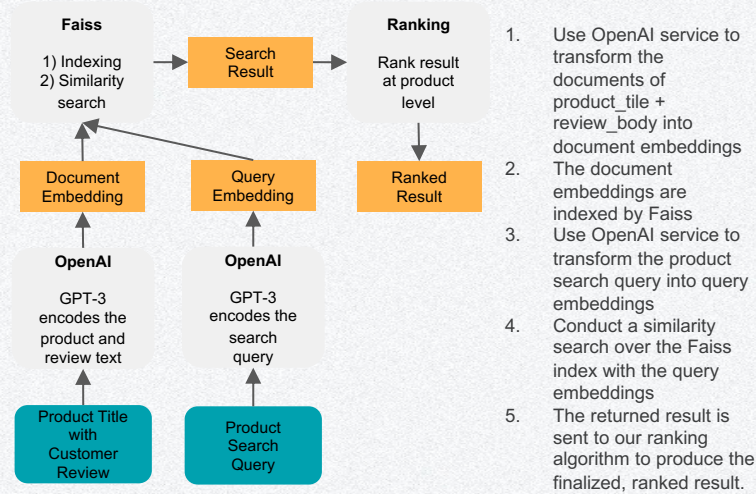# CUSTOMER REVIEW DRIVEN PRODUCT SEARCH ENGINE

Elaine Chen, Gen Ho, Varshini Rana (8-14-2023)

01 Introduction

When customers go online to purchase from e-commerce websites, product reviews are a vital source of information to reference as the target products cannot be physically touched. However, reading through all the reviews and comparing across different products could be time consuming and challenging.

In this project, we have built a product search engine which uses product descriptions and customer reviews as the content for search. This provides a richer context for customers to find their targets on e-commerce web sites. We have also introduced our ranking algorithms for calculating the product level similarity scores which beat the baseline method using average.

02 Data Source

We used the Amazon Customer Review dataset for our application. This dataset contains information of Amazon products and corresponding reviews from 1995 to 2015. There are 37 product categories where we have chosen the Apparel product group. We have also limited our scope to 2015.

https://github.com/gen-exody/nes

https://nescapstone.streamlit.app

03 Solution

Faiss
1) Indexing
2) Similarity search

Search Result

Ranking
Rank result at product level

Document Embedding

Query Embedding

Ranked Result

OpenAI
GPT-3 encodes the product and review text

OpenAI
GPT-3 encodes the search query

Product Title with Customer Review

Product Search Query

1. Use OpenAI service to transform the documents of product_tile + review_body into document embeddings
2. The document embeddings are indexed by Faiss
3. Use OpenAI service to transform the product search query into query embeddings
4. Conduct a similarity search over the Faiss index with the query embeddings
5. The returned result is sent to our ranking algorithm to produce the finalized, ranked result.
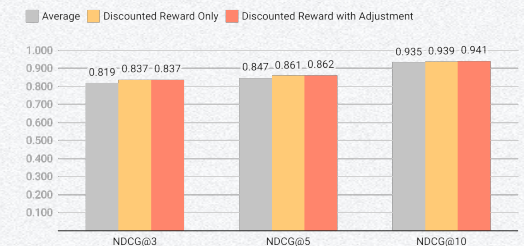
04 Ranking Algorithm

Since each search result is at product review level, we have designed and evaluated 3 methods to calculate the product level similarity scores.

1. Average
   - Serves as baseline
2. Discounted Reward
   - **Product Similarity Score** $= \sum_{i=1}^{n} S_i/2^i$, where S is the record level similarity score, n is the total number of records within a product, i is the current number or record
3. Discounted Reward with Adjustment by Opposite Query
   - **Adjusted Distance** $= D_{original} + k \times 1/D_{opposite}$, where $D_{original}$ is the cosine distance of the original query, $D_{opposite}$ is the cosine distance of the opposite query, k is a weight we set to 0.5 for our calculation

05 Opposite Query

We have introduced a novel idea by using semantic opposite queries to identify the contradicting concepts in reviews, in order to improve the accuracy of search results.

Original Query

Wrinkle free chiffon blouse, sleek style, long sleeve, slim fit, with comfortable inside layer.

Opposite Query

Wrinkled means having many creases or folds. Clumsy means lacking grace in movement or posture. Short means having little length. Bulky means large and unwieldy. Uncomfortable means causing discomfort.

06 Result

Both Discounted Reward and Discounted Reward with Adjustment by Opposite Query methods outperformed the baseline method across NDCG@3, NDCG@5 and NDCG@10.

Mean NDCG across Ranking Methods

Average | Discounted Reward Only | Discounted Reward with Adjustment

	NDCG@3	NDCG@5	NDCG@10
Average	0.819	0.847	0.935
Discounted Reward Only	0.837	0.861	0.939
Discounted Reward with Adjustment	0.837	0.862	0.941

Created with Datawrapper