Mov Bee Chatbot

Data:

I have downloaded the dataset from the website : http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html . The dataset contains metadata about characters, movie lines, movie titles, raw urls, movie conversation.

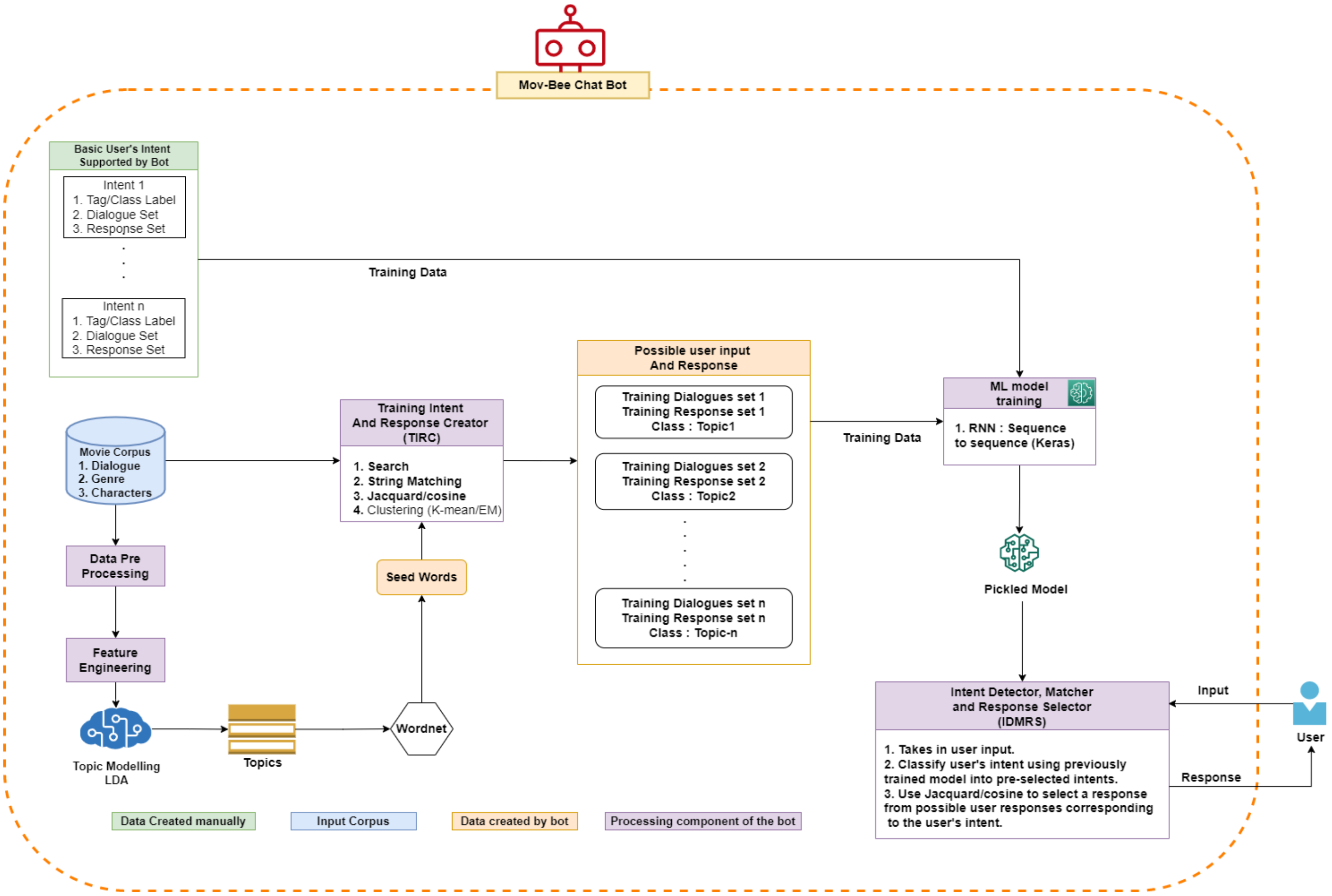
I am specifically using *movie_lines.txt* which consists of 304713 rows × 5 columns of the conversation between the characters where each line represents a dialogue in a particular movie. This dataset did not have any missing values. The source of my dataset is the dialogues and the target is the words generated using topic modelling(discussed below).

The objective of my chatbot is a question answer model such that given a user input, the chatbot will respond in terms of movie dialogue according to the question. For example, if the user asks a question like “give me your hand please” the chatbot will respond “All right, I'll work on it, I'll, do some research. But in the meantime, don't get crazy.”

The motivation behind this model is inspired from the movie “Bumblebee” directed by Travis Knight. The synopsis of the movie is that Bumblebee, Autobot(transformer) who can convert into a yellow Volkswagen beetle, is seeking refuge on the Earth from the battle whose current state is scarred and broken. Charlie, protagonist of the movie, discovers him and tries fix him. However, Bumblebee can only connect with her via tuning the radio channel. In the similar fashion, the name of my Chatbot is “Mov Bee”, who can only make conversation using the movie dialogues that is present in its database.

The chatbot data structure is such that the intent is generated using topic modelling and passing input as movie dialogue corpus. Using the intent, I am generating seed words that are the related/similar words of the intents. Based on the seed words, which is a list of words, I am applying Jaccard Similarity between seed words and the movie dialogues to find the pattern. “Pattern” is nothing but the movie lines that have a Jaccard score greater than 0.1. Hence, the “response” is the next dialogue of the pattern selected.

