MACHINE LEARNING FOR NLP

INFORMATION RETRIEVAL CHALLENGE BEATING BM25

KARUNATHASAN Nilany
SAMBATH Sïndoumady
DIA2

2023

TABLE OF CONTENTS

01 Introduction

02 Techniques & Approaches

03 Comparison of Models

04 Streamlit App

05 Conclusion

INTRODUCTION

This project focuses on enhancing information retrieval on the NFCorpus dataset, a collection of medical abstracts from PubMed publications. The primary goal was to beat the performance of the baseline BM25 model. We also developped a custom TF-IDF based retrieval system and a Streamlit app to provide an interactive interface for users to retrieve relevant documents.

Link to Colab Notebook
Link to Github Repository

TECHNIQUES & APPROACHES

II. Baseline Model : BM25

1. Overview

The baseline model, BM25, was provided as a starting point for information retrieval.
The implementation includes the following steps :

- Loading the NFCorpus dataset.
- Preprocessing documents and queries by tokenization and stopword removal.
- Utilizing the BM25 algorithm to compute scores for queries against a selected subset of documents.
- Evaluating model performance using the NDCG metric at the top 5 results.

2. Results

The NDCG score for the BM25 model on the selected subset of documents is approximately **0.8135**. This score serves as the benchmark for our subsequent custom TF-IDF model.

III. Custom TF-IDF-based Model

We chose to do a custom implementation of a TF-IDF-based information retrieval system, designed to be compared against the baseline BM25 model on the NFCorpus dataset.

The steps taken in this implementation include :

a) Text Preprocessing

The initial step involves preparing the text for analysis. This includes tokenization of both documents and queries, breaking them down into individual units. Stopwords are removed to focus on meaningful content, and a minimum word length criterion is applied for further refinement.

b) TF-IDF Vectorization

A critical component of the custom model is the creation of a TF-IDF vector representation of the document corpus. This step involves assigning numerical weights to terms based on their importance in the corpus. The scikit-learn TfidfVectorizer facilitates efficient vectorization, capturing the significance of terms within the entire document collection.

c) Cosine Similarity Calculation

The TF-IDF scores obtained are utilized for calculating cosine similarity between queries and documents. This metric measures the cosine of the angle between two vectors, providing a robust measure of similarity. The top documents are selected based on the computed cosine similarity scores.

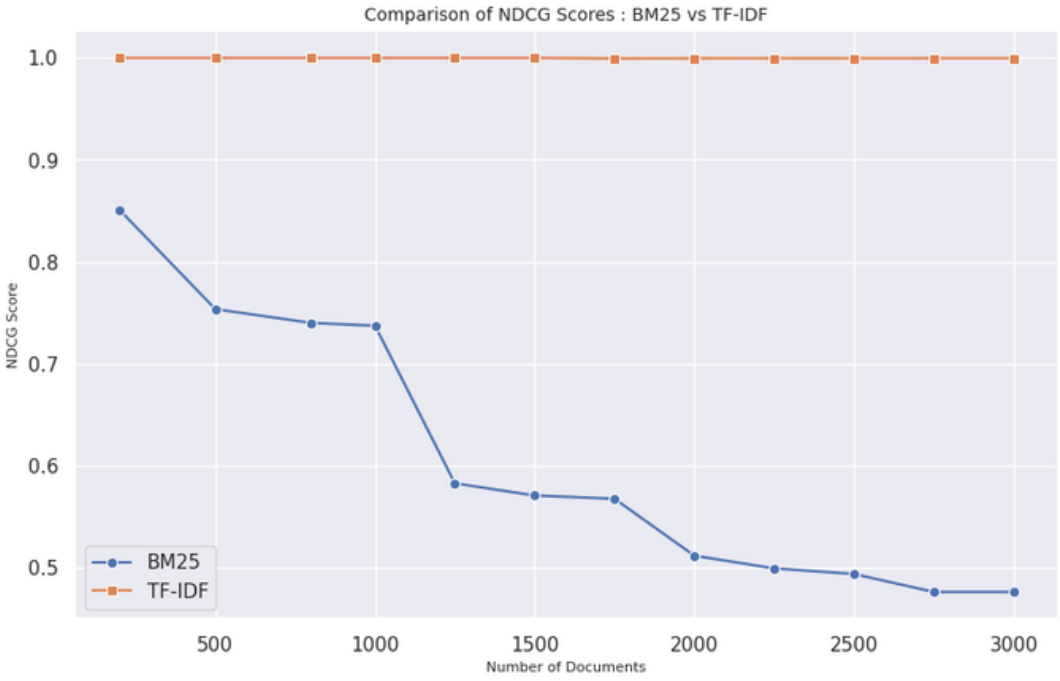
d) Performance Evaluation

The model's performance is assessed using the NDCG metric at the top 5 results. This evaluation strategy ensures a direct comparison with the BM25 baseline, focusing on the system's ability to retrieve relevant documents.

