Information Retrieval Search Engine

Authors:-
Ujjayan Pal CS21B084
Bapan Mandal CS21B016
Abhishek Mahajan CS23M002

*These authors have contributed equally to this work.

-:Abstract :-

Information retrieval (IR) refers to the method of acquiring pertinent data from extensive sets of structured or unstructured data. Search engines, a widely used implementation of IR, empower users to search and access relevant documents online. This research delves into strategies to enhance the efficiency of the vector space model in IR search engines. Various models, such as doc2query and Latent Semantic Analysis (LSA)[1], were examined, and their performance was assessed using evaluation metrics like recall, precision, F-score, nDCG, and MAP, with comparisons drawn against the VSM.

Keywords: LSA, doc2query

1 Introduction :-

The objective of this endeavor is to tackle the deficiencies of the previously created Vector Space Model (VSM) search engine utilized for Information Retrieval (IR) and tested on the Cranfield Dataset. The initial system exhibited multiple drawbacks that impeded its effectiveness. To surmount these challenges, this project utilizes diverse techniques like Doc2query and Latent Semantic Analysis to enhance the performance of the IR system. The primary aim of this project is to enhance the IR system's efficiency through the implementation of these methodologies.

2 Project definition:-

The aim of this project is to surmount the constraints of our search engine by incorporating enhancements. We have pinpointed several existing drawbacks of the system, which are as follows:

(a) Documents with comparable contexts but varying term vocabularies are not retrieved.

(b) The order in which terms appear in the document is lost in the vector space representation.

(c) Documents with vastly different contexts but similar term vocabularies are retrieved.

(d) Due to the sparse nature of the model, empty documents are retrieved as relevant for most queries.

In an effort to improve the Vector Space Model search engine and address its current limitations, we implemented several methods, including but not limited to Latent Semantic Analysis (LSA), and document expansion techniques like doc2query. Our objective was to elevate the performance of the search engine by harnessing these methods.

3 Motivation:-

A successful search engine is defined by its ability to achieve high recall while maintaining an acceptable level of precision. It should retrieve the majority of relevant documents while excluding irrelevant ones, ranking them based on their relevance and prioritizing the most pertinent documents. Conducting efficient searches using an inverted index also implies being limited by exact lexical matches between query and document terms. As a result, these retrieval models struggle to match related terms, leading to the vocabulary mismatch[3] problem . Vocabulary mismatch occurs when the relevance between a query and a document is not accurately estimated due to the absence of an exact lexical match between the query tokens and the document terms. This mismatch can arise when users express their intents using different words than those employed by the authors of the relevant documents.

The vocabulary mismatch issue has a significant impact on the entire retrieval process. When a relevant document and a query do not share any terms, the document remains unretrieved. Therefore, in this initiative, we aim to introduce a method that alleviates the vocabulary mismatch problem. The objective of this project is to overcome this limitation and improve the precision and recall of our information retrieval model by implementing techniques such as document expansion (Doc2Query)[5] and Latent Semantic Analysis (LSA).

4 Background and Related work:-

In our previous coursework, we constructed a search engine based on the Vector Space Model (VSM) approach. The VSM search engine incorporated various techniques, such as dividing documents into sentences, breaking down sentences into tokens, eliminating stopwords, stemming tokenized words, building an inverted index list, and representing both articles and queries using TF-IDF.

To rank documents, we measured the cosine similarity between articles and queries. We also assessed the performance of our search system using standard information retrieval metrics like Precision, Recall, F-score, Mean Average Precision (MAP), and normalized Discounted Cumulative Gain (nDCG).

Query expansion[2] is a widely adopted method to address the vocabulary mismatch issue. It involves expanding the query to encompass synonyms and terms related in meaning. This is necessary because users may articulate concepts differently from how they are represented in the dataset. Typically, query-term vectors are sparser than document-term vectors. While query expansion enhances the size of the query for better

document comparison, it can increase query complexity and introduce irrelevant terms, potentially leading to topic drift.

Another approach is to leverage title information in the dataset and assign higher weights to terms appearing in the title. While this helps identify the main concepts a document covers, it doesn't effectively address the vocabulary mismatch issue.

Document expansion, unlike query expansion, can offer greater advantages. Documents are usually longer than queries, providing more context for the language model to choose expansion terms. This holds even for passages or brief sentences, as contextualized language models handle natural language text more effectively than a simple set of keywords. An example of document expansion is the Doc2Query method. It employs a sequence-to-sequence transformer model to forecast queries that the document could address. These predicted queries are added to the document, which is then indexed and ranked based on a query to enhance retrieval outcomes.

