Conversational Chat Bot

Natural Language Processing – BUAN 6V99

Mitali Bharali

Masters, Class of 2020 – Business Analytics
Univeristy of Texas at Dallas

I. Introduction

The growth of Natural Language Processing is exponential in today's time and its implementation is being recognized in almost every industry. A recent study by the Kelsey Group reports that increasing numbers of companies are investing in voice or speech recognition and processing technologies at an alarming rate to save money by replacing operators and to improve service to their customers. Reading text from prints, Speech Recognition to Language Translation, and future adaptations not yet discerned. Given if we can acquire the full potential of what NLP is capable of, it can snowball into technology of its own, much like other engineering fields. In information retrieval sense, it would mean harnessing an astronomical amount of data if exploited to full capacity and hence tuning all areas of technical expertise.

Tapping this area meant starting with the basics from preprocessing technics to vectorizing words to apply similarity and making to basic conversation. This report illustrates these aspects of NLP to create a conversational chatbot implemented through an GUI application.

II. Dataset Description

There are three datasets used for this project:

i. Dialogue.tsv – that classifies non-technical, natural English dialogues.
ii. Stackoverflow.tsv – that classifies data into programming topic related questions, each of which belong to one class of programming language.
iii. Starspace.tsv – that maps to the corresponding answers to questions, used in creating word embeddings.

	text	tag
82925	Donna, you are a muffin.	dialogue
48774	He was here last night till about two o'clock....	dialogue
55394	All right, then make an appointment with her s...	dialogue
90806	Hey, what is this-an interview? We're supposed...	dialogue
107758	Yeah. He's just a friend of mine I was trying ...	dialogue

Fig1. Dialogue.tsv

	post_id	title	tag
9		Calculate age in C#	c#
16		Filling a DataSet or DataTable from a LINQ que...	c#
39		Reliable timer in a console application	c#
42		Best way to allow plugins for a PHP application	php
59		How do I get a distinct, ordered list of names...	c#

