Group 4 Assignment 2 – Search Engine Design

1. Aims

With the rapid expansion of the internet and digital technology, the amount of information available has become overwhelming. The average person has limited time and attention span, typically only looking at the first 10 search results (Brin & Page, 1998). As a result, people often require an efficient way to locate relevant information quickly. Information Retrieval (IR) methods and search engine development assist individuals in finding pertinent information amidst the vast amount of digital content. Sophisticated algorithms for natural language processing enable users to obtain relevant results quickly while also accommodating their individual interests and preferences. The potential for real-world impact to domains such as healthcare, e-commerce, and education make information retrieval an important and interesting field.

Our project aim is to provide a comprehensive understanding of the basic components of information retrieval such as indexing, searching, and retrieval by creating a search engine using techniques taught during the curriculum and new ones learned through research. We will utilise the CISI dataset released by the Glasgow University Information Retrieval group containing a collection of 1460 articles on various topics such as linear algebra, medical research, and art. This dataset's inclusion of metadata, such as article title, author, and information about other related or referenced articles from the same dataset, is an essential feature. We intend to create an efficient search engine capable of producing fast, accurate, and relevant results based on queries of varying sizes and styles. Our project aims to demonstrate the potential of information retrieval systems and contribute to the development of more sophisticated search engines. We also aim to highlight the power of different models and how they can be tuned to increase performance.

2. Planned Approach

The vector space model is a mathematical framework that is widely used in information retrieval and natural language processing that represents textual documents as vectors within a high-dimensional space. It is commonly used because it is easy and simple to implement and offers fast and effective ranked retrieval. In this model, each document is represented by a vector, and each dimension of the vector corresponds to a specific term in the document. The value of each dimension indicates the relative importance of that term in the document and it is usually computed using a weighting scheme such as term frequency-inverse document frequency (TF-IDF) (Sharma & Kumar, 2020).

However, the vector space model has its limitations which can make it less effective to use in some information retrieval datasets:

limitation	potential solution
Does not take into account contextual meaning of terms (Gaschi, 2020).	Use of NLP techniques such as latent semantic analysis (LSA)
inability to establish connections among important terms, thus blocking their linkage (Sharma & Kumar, 2020)	Use of word embeddings (Dumais, 2005).

Furthermore, non-optimised VSMs are worse at handling longer queries – performance decreases because longer queries are represented as longer, more sparse vectors that result in decreased similarity to documents that are actually relevant. As mentioned before, this is also interconnected to its lack of semantic fluency, as longer queries may have less relevant or less important terms which can end up becoming over-represented.

On the other hand, the BM25 information retrieval model is a ranking function that used by search engines to estimate the relevance of documents to a given search query and is based on a core probabilistic framework. This model is widely used because of its high level of efficiency and effectiveness for ranking documents based on their relevant query. As a probabilistic model, it overcomes some of the limitations of the VSM model such as query expansion and capacity to handle noisy data (Wu et al., 2005). Additionally, the framework used by BM25 models expand on the traditional TF-IDF formula to account for term saturation, term importance and document length, making it a more robust retrieval model.

As such, our tasks and the proposed product of our investigation will be:

1. The formulation of two search engine models, one based on the BM25 algorithm and one using the vector space model.
2. A comparative analysis of the performance of the two prototypes using different evaluation metrics.
3. A focus on fine-tuning features of the better performing model.

However, we recognise that the BM25 model also has its limitations – it also doesn't consider the semantics and meanings of query terms and the indexed document, leading to a suboptimal output of results particularly when the query terms have multiple meanings (Gaschi, 2020). As such, search engine performance on our dataset may not be improved majorly, but we expect to see some increase in performance. In future, we hope to improve performance of our search engine through the integration of features into our search engine that encompass the semantic information of both the query and documents. This enhancement can be achieved by leveraging natural language processing (NLP) tools that facilitate the extraction of features such as sentiment analysis scores, which will work to enhance the precision of the model by capturing the core meaning of both the query and document (Kim et al., 2017).

If the BM25 model performs better, we will then focus on developing the model and tuning it to increase performance. This process can involve steps such as:

- optimisation
- modifying parameters (k1, k2, k3, b, term weight)
- tuning the ranking function

Overall, the comparative approach that we describe above has proven to be a valuable approach to developing search engines based off research and our general further reading into the development process of existing high performing information retrieval systems. It will give us the flexibility to make changes and improvements based off results which we hope will enable us to produce a well performing efficient search engine.