Another strategy involves uncovering underlying concepts within documents through a technique known as Latent Semantic Analysis. In a standard vector space model, the vectors' dimensions are assumed to be orthogonal, meaning the terms representing these dimensions are unrelated. However, in reality, terms in different documents can be related, with similar terms appearing in similar documents. Latent Semantic Analysis aims to leverage the implicit higher-order structure in how terms are associated with documents (referred to as "semantic structure") to enhance the identification of relevant documents based on query terms.

5 Proposed Methodology:-

In our proposed method, we use Doc2Query to expand the document and then perform indexing using Latent Semantic Analysis to give better results.

5.1 Doc2Query

We use an variation of Doc2Query known as DocT5Query[4], in which we use a T5 model to predict new queries in place of a transformer model. The T5 model is a pre-trained encoder-decoder model particularly effective for text generation. We use a pretrained model[6] which is trained with datasets including these below besides others:

- (title, body) pairs from Reddit
- (title, abstract) pairs from S2ORC
- (title, body) pairs and (title, answer) pairs from StackExchange and Yahoo Answers!
- (title, review) pairs from Amazon reviews
- (question, duplicate question) from Quora and WikiAnswers
- (query, paragraph) pairs from MS MARCO, NQ, and GooAQ

With this pretrained model, we predict 3 queries for each document using top k random sampling and append them to the corresponding documents without any markup to separate the original document from these queries.

5.2 Latent Semantic Analysis/Indexing

After performing tokenization, lemmatization and stop word removal in these documents. We perform dimensionality reduction to get rank-k approximation term-doc matrix. We then perform cosine similarity between the vectors (query and doc) in this k-dimensional space to rank the documents for the query. We took k=609, since that gave us the best results.

This is obtained by singular value decomposition of the term-doc matrix. This results in the decomposition of the rectangular matrix into two square matrices (U and V) and one rectangular diagonal matrix(Σ). So, the matrix is decomposed into the form

$$X = U\Sigma V^T$$

Here, U essentially models the similarity behaviour between terms and V models similarity behaviour between documents. Arank-k approximation of X is made by analysing the eigen values present in the diagonal of Σ. This helps remove irrelevant features and in turn gives us an improved representation of X. We then transform the query vectors into this k-dimensional space and compare against the document vectors in this space to rank the documents

6 Experiments and results:-

We use the cranfield dataset to perform our experiments. The Cranfield dataset contains 1400 documents, 225 queries and query-document relevance judgements. We use the performance measures that were designed in our previous work to evaluate the performance of our approach against other approaches. The figure 1 shows the performance of our approach on this dataset in terms of Precision@k, Recall@k, F-Score@k, MAP@k and nDCG@k

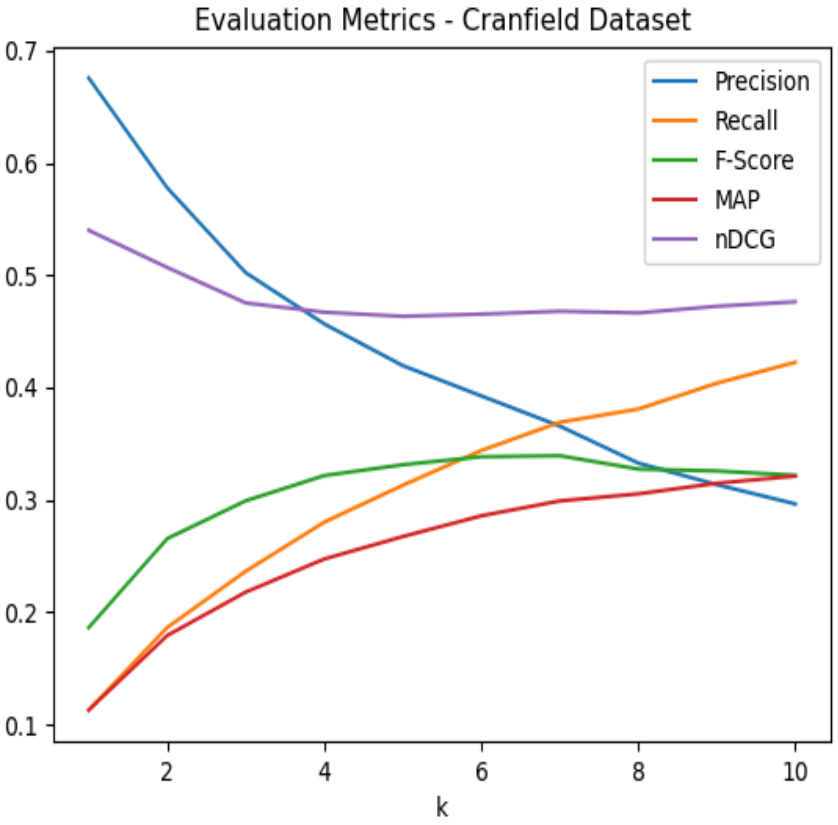

Fig. 1 The plot shows the performance of Doc2Query with LSA based on different performance measures