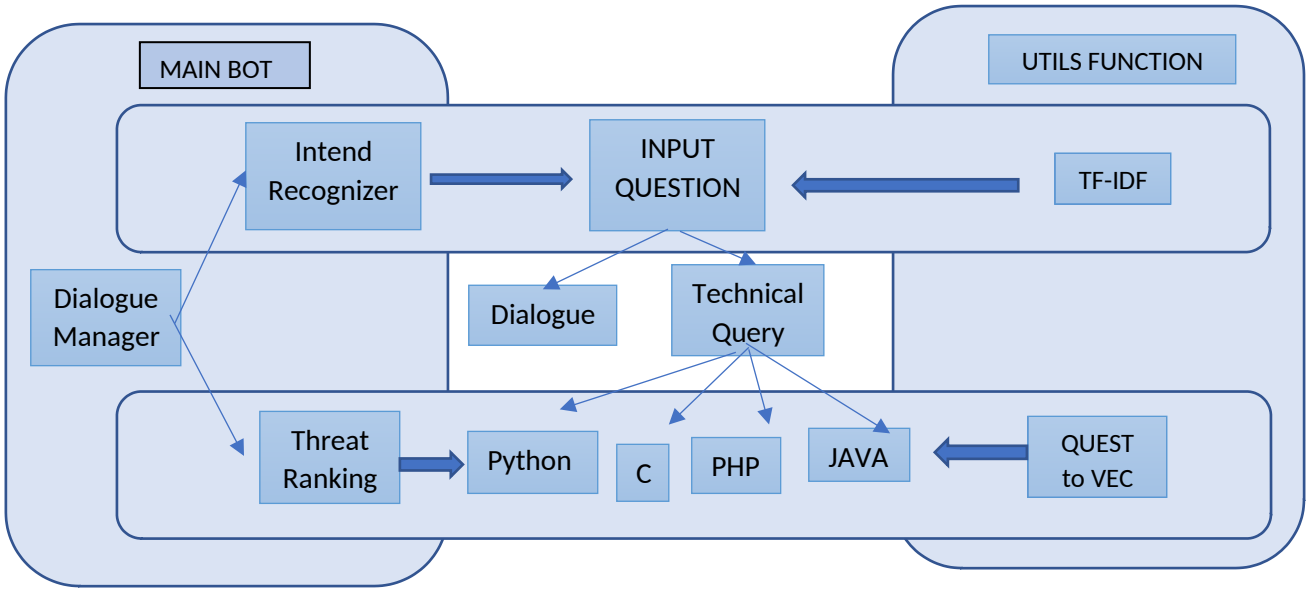
Fig2. Stackoverflow.tsv

	post_id	title	tag
2168983	43837842	Efficient Algorithm to compose valid expressio...	python
1084095	15747223	Why does this basic thread program fail with C...	c_cpp
1049020	15189594	Link to scroll to top not working	javascript
200466	3273927	Is it possible to implement ping on windows ph...	c#
1200249	17684551	GLSL normal mapping issue	c_cpp

Fig3. Starspace.tsv

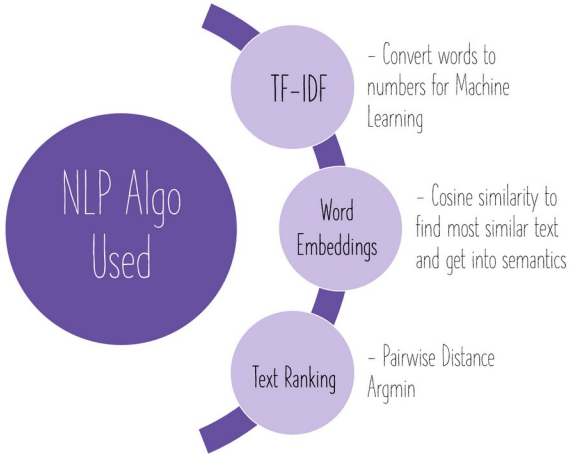
Each of these datasets has 3 millions observations, hence its recommended to use cloud interface for data retrieval and modeling. There are no missing values in any of the dataset.

III. Architecture



The code is broadly divided into two parts, one that has the utility functions →Utils and the other that has the Main Bot Handler → Dialogue Manager. Utils has text_prepare function that preprocesses, td-idf_features that vectorizes and Question-to-Vec function that creates embeddings, each called as required. The Main Bot has two broad categories – Intend Recognizer that classifies the input question and, Threat Ranker that gets the best answer for the question vector.

IV. Natural Language Processing



i. Preprocessing

The preprocessing starts with basics of NLP. One function *text_prepare* is used to perform all the below preprocessing:

a) Lemmatization
b) Stemming
c) Stopwords removal

Lemmatization normally aims to remove inflectional endings only and to return the base or dictionary form of a word, known as the lemma. While stemming often includes the removal of derivational affixes. Stopwords would remove frequently used English auxiliary words which contribute no specific meaning to the sentence, usually used for grammatical purpose – a, the, an, or, so…

ii. TF-IDF

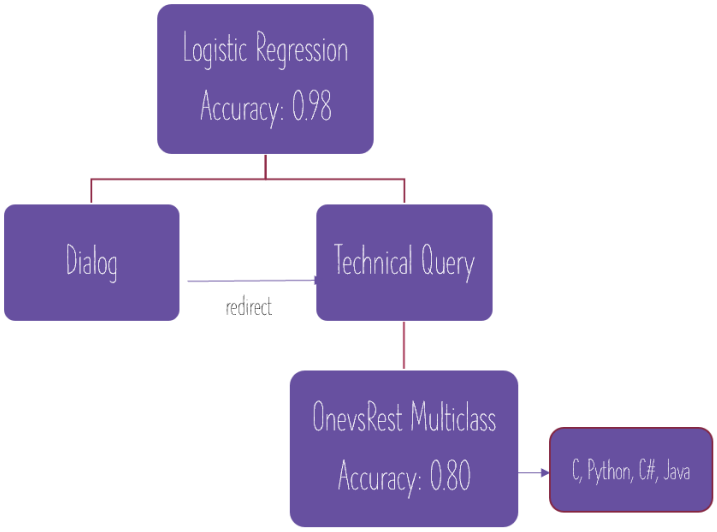
This is another form of the preprocessing making it more readable, extracting features for the machine learning algorithm to work on. Tfidf_features is made and once called; data is dumped in the pickle to later use it in the bot.