3. Dataset

Our project will be using a public dataset from the University of Glasgow's Information Retrieval Group called the **CISI collection**. The dataset can be accessed or downloaded using the following link: https://ir.dcs.gla.ac.uk/resources/test_collections/cisi/ . Our project will be using 3 files available as part of the CISI dataset:

1. **CISI.ALL**
   A 2.2 MB txt file containing 1460 documents. Each file contains a unique ID (.I), title (.T), author (.A), abstract (.W) and list of cross-references to other documents in the dataset (.X). The CISI.ALL file containing the documents, and basic document related information will be used in conjunction with the CISI.QRY file containing queries to create experiments with our IR models. The articles included in the dataset pertain to a variety of topics ranging from linear algebra, medical research, and art.

2. **CISI.QRY**
   A 68 KB file containing a total of 112 queries. Each query has a unique ID (.I) and query text (.W).

3. **CISI.REL**
   An 81KB file that contains the 'ground truth' of the mapping of query ID's (column 0) to document ID's (column 1). One query can be mapped to multiple document ID's. This file will be used along with the CISI.ALL and CISI.QRY files to train and evaluate our IR model. Thus, any non-relevant documents have not been mapped to the query. It is worth noting that there is notion of level of relevancy or ranking on which documents are more relevant to a query based off this file.

   Example figures depicting the general structure of the 3 files being used:

CISI.ALL

```
.I 1
.T
18 Editions of the Dewey Decimal Classifications
.A
Comaromi, J.P.
.W
   The present study is a history of the DEWEY Decimal
Classification.  The first edition of the DDC was published
in 1876, the eighteenth edition in 1971, and future editions
will continue to appear as needed.  In spite of the DDC's
long and healthy life, however, its full story has never
been told.  There have been biographies of Dewey
that briefly describe his system, but this is the first
attempt to provide a detailed history of the work that
more than any other has spurred the growth of
librarianship in this country and abroad.
.X
1 5 1
92 1 1
262 1 1
556 1 1
1004 1 1
1024 1 1
1024 1 1
```

CISI.QRY

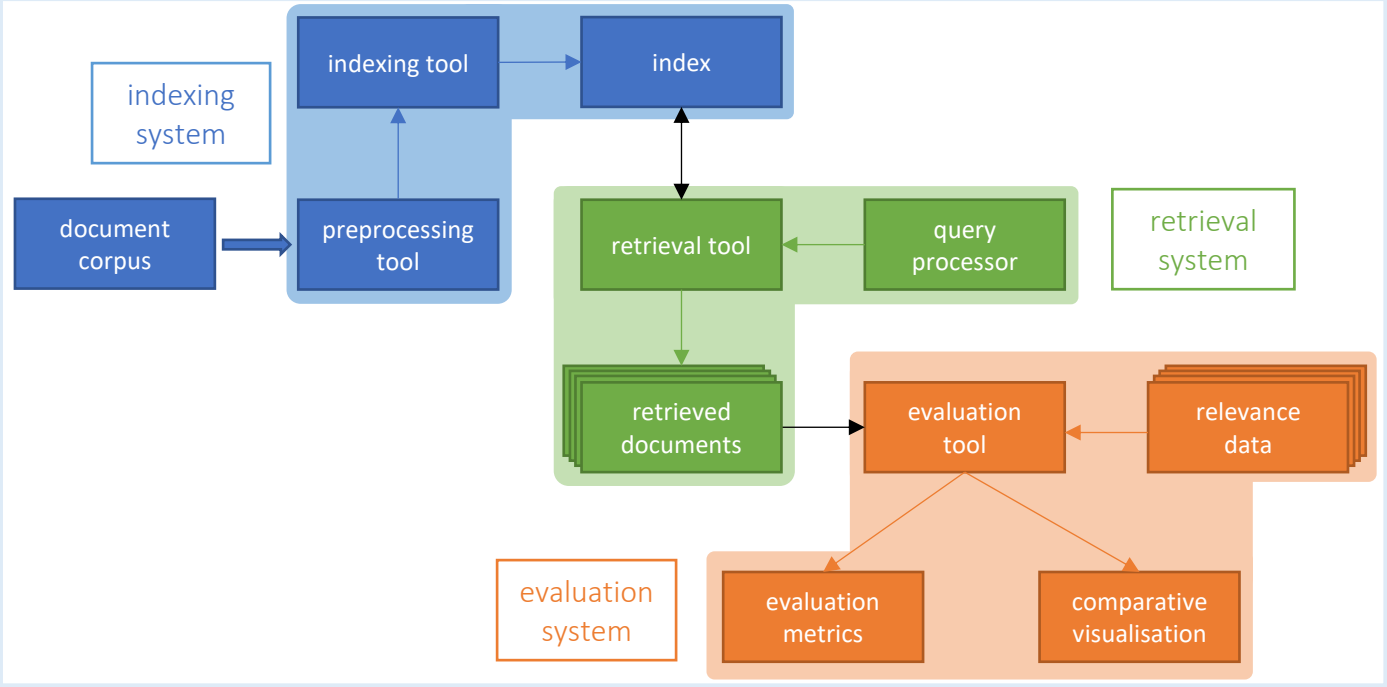
```
.I 1
.W
What problems and concerns are there in making up descriptive titles?
What difficulties are involved in automatically retrieving articles from
approximate titles?
What is the usual relevance of the content of articles to their titles?
.I 2
.W
How can actually pertinent data, as opposed to references or entire articles
themselves, be retrieved automatically in response to information requests?
.I 3
.W
What is information science?  Give definitions where possible.
```

