SI 650 Final Project

Introduction

This project aims to develop an information retrieval system focused on retrieving pertinent information from an extensive collection of health products in response to user queries. The information retrieval system takes a user query as input and generates output documents that encompass the most relevant information based on the provided query.

In the realm of information retrieval system studies, various approaches are employed, ranging from optimizing queries and metadata to pruning indexes or implementing models with enhanced behaviors. Given the constraints of time, this project focuses on an exploration of two models—BM25 and Doc2Vec. The objective is to determine which model exhibits superior performance and assess their effectiveness across different types of queries.

In this project, I utilized BM25 as the baseline model for my information retrieval system, aiming to present the most relevant documents (in this case, health products from Amazon) for 20 predefined queries. Additionally, I introduced the Doc2Vec model as my 'proposed' approach. The findings indicate that the Doc2Vec model outperforms BM25, particularly in scenarios where queries are more straightforward. Unlike BM25, which thrives on specific and detailed queries, Doc2Vec showcases consistent performance, even with simpler queries consisting of just one word. Surprisingly, both models performs pretty well, achieved NDCG scores around 0.92.

The information retrieval system implemented in this project serves as a valuable tool for enhancing consumers' decision-making processes when it comes to purchasing products. By tailoring recommendations to individual interests and needs, the system offers a superior alternative to having no recommendations at all. Furthermore, the system contributes to companies' understanding of their products, providing valuable insights that can inform strategic decisions and improvements.

Through this project, I have gained insights into the nuanced nature of queries, realizing that their effectiveness isn't solely determined by specificity or simplicity, but rather varies based on the employed models in the system. Different models exhibit distinct performance characteristics. Additionally, identifying an 'appropriate' query for optimizing results in the information retrieval system can prove to be a nuanced and challenging task.

Data

The data used for this project is attained from Amazon, which contains customer reviews from Amazon, including 428, 781 feedbacks ranging from June 1995 to March 2013. [1]
The dataset is a text file, which contains product ID, product title, product price, user ID, profile name, review helpfulness, review score, review time (unix time), review summary and review text.

For data preprocessing, I first convert the .txt file dataset into a .csv file for better implementation. Since my goal is to create a recommender system, the products with low rating scores, those that are not up to date, and those with low click frequencies would never be considered. Thus, I choose to filter those out.

The data ranges from June 1995 to March 2013, so I first convert the time form from unix time to the normal time form, and count the data for each year. From the image below, I found that the data is not as much before 2005, and they are also not up to date at all, so I choose to filter only those data from 2005 to 2013.

After this, I filtered ‘review/score’ data. Since we only care about those products that are popular and with great reputation, I filtered out those with reviews scores lower or equal to 3, and only keep those review scores 4 and 5.

What’s more, some reviews are counted as helpful while others are counted as not helpful, so I filtered out those with zero helpful reviews. Additionally, some categories showed up with low frequency, which indicates that they are not popular choices. Thus, I also filtered out those products with lower frequency. To be specific, I only kept those products that appeared more than 10 times in the dataset. After all these steps, I obtained 106,654 feedbacks from 428,781.

Furthermore, I consolidated the review texts associated with the same product, resulting in 3230 distinct products. This adjustment was prompted by the observation that during the initial execution of the BM25 method, certain queries yielded identical products across the entire top 50 output. While varied in product ID, price, and review contents, these outputs were essentially pointing to the same product.

Following data preprocessing, I crafted a diverse set of 20 queries, ranging from simpler ones like 'travel' and 'comfortable clothes' to more intricate and specific ones such as 'Healthy snacks for weight loss' and 'Top-rated sunscreen for sensitive skin'. Subsequently, I executed the BM25 and Doc2Vec methods independently with these predefined queries, documenting the top 50 scores for each method, where for the documents, I tokenized the corresponding combined review texts and used them as the ‘document’ for each product. I manually assigned ground truth relevance scores to the retrieved products, using a scale of 1 to 5, where 5 denotes high relevance and 1 indicates low relevance. Post-annotation, I evaluated the models using Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) based on the annotated ground truth relevance scores to assess their performance.

Relate works

In the research fields, researchers explore much more on information retrieval systems, including optimizing the query, metadata, pruning index or performing models with better behaviors.

