BOT THE BUILDER

Team members

Daksh Verma (SID 19103065)
Priyanshu Kaushal (SID 19103067)
Kanishka Tiwari (SID 19103015)
Navdeep Kumar (SID 19103079)

Mentor
Dr. Sudesh Rani
Punjab Engineering College, Chandigarh, CSE

Problem Statement - Chatbots in websites and applications are common these days. With the evolved techniques in natural language processing and natural language understanding, chatbots are able to answer the way humans do when asked some query.

No business leader will want to set up a team to do the work that a chatbot can do in less time and more efficiently

But, to include chatbot facilities in any project, he/she shall have developers who can create customized bots to answer some set of questions or should himself know how to develop one.

Even if he does so, there will be many other tasks that need to be done correctly to have a fully functional chatbot. Here are some of them

- Need to find a way how the chatbot finds the meaning of the question asked by the user.
- Need to handle how chatbots performs human like conversations (greetings, see offs)
- Need to handle how chatbots perform on unseen questions.
- Need to manage that if two different questions have same/similar meaning then, they should have one common answer
- If you need to deploy the chatbot to different applications or environments (like iOS, Android), then you need find different ways to deploy in different environments

Proposed Solution - We as a team, came up with the idea to create a platform where users can create their own/customized chatbots. The user will provide the data with question answers from a specific set of domains and all the other tasks will be performed by our system. The following points try to bring a concrete solution to the above stated problem.

- **Use of vector embeddings** - The user provided dataset will be converted to vector embeddings. These are high dimensional numeric representations of a question and will be unique for all the different questions. These values form the core of our system as these values store the sentence's meaning in them.
- **Inbuilt trainable data for general talks** - Our system will hold the data for the general human conversations and the chatbot will be trained on them. If the user wants to provide a specific way to behave, they can provide that in the dataset itself.
- **Web Search and score/metric to return no answer** - If our system is not able to find any correct answer according to the users' dataset, then it will perform a web search and try to find the questions from the internet. If he's still unable to do so, chatbot will inform the user about its inability to answer.
- **Comparison of embeddings** - The embeddings created for the dataset represents the meaning of the sentences. If two questions are similar then they will have similar embeddings.
- **REST APIs (plug and play)** - The chatbot service will be provided through REST APIS. That means that you don't need to deploy the chatbot anywhere and can use the chatbot through a link from any kind of environment.

Tech Stack

Frontend

- *React.js*
- *Redux*
- *Bootstrap*
- *HTML*

Backend

- *Spring boot (Java)*
    - *Spring Data JPA (Hibernate)*
    - *Rest APIs (REST Controller)*
    - *Web Security (JWT token + Google Oauth 2.0)*
- *Flask (python)*
    - *Rest APIs (Flask Restful)*
    - *Flask SQLAlchemy*
- *Tensorflow and keras (for AI related tasks)*
    - *RNN (Natural Language processing)*

Database

- *MySQL (SQL database)*