We evaluated our approach by comparing Precision@k and nDCG@k for values k=1 to k=10 with other methods discussed in this study. We selected Precision and nDCG as our evaluation metrics, as they are particularly relevant for search engine applications where precision is slightly more crucial than recall. The system's primary goal is to return the most relevant documents within the top few results, rather than retrieving all relevant documents. nDCG is also a valuable metric for comparison, as it considers the graded relevance of documents to the query. Since we have access to the query-document relevance judgments for this dataset, we can utilize these scores to calculate nDCG scores.

It must be noted that Doc2Query is non-deterministic, so it will give different results in different iterations. We are using the averaged results across 15 iterations for our comparison. The Precision@k values for different approaches is given in the figure 2. It shows that the precision values are much better than the VSM model and slightly better than the otherLSA models.

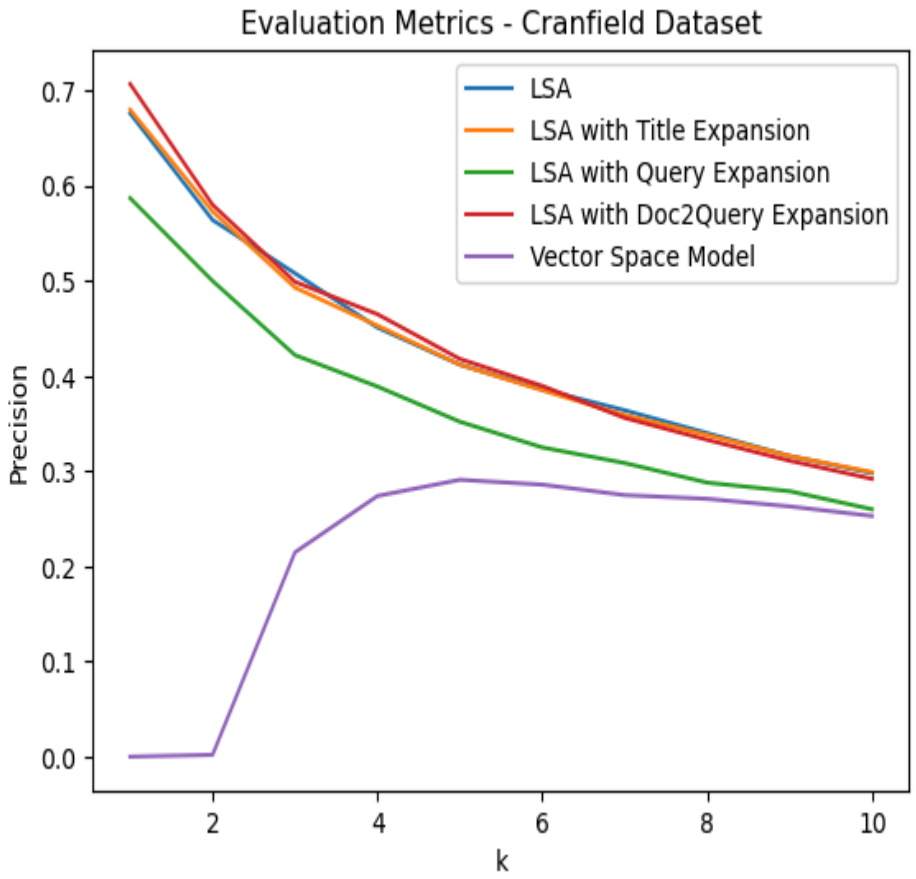

Fig. 2 The plot shows the Precision@k of the different approaches

The nDCG@k values for different approaches is given in the figure 3. The plot clearly shows that it performs much better than all the other approaches