Chen Luo and his team studied query attribute recommendation at Amazon research, their research focused on how information retrieval systems contributes to the long term user experience through offline and online experiments, given the fact that product search queries are usually short, which limits the search engine's power to provide high-quality services [2]; M Zadel and his team presents an application of web service for music information retrieval to generate clusters of related musical artists based on cultural metadata, exploring emergent popular opinion about music [3]; Ellis and his team provide a flexible behavioral model for information retrieval system design seeking patterns of academic social scientists [4]; Mauldin studied retrieval performance in FERRET – a conceptual information retrieval system, focusing on the use of the Webster's seventh dictionary to increase the system's lexical coverage [5]; David and his team introduced static index pruning methods that significantly reduce the index size in information retrieval systems by investigating uniform and term-based methods that each remove selected entries from the index [6].

Methods

BM25

The baseline model for this project is Okapi BM25. BM25is a bag of words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document [7].

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

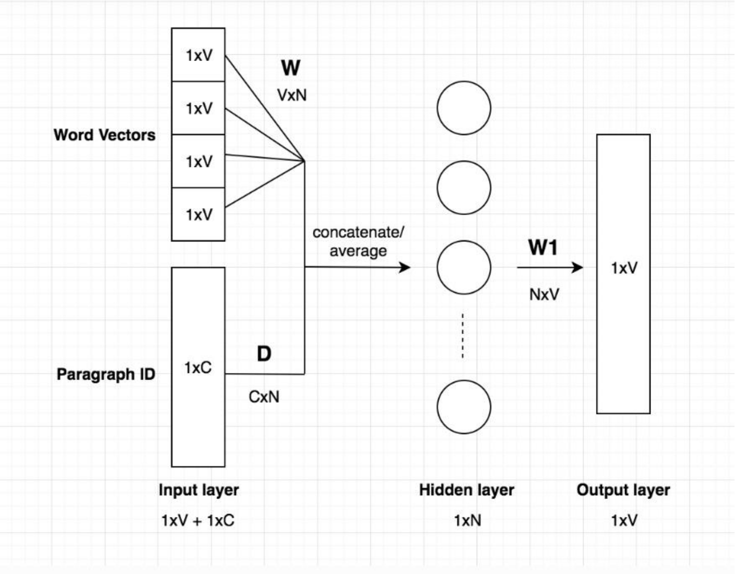
Where f is the number of times that qi occurs in the document D, |D| is the length of the docuemtn D in words, and avgdl is the average document length in the text collection, K1 ad b are parameters. The IDF refers to Inverse Document Frequency, computed as:

$$\text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

I used library rank_bm25 to process the data using BM25 model.

Doc2Vec

For comparison, I also used Doc2Vec as my 'proposed' model. Doc2Vec is a neural network-based approach that learns the distributed representation of documents. It is an unsupervised learning model that maps each document to a fixed-length vector in a high-dimensional space [8] [9].



For processing the data, I used Doc2vec, TaggedDocument from genism.models.doc2vec.

Evaluation and results

I used Doc2Vec method compared against the baseline model BM25, and it turns out that Doc2Vec performs slightly better than BM25, the MAP score of BM25 is 0.62 while the MAP score of Doc2Vec is 0.66, and the NDCG score of BM25 is 0.92, while the NDCG score of Doc2Vec is 0.93.

BM25		Doc2Vec	
MAP score	NDCG score	MAP score	NDCG score
0.62310782	0.92014947	0.66417989	0.92602017

Where MAP refers to Mean Average Precision that evaluate model performances base on binary dataset. In this project, I convert the relevance scores that larger than 3 to be 1s while the rest to be 0s, the goal is to put highly relevant documents high up the recommended lists. NDCG refers to Normalized Discounted Cumulative Gain, its advantage is that it evaluates the position of ranked items and operates beyond the binary relevant/non-relevant scenario [10].

Discussion & Some Other things I have Tried

In the initial stages of this project, I encountered challenges in defining queries, particularly with queries like 'travel,' where the baseline model BM25 consistently returned the product 'travel socks' as the top result. Various attempts with different queries did not yield diverse top 50 outputs. To solve this problem, my first approach is to tokenize not only the review text of the product, but also combined their review summary and even the product names, but I found the output would be biased if the keywords appeared in queries also showed up in the product title. What's more, the results are similar even with those tokenized product title, review summary and review texts combined together. I still got several same products in the top 50 outputs from the system. Another approach I did was to combine all the review texts for products with the same name, The motivation behind this approach stemmed from the realization that the system consistently identified products with the same name differently due to distinct product IDs and associated reviewers. The adoption of this second approach proved to be a successful solution, as amalgamating all the review texts associated with a specific product name eliminated the possibility of encountering identical names among the top 50 most relevant products from the information retrieval system. However, I contemplated potential drawbacks to this approach. For instance, certain products might appear more frequently in the reviews simply because they are older, having been in existence longer than others, as a result, their corresponding combined reviews might be much longer than others. The use of longer tokenized input could introduce bias, potentially favoring products with an extensive history of reviews. I recognize that there may be newly invented products with fewer reviews that are nonetheless superior or more effective than their older counterparts. Thus, this approach may inadvertently favor products with a longer presence in the market, potentially overlooking innovative yet less-reviewed items that could offer superior value.

