Leveraging OpenAI's Ada Embedding Model for Zero-Shot Classification at TREC 2023 Clinical Trials

Luke Richmond, Graduate Student, Luke.Richmond@marquette.edu
Priya Deshpande, PhD, Priya.Deshpande@marquette.edu
DICDSL Lab, EECE Department, Marquette University, Milwaukee, WI

Abstract:

This paper briefly discusses our submission to the TREC 2023 Clinical Records Track. The track challenged participants to match patient details with medical research trials based on whether the patients were believed to be a good fit. Our method utilized OpenAI's Ada model, a market solution for finding similarity based on given strings. By using a prebuilt solution, we sought to produce a solution that gave results better than random guessing with both low design cost and low overall monetary cost.

Introduction:

Recently, there has been an uptick in patients searching for clinical trials that apply to them.[1] To tackle this the 2023 Clinical Trial challenges participants to take 40 patient descriptions and find the trials that best match their specific conditions. With around 450,000 trials to consider, the large amount of de-identified data makes even simple methods time consuming when applied to the whole dataset.

OpenAI created ChatGPT, which can be used to provide human-like responses to natural language prompts. Originally, we were interested in using ChatGPT to compare similarities between patient and trial descriptions. However, OpenAI's Ada model provides more direct comparisons between strings than ChatGPT.[2] Furthermore, the Ada model can perform zero-shot classification with no training for the intended specific use case.

Methodology:

Data for this competition was provided as xml files with neatly categorized text fields. Although xml files make data parsing easier, they are not expected input for the Ada model. To avoid confounding similarity in xml formatting and to reduce the size of the files, we converted the xml files to a more human readable format where text fields were placed in txt files separated by new lines. Furthermore, certain categories were disregarded if they were considered too irrelevant, with the idea being that similarity in irrelevant categories would confound later similarity calculations. For trials, the included categories were: “Brief Title”, “Brief Summary”, “Detailed Description”, “Condition”, and “Eligibility”. Finally, some resulting text files were still too large to feed into the Ada model. Any file too large was simply disregarded.

The Ada model works by assigning a cosine embedding to each text file input. To create the embeddings, the text files must be sent to the OpenAI's servers through OpenAPI. There is an associated monetary cost based on the length of the file for every embedding created. However, once the embeddings are created, they are not dependent on the rest of the dataset, so adding new trials or patient data to this collection would only mean an associated cost for processing the new data, not reprocessing old data.

Finally, after the embeddings are calculated, the k-nearest trial data neighbors of each patient data were taken to find the applicable. This k-nearest neighbors' calculation is not performed on OpenAI's servers and only has a time cost associated with it. As there were 40 patient data files and the max submission size was 1000 lines, we submitted the closest 25 trials found for each patient. For a similarity metric, the maximum and minimum distance between any patient data and trial were found. The minimum distance was taken to be 1 on our similarity scale and the maximum was taken to be 0. All other distances were linearly scaled between these two.

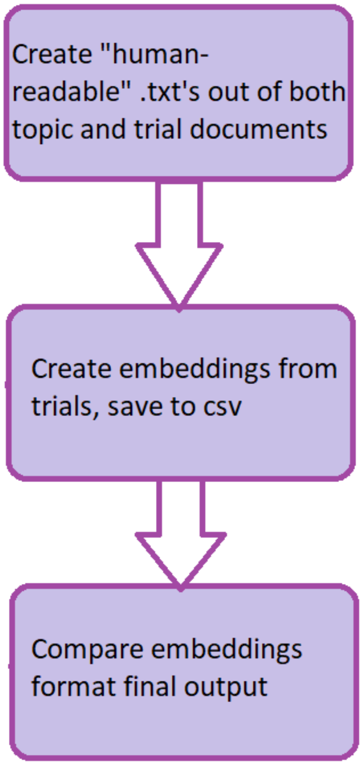
Create "human-readable" .txt's out of both topic and trial documents
Create embeddings from trials, save to csv
Compare embeddings format final output

Figure 1. A block diagram summarizing the flow of our method

Results:

Upon first review, many of the trials did seem applicable to the topics they were matched with. However, one notable drawback to the system is that embedding similarity does not consider negatives. Immediately, it was noticeable that patients with a "condition" were considered similar to trials that specifically listed "no condition".

Overall, the results can best be analyzed through the numerical results of the trec_eval output. In total, there were 22,366 trials considered related to at least one patient description. Of our 1000 submitted trials, only 417 were considered a match. Our run had an overall mean average precision (map) of 0.0228. As map is a normalized metric, our result seems poor compared to possible maximum scores. While these results are better than random guessing, they are not viable for a competitive matching method.

Conclusions:

Overall, our system's inferior performance is lacking compared with the more tailored solutions crafted by other groups for this competition. However, the ease of creation combined with the total cost being less than 50 dollars for usage of the model lends some appeal to the use of similar systems in the future. Preprocessing rules considering negatives could likely push these results even further. Furthermore, OpenAI has higher quality models that would likely also improve results. Using a higher quality model would not see a dramatic increase in cost especially when embeddings only need to be run once for each file. For small scale operations that do not have the resources to craft a fully handmade solution, leveraging a market solution may be a reasonable option to outperform random guessing.

References:

[1] "2023 Clinical Trials Track", trec-cds.org. https://www.trec-cds.org/2023.html (accessed Feb. 23, 2023)

[2] Neelakantan, Arvind, et al. "Text and code embeddings by contrastive pre-training." *arXiv preprint arXiv:2201.10005* (2022).