iii. Word Embeddings

A word embedding is a learned representation for text where words that have the same meaning have a similar representation. The vectorized data is measured for proximity with similarity function in the vector space which translates into the similarity in the meaning of the words in textual language. This is used to train the bot to find the similarity between the question vectors (Stackoverflow data) and the existing threat- most likely answers data (Starspace data). Cosine similarity is used here. This helps in ranking model – where the most likely answer from the given programming topic searched for the input question. The evaluation matric used here is pariwise distance argmin.

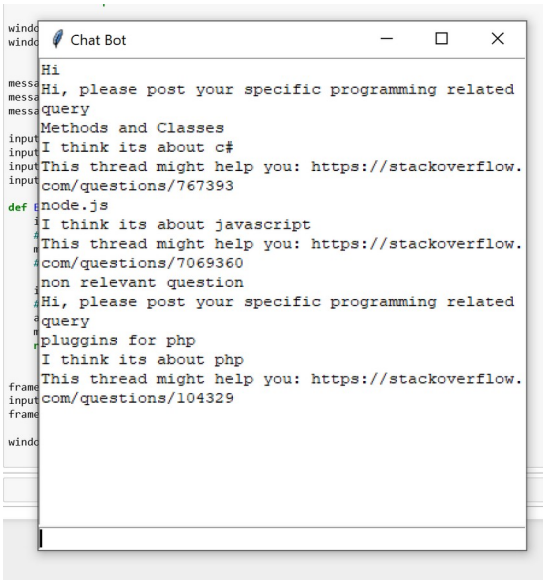
V. Machine Learning

There are majorly two machine learning used for training the dataset – Logistic Regression and Multiclass OnevsRest. The logistic regression is used in the Indent Recognizer section to classify the question vector into Dialogue or Technical Query. For this, a concatenated dataset of Dialogues.tsv and Stackoverflow.tsv is used. Multiclass OnevsRest is used in labelling the tags against the right query – Python tags to python-based queries and so on. A word embedding is formed from starspace.tsv data that is used as answers to the question vectors, that is used in Threat Ranker. The ideal approach to deciding the algorithm would be to start with simplest models with least parameters and extend to advance deep learning models like RNN and LSTM. While in this project, accuracy and performance of simple models were high, deep models would have overfitted the dataset and performed poorly on new dataset.



VI. Output

The output of the algorithm looks as below. The main_bot function logic calls the dialogue manager function against the input data. The dialogue manager tests it for intend and once recognized as technical query, predicts the onevsrest model and uses the threat ranker to find the optimum solution.



VII. Recommendation

Chat bot is right now the most widely used application of NLP, finding its implementation in customer experience improvement process. Any online website of a company can help redirect lost customer to the right section, hence reducing the TAT time, checkout time and hence increasing conversion rate. Effective chat bots can help navigate customer into right products, which could be useful in giants like Amazon, Walmart. Recommendation systems can also be

greatly benefited from this. In totality, it improves customer experience and helps in long term relationship management.

VIII. Conclusion

This project is a small illustration of how to implement a simple Chat Bot to a website for a customer relationship management. There are several areas of improvement here, like tokenizing with a software API, using further more data through cloud platforms like Big Query and AWS, modeling with deep learning algorithms and creating further more dialogue blocks to refine the prediction.