CISI.REL

```
1   28  0  0.000000
1   35  0  0.000000
1   38  0  0.000000
1   42  0  0.000000
1   43  0  0.000000
1   52  0  0.000000
1   65  0  0.000000
1   76  0  0.000000
1   86  0  0.000000
1  150  0  0.000000
1  189  0  0.000000
1  192  0  0.000000
1  193  0  0.000000
1  195  0  0.000000
1  215  0  0.000000
1  269  0  0.000000
1  291  0  0.000000
1  320  0  0.000000
1  429  0  0.000000
1  465  0  0.000000
1  466  0  0.000000
1  482  0  0.000000
1  483  0  0.000000
1  510  0  0.000000
1  524  0  0.000000
1  541  0  0.000000
1  576  0  0.000000
1  582  0  0.000000
1  589  0  0.000000
1  603  0  0.000000
1  650  0  0.000000
1  680  0  0.000000
1  711  0  0.000000
1  722  0  0.000000
1  726  0  0.000000
1  783  0  0.000000
1  813  0  0.000000
1  820  0  0.000000
1  868  0  0.000000
1  869  0  0.000000
1  894  0  0.000000
1 1162  0  0.000000
1 1164  0  0.000000
1 1195  0  0.000000
1 1196  0  0.000000
1 1281  0  0.000000
2   29  0  0.000000
2   68  0  0.000000
2  197  0  0.000000
2  213  0  0.000000
2  214  0  0.000000
2  309  0  0.000000
2  319  0  0.000000
2  324  0  0.000000
```

4. Search Engine Architecture and Retrieval Models

As we are comparing the performances of two different models, our search engine architecture may be subject to slight change, but we have given an outline below:



Indexing System

The indexing framework is the same for both and has 2 steps:

1. Pre-processing tool: the first pre-processing step is designed to extract the relevant information from our data – the unique ID, author, title and the abstract. The pre-processing tool then performs natural language processing of the extracted data via tokenization. Following this, further term processing will take place including stop word removal and stemming. The details of this step are tentative and subject to change.
2. Indexing tool: the indexing tool maps each term to its associated document. At this stage various metrics independent of the query such as term frequency and inverse document frequency are calculated.

Retrieval System

The retrieval framework also contains 2 steps that differ depending on which model is used:

3. The query processor interprets and analyses the query syntax (like the pre-processing tool) to create a vectorised search request that can be robustly read against the index.
4. The retrieval tool executes the search request against the index:
    a. using the BM25 algorithm, which then returns the top-n (the value of n will be deliberated at a later stage) results as a collection of documents. Results will be ranked by the BM25 score. As a probabilistic retrieval model, it uses the inverse document frequency but has additional parameters to account for term saturation, term importance and document length.
    b. against the vector space model algorithm. Similarity is computed and the results are then displayed, with documents shown in descending order of similarity. Results will

be ranked using cosine similarity, which uses the cosine of the angle between the vector of the given query and the vectors of documents in the index.

Evaluation System

5. Our models will be tested using queries from the CISI.QRY file and the results returned from the retrieval system will then be compared to the CISI.REL data containing the 'ground truth'. To evaluate performance for individual queries, we will use precision, recall and F1.
6. Additionally, the evaluation system includes an appropriate visualisation framework allowing for comparison of individual query performance.
7. The metric to be used to evaluate overall model performance for both retrieval models is the mean average precision (MAP). This allows for evaluation of more than one query to give a better indication of overall model performance (Yue et al., 2007). Visualisations comparing the BM25 and VSM models will also be included.

