Swag Das
Zaki Kidane
Brandon Roemer

Project Proposal (Midterm Report 2) - AdviceAI

Summary

The end goal is to design an AI that produces advice relevant to a user's prompt for advice and their demographics. Our first task was to collect a significant amount of advice. So far we have one advice dataset with 1700 snippets of potentially relevant advice. We tried using Text Distance algorithms such as Jaso-Winkler, MLIPNS, Jaccard, Cosine Similarity, Bag, and even MRA. These were underwhelming in practice. We are currently using a TensorFlow Universal Text Encoder model for encoding questions and answers on the dataset and based on our manual testing, it is quite promising. Our current model ranks the entire dataset by relevance to the provided prompt and demographics, along with providing a similarity score for each. We are considering using this to fine-tune our dataset and increase performance.

A website with a user-interface prompting a user to ask a question and that generates an advice has also been added. Having shared the link on social media, about 600 questions have been collected in our database. Those questions will be used to test and compare the responses from different advice corpus as well as different models that we try out. In addition, we found advice varied a lot by age so we added an age demographic to our model so it can better represent individuals. Some issues that we need to resolve are that we are still dealing with problems related to load times and are writing a python API to handle the requests for our website as the python model runs much faster than the javascript one. We also found hypertuning the Universal Sentence Encoder isn't easy without enough data but now with the extra data we have collected that is specific to our model, we might be able to fine tune it to questions relevant to our use case.

Techniques

Our current model is Tensorflow's Universal Sentence Encoder in Question and Answer mode. This essentially vectorizes questions and answers, or in our case, prompts and advice. The results with the testing we have done so far are good, and are even resistant to grammar and spelling issues. However, we still have yet to improve it in order to make sure the advice is relevant in almost all cases. To calculate the relevance we take the dot or inner product of the vectorized prompt and each vectorized answer. This is the "relevance score", and we can just sort our dataset by this score to find the most relevant advice. Our future plans are mostly related to improving this current pipeline.

Goals

From here we want to increase accuracy and decrease load times. First and foremost we will continue to enlarge our advice dataset as this directly leads to increased accuracy. The AI can't provide advice it doesn't have. We will also enlarge and diversify our questions that will be used to test the algorithm. One more complex method we are working on is comparing subsections of our advice output (generated by the model) in accordance with their average relevance to a given corpus of questions. We expect this to improve accuracy by having a more optimized body of advice. Another idea is to try distilling the Universal Sentence Encoder in a method to HuggingFace's DistillBert. This is plausible because our prompts and advice only take up a small subspace of the full model. This would significantly decrease load times as it currently takes ~4 minutes to load the model. We also plan on doing additional research into the Tensorflow Model to find any parameters we can tweak to meet our needs. Our learning

objectives are to explore and familiarize ourselves with Natural Language Processing packages that perform well in sentence encoding and sentiment analysis. We would also like to learn how to manipulate data so that our models work better.