ML Project Proposal : Retrieval-based philosopher chatbot

Iris Bisson and Gregoire Trinh Ngo

Project idea:

Because we are both interested in philosophy, we initially wanted to make a chatbot that would speak back to the user in the tone and with the ideas of a specified philosopher. However, rather than generate text, we then thought of retrieving quotes that match what the user is asking about. Our project will hopefully feel like talking to our favourite philosophers, but with real quotes from their body of work.

Dataset:

Our main dataset includes several hundred quotes from a set of 9 famous philosophers, organized by author.

If we need more data for certain authors, we found

- Bibliography of Nietzsche
- Some from Freud, Jane Austen, Mark Twain and Maya Angelou

If we find nothing from the specified author, we can look at more general datasets of quotes:

- quotes with associated author and popularity (value from 0 to 1)
- another huge dataset of quotes with the associated author and the category (love, life, philosophy, motivation, family…)
- more specifically, some stoic quotes

Methodology:

1. Data pre-processing

First, we will get rid of all useless information like punctuation and capital letters (get everything in lowercase). We will then create TF-IDF vectors : each quote becomes a vector composed of scores for words of interest. This score takes into account *Term Frequency* TF (how many times a word appears in the quote) and *Inverse Document Frequency* IDF (whether the word is very common and present throughout the dataset, in which case it's probably useless, or if it shows up in only certain quotes, in which case it is probably talking about something specific).

- Need to talk to TPM to see if this is a good approach and suitable for our project

2. Machine learning model

We will receive the user input and also encode it into a TF-IDF vector. Then, we will find the use of cosine similarity to determine if each quote in the database is in the same “direction” as our user input, if it has no similarity or if it's completely opposite. We will choose the most similar/relevant quote and display it

3. Evaluation metric

Some things to check for how well our unsupervised model is doing:

- Precision@k - are the first k retrieved quotes relevant to the prompt
- Other methods, we need to ask TPM

Application:

We want to create a nice looking web app where

- Input: user's text prompt, like asking a philosophy question or talking about their experience/opinion. It shouldn't be something random like “what's 2+2”, in which case we'll probably display something like “Sorry, two thousand years of philosophy still hasn't found an answer to your question!". The user will also have specified beforehand (for example on the side toolbar) what philosopher they want to talk to.

- Output : A quote from the specified philosopher that best matches the user's input. Hopefully, the response is relevant and answers the question. If we can't associate anything the philosopher said with the user's prompt, we'll pull something from the general quotes dataset, maybe presenting it like "My good friend [...] told me that [...]"

Note: If this project does not end up being feasible (not complex, not long), we would be interested in a supervised classification model where we study quotes from a list of philosophers, and then we give the model an unseen quote from such a philosopher and it says who wrote it. We would then be able to extend that to the user interface, and end up telling the user to go read a certain book/author.