Design Architecture:



Data Preprocessing:

Once the data is loaded, the movie dialogues needed to be preprocessed. I am tokenizing the movie lines and applying lemmatizer and selecting only the words greater than 3 characters. I have also removed any special characters and stopwords. The function *text_wrangling* normalizes the movie dialogue.

After normalizing the dialogues, I have applied feature engineering to extract the features from the movie dialogues. From the scikit-learn library, I have used *CountVectorizer* to implement the feature engineering. The function *featureEngineering* performs feature engineering and returns *cv_features* a ndarray of shape --.In the CountVectorizer, I had used min_df = 2, which selects the words that have a frequency of occurrence greater than 2, and max_df=0.5, which ignores the words that have occurred more than 50% of the time.

The cv_features are then passed as input to the function *topic_model*, where I have implemented Latent Dirichlet Allocation from the library scikit-learn. The *LatentDirichletAllocation* function produces 100 topics from the movie dialogue. The topics generated from this function are called “intents”. As described above, I am generating seed words using the *wordnet.synsets* from the nltk library. Thus, forming the intents described in the example above.

Data Training:

The model used for the chatbot is simple rule-based model. The intents have been trained using sequence to sequence model in keras where training data is movie dialogues and labels are intents. For detail, refer to the diagram. I trained the model with 100 epochs and gained an accuracy of 45%.

As discussed above regarding the intent creation, I have also created an intent for greetings manually and added it to the intent as shown in the diagram. An example of the intent is as below:

"**christmas**": {

"**tag**": "christmas",

"**seedWords**": ["Noel","Christmas Day", "Christmastime","Christmastide", "Yuletide","Yule","Christmas","christmas"],

"**pattern**": {"122179": "Merry Christmas eve.","122262": "And how was your Christmas?","464246": "One more day, baby. One more day to Christmas.","463905": "Christmas... Eve..."},

"**response**": {"122179": "Not if you work for FedEx.","122262": "Terrific. Yours?","464246": "I've been doing good, though.","463905": "You know where the guards are. You know how to get in and out. You know where the money is. We're taking down that casino, convict. You're the guy gonna tell us how."}

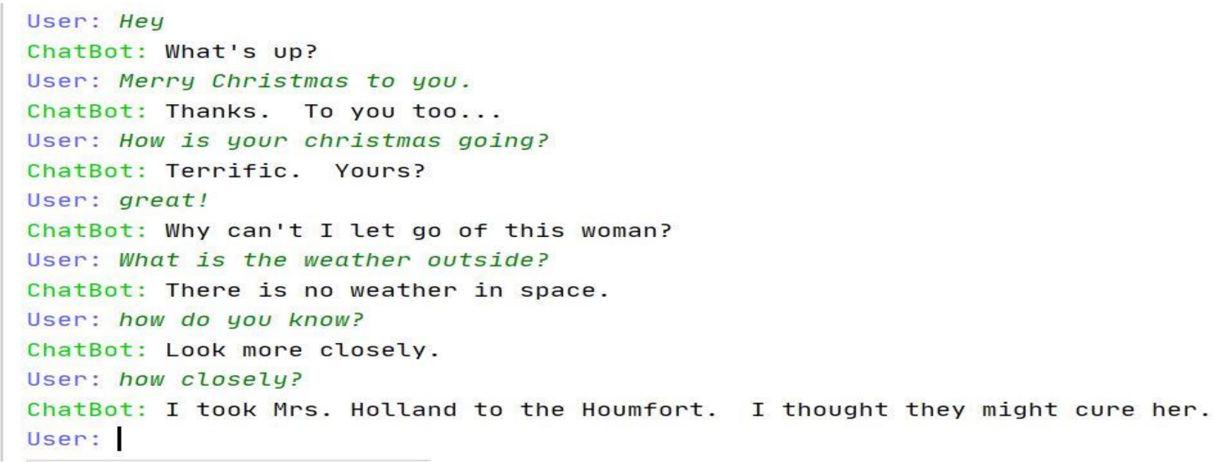
(Here, the term Tag is the intent and “Christmas” is the topic generated using LDA. Thus, the topic name is also intent.)

So, when the user gives an input, it classifies the user intent based on the training model and then applied Jaccard Similarity to select a response from the possible user responses corresponding to the user’s intent.

Evaluation:

The chatbot is working as expected, however since the accuracy is only 45% it does not perform very well. For future work, I can implement cosine similarity on the text data which may yield better results. On the next page I have attached a sample of the Chatbot responses. I have submitted the code on github for your reference. The link to the code is :

https://github.iu.edu/NLP/Mov-Bee-Chat-Bot/blob/main/Final%20Project.ipynb



```
User: Hey
ChatBot: What's up?
User: Merry Christmas to you.
ChatBot: Thanks.  To you too...
User: How is your christmas going?
ChatBot: Terrific.  Yours?
User: great!
ChatBot: Why can't I let go of this woman?
User: What is the weather outside?
ChatBot: There is no weather in space.
User: how do you know?
ChatBot: Look more closely.
User: how closely?
ChatBot: I took Mrs. Holland to the Houmfort.  I thought they might cure her.
User: 
```

References:

1. https://online.datasciencedojo.com/blogs/building-a-rule-based-chatbot-in-python
2. https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281
3. https://github.com/amilavm/Chatbot_Keras/blob/main/train.ipynb