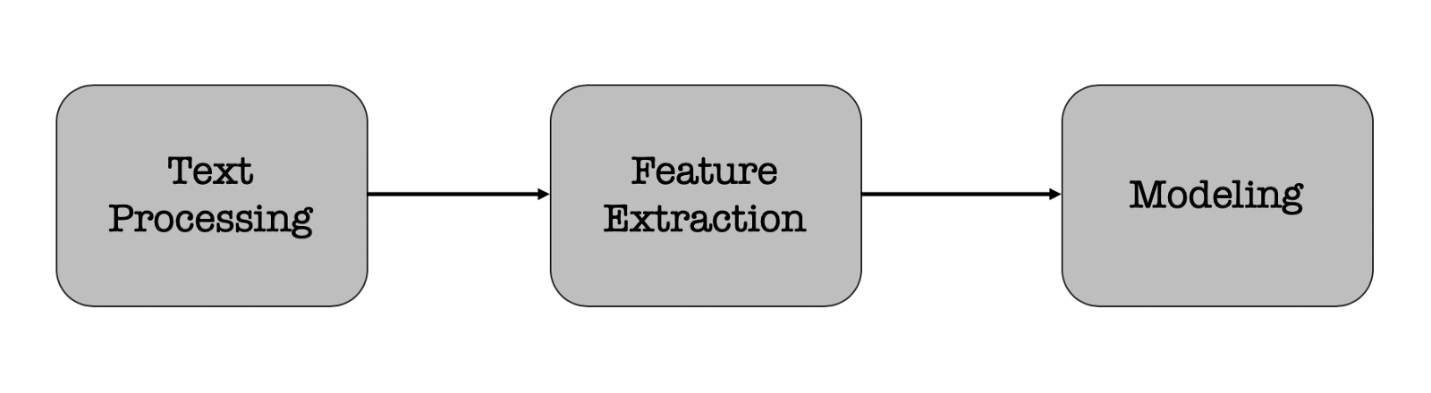
Scott Berry – Scorberr

NLP Chatbot Final Project - Palantír

https://github.com/berryscottr/palantir

The data used is the entire text of The Lord of the Rings book trilogy, this text was hosted in a Github repository. This data was saved in the project folder as a simple text file. This file is 2.56 MB in size with 746,767 words.

This chatbot’s purpose is to discuss the events of Middle Earth from the book series The Lord of the Rings with the user. The chatbot is designed to emulate a “seeing stone” from the book series and as such the user can discuss anything about the fictional world with the seeing stone. This is an application of the chatbot that is purely for entertainment due to the dataset provided, but with an encyclopedia text there would be far more practical purpose.

The chatbot is trained by using the standard order of Natural Language Processing pipeline: 

First the complete Lord of the Rings text is entered into the notebook in full lower case. Next, the text is tokenized into sentence tokens and word tokens. Additionally, when the chatbot is conversing, it uses a lemmatization function from the NLTK library to convert user input to tokens. This function also assigns sentiment scores to each token. Dependency issues with Chatterbot’s functions prevented ability to use the Chatterbot module so NLTK was more heavily utilized.

In addition to the sentiment scoring, feature extraction is accomplished by combining three NLP techniques: bag of words, TF-IDF, and cosine similarity. First, the bag of words is created to gather counts of each words presence in the text. Next, TF-IDF is used to assign weights to each word so that relevant words in each phrase carry the most weight. Finally, cosine similarity is used to assign temporary relative values between words.

Bag of Words model works by processing all of the text input from the dataset and assigning them to a matrix with unique words and the count of how frequently they appear in the data. With this, a phrase such as “I like ice cream” would be represented by a vector: [543, 47, 6, 2]. Bag of Words has no contextual information regarding where words were located and which words were around them. However, this representation shows that the low count words in the vector are likely the most important words in the phrase.

The TF-IDF (Term Frequency – Inverse Document Frequency) Vectorizer works well in conjunction with Bag of Words as the TF-IDF Vectorizer converts these counts of each word to weighted values ranging from 0 to 1. Now, the phrase “I like ice cream” would have a vector representation as: [0.03, 0.12, 0.87, 0.98]. This achieves converting the relative frequency of the Bag of Words to an absolute measure that can be applied to text analysis.

The Cosine Similarity model is essential for the chatbot response as this model is how the chatbot determines which response to use from its dataset. The vector representation of the input text is compared across the corpus of the dataset and this model selects the most similar set of text based of the cosine similarity equation: Text

Description automatically generated. Once the most relevant phrase is found, the chatbot responds to the user. What this approach suggests is that the chatbot would respond with a phrase about “ice cream”, “cream”, or “ice”. The chatbot would not respond with a phrase such as “I like \_\_\_\_”.

Response generation is accomplished by the three aforementioned NLP models. In addition to those, greeting phrases are searched for on input with the chatbot returning a random greeting response. Also, the chatbot seeks for a goodbye phrase for conversation to end.

The response generation model function is prompted by user input. The model adds the user input to the dataset (LOTR text) and normalizes the text via the lemmatization function and removal of stop words. The user input is then compared to the existing data and cosine similarity is used to return relevant entries from the text corpus to make conversation.

The chat-orb Palantír is clearly a far from perfect chatbot. The shortcomings of the chatbot returning just relevant phrases from the text rather than truly understanding the user becomes clear when the bot returns non-relevant responses that just share similar words and concepts. This method would likely work better in a dataset that is an informational book rather than a fictional story such as Lord of the Rings. Further, when referring to a previous user input it was clear that the chatbot had no memory aspect. With these shortcomings noted, every response could be interpreted as a human-like response and while responses were not always relevant, they could still be made sense of.

Text

Description automatically generated

Resources:

https://www.datacamp.com/community/tutorials/building-a-chatbot-using-chatterbot