COMPARISON OF MODELS

The NDCG (Normalized Discounted Cumulative Gain) score for the BM25 model on a selected subset of 150 documents is approximately 0.8135 whereas the average NDCG score for the TF-IDF model on the same selected subset of documents is approximately 0.999.

We did a graph that compares the NDCG scores of BM25 and TF-IDF algorithm, across different numbers of documents. The x-axis represents the number of documents, while the y-axis represents the NDCG scores. The blue line with circular markers represents the NDCG scores for BM25, and the orange line with square markers represents the NDCG scores for TF-IDF.



The custom TF-IDF-based retrieval system maintains a stable NDCG@5 near 0.99, emphasizing its reliability across varying document counts. Meanwhile, the BM25 model exhibits a decreasing trend in NDCG score as the number of documents increases.

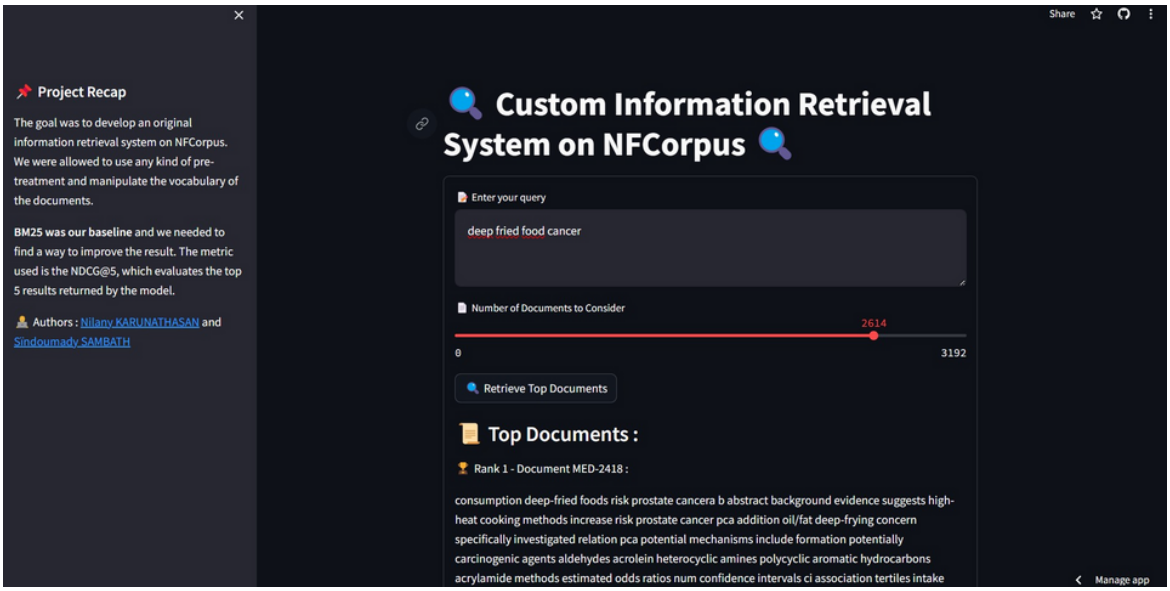
STREAMLIT APP

1. Overview

The development of a Streamlit application enhances the accessibility and user interaction with the custom TF-IDF-based information retrieval system. The application serves as a practical interface for users to input queries and retrieve relevant documents based on the TF-IDF model's calculations.

2. Features

- Intuitive Interface for Query Input
- Slider to Specify the Number of Documents
- Real-time Display of Top 5 Documents



3. How to use

- Public access : visit the public app through this link
- Local Access :

1. Clone the GitHub repository to your local machine using the following command :

```
git clone https://github.com/nilanyK/nlp_esilv.git
```

2. Change to the project directory using the command :

```
cd PROJECT1
```

3. Execute the Streamlit app by running the app.py file :

```
streamlit run app.py
```

4. Access the app :

- The output will display the link on which the server is running. Click on the link if it doesn't open directly.
- Enter your query and choose the number of documents to consider.
- Click the "Retrieve Top Documents" button to initiate the retrieval process and view the top documents in real-time.

CONCLUSION

In conclusion, this project aimed to elevate information retrieval within the NFCorpus dataset, specifically comprised of medical abstracts from PubMed publications. The overarching objective was to surpass the performance of the baseline BM25 model and it's a success. To achieve this, we introduced a customized TF-IDF based retrieval system and implemented a user-friendly Streamlit app, thereby enhancing the accessibility and interactivity of the retrieval process for users. The endeavors undertaken in this project not only contributed to advancing retrieval capabilities but also underscored the significance of tailoring models and applications to specific datasets, paving the way for more effective information extraction and utilization in the realm of medical research and literature.