

AI(AI) chatbot.

Our mission in this project is to come up with a chatbot that recognizes the feelings in the conversation partner and reply accordingly. This is considered a straightforward task for humans to recognize and acknowledge others' feelings. However it is a significant challenge for AI systems.

The data set we used for training and testing in this project is One of the available datasets that helps in recognizing the feelings and improves the conversations in chatbots is the Facebook Empathetic Dialogues, they provide a novel dataset of 25k conversations grounded in emotional situations.

Pipeline Stages:

We divided our work into three different pipeline parts as Follows:

- **Data Part:**

this part divided into two tasks:

- **Data Cleaning:**

We noticed that data was not structured properly, and the comma separator of the CSV file was making a lot of shifting in the rows, in addition to the shifting of the Arabic translation with the English utterances and prompt. We solved this issue manually by relocating every translated utterance and prompt with its corresponding English sentences and dropping the rows that make the problem . We apply the cleaning task to both the training and testing data.

- **Data Preprocessing:**

In this task, we changed the structure of the data to be [features, labels] structures to be able to fine-tune our model on it. Our features will be the input context which will accumulate 3 utterances **(to maintain the history of the conversation)** before the answer and the labels will be the emotion of the context and the response. In the end, we loaded the data into the Dataset class after applying 'arabert' preprocessing on context and emotion columns which will do all the preprocessing tasks on the Arabic sentences including the segmentation task.

● Training Part:

this part divided into five tasks:

- Prepare the ***train/dev/test*** sets:

In this task, we split the testing data into dev and test data and shuffle all the data with Random Seed equal to 42. Then we map all the data into one length numeric vector equal to 160 to be able to train the model.

- Load the ***PreTrained*** “ARABERT” model:

Using the concept of transfer learning it will be very helpful to start with a model that is trained on a similar task to the one we are working on so we used the “ARABERT” model as a start for this task, having trying different pretrained models this one proved to be the most promising as it provided the most blue score and the best accuracy and provided outstanding results so we choose to continue with as a pretrained model.

- Setting up the model hyperparameters to make the model ready for training:

In this task, we configure the needed hyperparameters of the model such as batch size, pad_token_id, max_length, and vocab_size and we build the compute matrix for the trainer and initialize the trainer.

- Fine-tuning the model:

In this task all the magic happened here. We let the model learn from our data and generalize a response behavior.

- Evaluate the model:

At the end, we evaluate the model by calculating the BELU score and loss for the model against our testing data.

- Last but not least, we built an interactive GUI for testing our model and to make it easy for anyone to use the application.

Other approaches:

Chatterbot was used to create the chatbot and train it on the provided dataset after converting it to a list of conversations.

Sentiment analysis was done using TextBlob, however, the accuracy was too bad because of the dataset. The accuracy remained bad even after mapping all empathies to only 5 main empathies or even to only two, positive and negative.

Although the rule-based chatbot created using Chatterbot performs pretty well on given questions, it wasn't possible to do sentiment analysis using TextBlob, and other ML models weren't available for direct use, thus the approach was not as successful as the ARABERT model.

Please refer to the notebook for more details on the code.