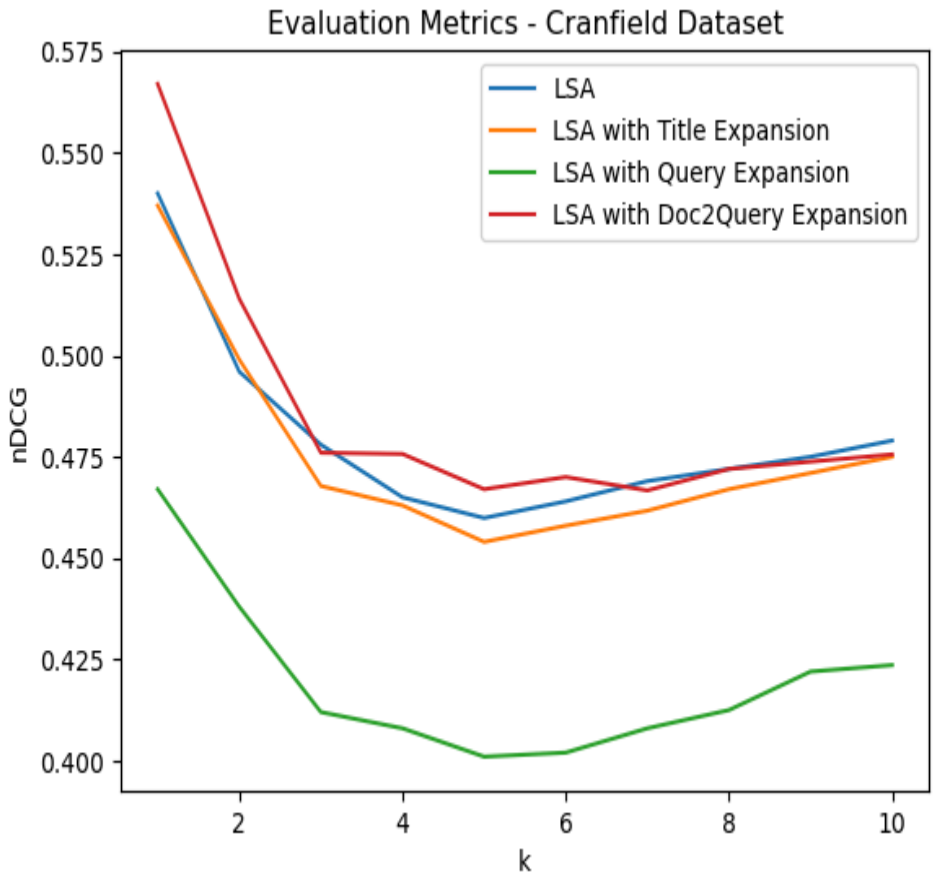

Fig. 3 The plot shows the nDCG@k of the different approaches

7 Observations:-

Based on figures 2 and 3, it is evident that our approach significantly outperforms the baseline VSM approach. Additionally, while the precision values are not substantially higher compared to LSA (without expansion), Doc2Query clearly exhibits superior performance in terms of nDCG. This suggests that although LSA and LSA with title expansion retrieve a comparable number of relevant results as LSA with Doc2Query expansion, the Doc2Query expansion is more effective in retrieving the most relevant documents within the first few ranks. Another notable observation is that LSA with query expansion performed worse than LSA without expansion, which could be attributed to the topic drift issue inherent in the query expansion approach.

8 Drawbacks:-

While our method shows a slight improvement over LSA, it is important to acknowledge that it comes with significantly higher computational demands and time requirements. Nevertheless, the combination of Doc2Query with LSA could be more suitable for real-world applications dealing with more varied data compared to the dataset used in this study. Since the Cranfield dataset is domain-specific, the application of Doc2Query did not yield enhanced results.

9 Conclusion:-

Following the exploration of various methods, we discovered that the integration of Doc2Query with LSA yields enhanced outcomes compared to both the baseline VSM model and the base LSA approach. These improvements can be attributed to the following factors:

• The generated queries effectively bridge the lexical gap in the search engine by incorporating synonyms, thereby enhancing query-document matching.

• Additionally, the re-weighting of words assigns higher importance to crucial terms, even if they appear infrequently within a paragraph, which further refines the search results.

References:-

[1] Deerwester S, Dumais ST, Furnas GW, et al (1990) Indexing by latent semantic analysis. Journal of the American society for information science 41(6):391–407

[2] Efthimiadis EN (1996) Query expansion. Annual review of information science and technology (ARIST) 31:121–87

[3] Furnas GW, Landauer TK, Gomez LM, et al (1987) The vocabulary problem in human-system communication. Communications of the ACM 30(11):964–971

[4] Nogueira R, Lin J, Epistemic A (2019) From doc2query to doctttttquery. Online preprint 6

[5] Nogueira R, Yang W, Lin J, et al (2019) Document expansion by query prediction. arXiv preprint arXiv:190408375 [6] Reimers N (2021) doc2query. URL https://huggingface.co/doc2query/all-with prefix-t5-base-v1