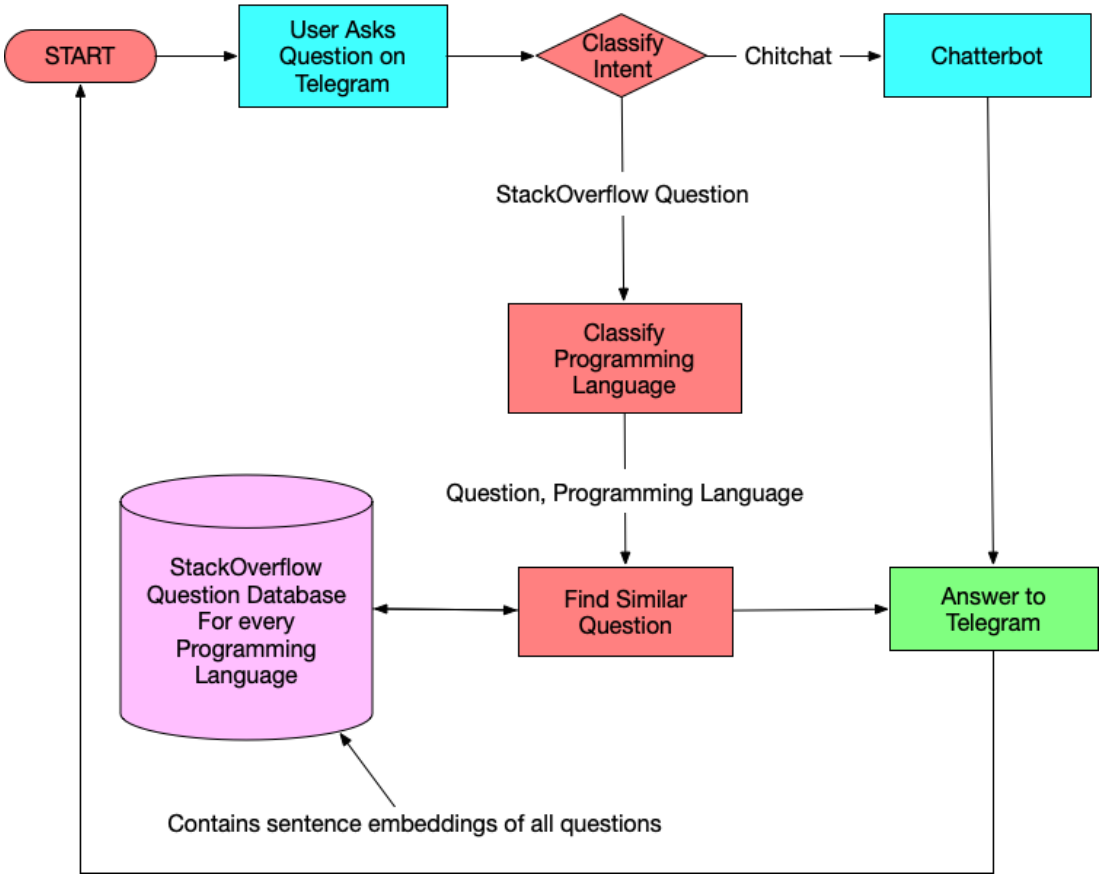
- **tagged_posts.tsv** — StackOverflow posts, tagged with one programming language (positive samples).

Resources to build our Chatbot.

- **GOOGLE COLAB** to train the model.
- **STACKOVERFLOW** dataset.
- **CHATTERBOT**, a Python library to generate automated responses to a user's input for Chitchat type questions for our Chatbot.
- **TELEGRAM** to instantiate the bot, by creating a Chatbot UI and connecting it to telegram app backend and run our Chatbot logic.
- **AWS** to host the Chatbot.

4. DESIGN APPROACH AND DETAILS

FLOWCHART



5. SCHEDULE, TASKS AND MILESTONES

Review-1 (Feb 2021)

- Understanding of the project, the objective and tool requirements, formulation of project plan.

Review-2 (Feb-Mar 2021)

- Creation of Intent-Classifier and Programming-Language(Tag) 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.

Review-3 (Apr-May 2021)

- To store question database embeddings to get most similar question for the one the user has asked.
- To fit all the pieces in our SimpleDialogueManager Class in our Telegram Bot Handler that responds to the question the user has asked.
- Demonstration of the complete working project of the bot responding to a user's queries.

6. PROJECT DEMONSTRATION

Model Creation

1. Import Libraries
    - Import the required libraries.
2. Read the Data
    - Read the dataset files and store them as a dataframe.
3. Create training data for intent classifier

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

4. 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 with pickle to use it later in the running bot.
5. Create Programming Language classifier
    - Prepare the data for this task and split the data into training set and test set in 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.
6. Store Question database Embeddings
    - Load GoogleNews-vectors-negative300.bin, pre-trained word2vec model from google.
    - Convert every question to an embedding and store them so we don’t calculate the embeddings for the whole dataset every time.
    - Whenever user asks a stack overflow question, use cosine similarity to get the most similar question.

- For each tag create a .pkl file with two data structures, which will serve as 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 answer are stored.

7. 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 minimum distance between this vector and tag_vectors (set of vectors) to find the index of most similar post.
    - In essence, to have a function to get most similar question's post id in the dataset given we know the programming Language of the question and the question.

Telegram Setup (main.py)

- Set up a bot by talking to the BotFather in telegram and creating name/user name for the bot.
- 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 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 data for Chitchat type questions.
- get_similar_question function - given the question and the programming language of the question (tag), loads the particular tag's post_ids and post_embeddings from the .pkl file, converts the question to a vector and computes minimum distance between this vector and post_embeddings (set of vectors) to find most similar question's post id 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 and for a programming question, it finds the tag (programming language) and generates the thread (stackoverflow link) to the most similar question as a response.

7. RESULT & DISCUSSION

Results

The bot responds to a programming question with a stackoverflow link for the question asked and it 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 chit-chat 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 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).

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).

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

Discussions

We can improve on this present chatbot by increasing classifier accuracy, handling edge cases, making it respond faster, or maybe adding more logic to handle more use cases.

For a chit-chat mode I have used a pre-trained neural network engine available from ChatterBot, we can also use Seq-2-Seq models or train our own models to create such bots.

In the near future, I plan to extend my work on a large scale study such as to answer questions from all domains, i.e, open-domain question answering.

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 its implementation.

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 backend.

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 Standardisation 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] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.