Charles Di Renzo
11/08/23

Fetch Project Overview

Develop a search tool that returns a list of relevant offers based on a user's query, which could be a category, brand, or retailer. The tool also provides a similarity score for each offer, indicating its relevance to the query.

Instructions

I chose to use GoogleColab and write my code in an ipynb file. You can run it locally if you have all the appropriate packages installed, but I used GoogleColab for simplicity. There isn't much to it but upload the files and run the cells. Change the 9th line in cell 10 based on what you're looking for. The result should be the top 5 closest matching items and a similarity score for each item.

Methodology

I approached this project using the methodologies that I was taught in graduate school in my 'Prescriptive Analytics' class which focused a bit more on operations research methods. In our final assignments we built a movie recommender that would take a movie and recommend a similar movie based on a score. To do this we:

- Cleaned the data by performing text normalization, putting everything into lowercase and removing punctuation.
- Cleaned the data by looking for special cases such as the lists in the brand_Categoories.csv
- Performed Tokenization and Lemmatization along with removing any potential stopwords.
- Using TFIDF for Vectorization which should numerically represent words in the datasets.
- Similarity is calculated with cosine similarity. Offers are ranked by similarity scores on a scale of 0 to 1 with 0 being no similarity to 1 being perfect similarity.

Thoughts and Comments

This is an older method of building recommendation systems, but that means that it's tried and true. I have a tendency to stick with the simplest tool that will get the job done. The reason for that is that the simplest tools that get the job are to implement, troubleshoot, and build additional functionality into. My experience has taught me that dealing with the high turnover of the tech sector and the need for reliability means that using the simplest tools is usually the most cost effective and reliable.

Alternatives

I wanted to try something a bit more special for this project and see if I could successfully implement it. Recently Josh Starmer of the 'Statquest' youtube channel came out with a video explaining Word Embedding with PyTorch and Lightning. I wanted to see if I could use word

embeddings with nn.Linear to create a more accurate recommendation system. Given that these systems like to assign values and weights to words I thought that I could use those values and weights to create a more accurate recommendation system. I was unable to get it to function as desired, so I decided to go with the easier to implement and tried and true recommendation system described above.

Conclusion

Even if I am not selected for this position, I'm happy that I was considered. I would appreciate constructive feedback on this project as NLP is something that I find interesting and I'd love to learn more about and improve further. Thank you.