While annotating the ground truth relevance scores, I found that the recommended products from BM25 are more relevant while the provided input queries are more specific, because with only one single words, the BM25 would provide anything that is relevant. Also, if provide queries with keywords that has not appeared in the review texts, for example, 'comfortable clothes', since the dataset is related to health products, there were limited clothes, thus BM25 keep recommending products such as detergent, but not clothes, while using Doc2vec, the recommended products would include athlete socks that are more relevant to clothes. However, if the queries provided are very detailed and longer, such as 'Health snacks for weight loss', the performance of BM25 is much better than Doc2vec model in my system, where BM25 recommended protein bars and coconut oil while Doc2vec recommended battery, bags and tablets. Thus, it is tricky to find the best query that provides the most relevant products that we want, and the logic of input queries varies with different models used in information retrieval systems. As a result, the MAP score of BM25 is 0.62 while the MAP score of Doc2Vec is 0.66, and the NDCG score of BM25 is 0.92, while the NDCG score of Doc2Vec is 0.93. overall, according to the NDCG scores, the performance is relative satisfactory for an end user.

Conclusion

In this report, I built an information retrieval system that with different inputs from users, the system would provide the names of the top 50 most relevant products that corresponding to the input query. Using BM25 as the baseline model and Doc2Vec as the 'proposed' model. Doc2Vec performs slightly better according to the evaluation scores – the MAP score of BM25 is 0.62 while the MAP score of Doc2Vec is 0.66, and the NDCG score of BM25 is 0.92, while the NDCG score of Doc2Vec is 0.93. To be specific, BM25 performs better with more specific queries while the performance of Doc2Vec does not vary much for different queries, but the performance might be dependent on the document contents, which in this case is the combined review text for each product.

What I would have done differently or Next

If possible, I would have explored more time-consuming models, including certain deep learning models (such as bi-encoder, etc), to assess their potential impact on the system's performance. Additionally, I aim to experiment with learn-to-rank methods and leverage metadata, incorporating factors such as the length of reviews, the cumulative review count for a product, the average and latest review timestamps, average helpfulness ratings, and average review scores for product with the same name. For future work, If these approaches yield improvements in system performance, I intend to apply similar logic to diverse datasets for comprehensive evaluation.

Reference

[1]https://snap.stanford.edu/data/web-Amazon-links.html

[2] Chen Luo, William Headden, Neela Avudaiappan, Haoming Jiang, Tianyu Cao, Qingyu Yin, Yifan Gao, Zheng Li, Rahul Goutam, Haiyang Zhang, and Bing Yin. 2022. Query Attribute Recommendation at Amazon Search. In Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22). Association for Computing Machinery, New York, NY, USA, 506–508. https://doi.org/10.1145/3523227.3547395

[3] Zadel, M., & Fujinaga, I. (2004, October). Web Services for Music Information Retrieval. In ISMIR.

[4] ELLIS, D. (1989), "A BEHAVIOURAL APPROACH TO INFORMATION RETRIEVAL SYSTEM DESIGN", Journal of Documentation, Vol. 45 No. 3, pp. 171-212. https://doi.org/10.1108/eb026843

[5] Mauldin, M. L. (1991, September). Retrieval performance in ferret a conceptual information retrieval system. In Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 347-355).

[6] David Carmel, Doron Cohen, Ronald Fagin, Eitan Farchi, Michael Herscovici, Yoelle S. Maarek, and Aya Soffer. 2001. Static index pruning for information retrieval systems. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01). Association for Computing Machinery, New York, NY, USA, 43–50. https://doi.org/10.1145/383952.383958

[7] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval, 3(4), 333-389.

[8] P. Karvelis, D. Gavrilis, G. Georgoulas and C. Stylios, "Topic recommendation using Doc2Vec," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-6, doi: 10.1109/IJCNN.2018.8489513.

[9] https://shuzhanfan.github.io/2018/08/understanding-word2vec-and-doc2vec/

[10] https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832