5. Tools Used

All members of the team are on the MSc Data Science and AI course and have programming capabilities limited to python. All coding will be conducted in python and project work will focus on using the packages and software introduced during semesters 1 and 2 teaching. The following table contains a list of tools that we intend to utilise. As we intend to take a systematic trial, error, and improvement-based approach to the project, new software and tools may be integrated throughout the project on a needs-based basis.

	Software/Tools	Description	Hyperlinks
1	Python & math module	Programming language used to implement our Search Engine. Built-in modules such as math will be used for calculating trigonometric ratios for a given angle. It will be used to calculate IDF for the BM25 implementation or cosine similarity for VSM.	python math
2	NumPy & Pandas	NumPy supports vectorisation used to perform operations on arrays. Pandas also supports vectorisation and will be used to analyse, clean, explore and manipulate our chosen CISI dataset.	NumPy Pandas
3	Google Collab/Jupyter Notebooks	Web-based IDE that will be used to implement our project. This will also be used to store CISI dataset and documents as well as other resources and documents.	Google Collab JupyerHub
4	GitHub	Code repository and merging.	GitHub
5	CountVectorizer/TfidfVectorizer	Sklearn packages used to convert a collection of text documents to a vector of term/token counts.	CountVectorizer TfidfVectorizer
6	NLTK (Natural Language Toolkit)	Library for natural language processing techniques. Possesses functionality to perform tokenization, stemming, and stop-word removal.	NLTK
7	ElasticSearch	Library used to index and retrieve documents during the retrieval phase of the project.	ElasticSearch
8	PyLucene	Based on Lucene it is a package used to index and search documents that we could consider to indexing and searching documents if we encounter issues with Elasticsearch.	PyLucene
9	Whoosh	Like Lucene, Whoosh is a fast full-text indexing and searching library that can be implemented in Python. We will consider using Whoosh for indexing and searching if we encounter issues with our other options.	Whoosh
10	Matplotlib and Seaborn	Packages used to produce visualisations during the Search Engine Evaluation stage.	Matplotlib Seaborn

6. Roles and Responsibilities

As a team, we have decided to delegate tasks based on individual strengths. However, as this is a group project, we aim to collaborate on every phase of the search engine design. This is reflected in the roles and responsibilities in the table below:

	Member	Responsibility
1	Sobana Navaratnasingham	Project plan, selection and preparation of dataset, retrieval system framework and implementation, documentation of tools used
2	Sabrina Tohow	Project timeline, indexing framework and implementation, search engine architecture design
3	Jacqueline Ochonma	Evaluation framework and implementation, evaluation visualisation, investigation finalisation

7. Project Timeline

	Week 3 (6th Feb)	Week 4 (13th Feb)	Week 5 (20th Feb)	Week 6 (27th Feb)	Week 7 (6th March)	Week 8 (13th March)	Week 9 20th March)	Week 10 (27th March)	Week 11 (3rd April)	Week 12 (10th April)
Primary Research										
Obtain Dataset										
Draft Project Plan										
Finalise Search Engine Design										
Implement Indexing System										
Implement Retrieval System										
Implement Evaluation System, Testing & Quality Checks										
Complete Presentation and Final Demo										

8. References

Brin, S. &Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer networks (Amsterdam, Netherlands : 1999).* 30 (1-7): 107.
Yue, Y. et al. (2007) "A support vector method for optimizing average precision," Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval [Preprint]. Available at: https://doi.org/10.1145/1277741.1277790.
Gaschi, F. (2020) Building a medical search engine - step 3: Using NLP tools to improve search results, Medium. Posos. Available at: https://medium.com/posos-tech/building-a-medical-search-engine-step-3-using-nlp-tools-to-improve-search-results-8fc0afbee75b (Accessed: March 6, 2023).
Kim, S. et al. (2017) "Bridging the gap: Incorporating a semantic similarity measure for effectively mapping pubmed queries to documents," Journal of Biomedical Informatics, 75, pp. 122–127. Available at: https://doi.org/10.1016/j.jbi.2017.09.014.
Sharma, A. and Kumar, S. (2020) "Semantic web-based information retrieval models: A systematic survey," Data Science and Analytics, pp. 204–222. Available at: https://doi.org/10.1007/978-981-15-5830-6_18.
Wu, J. et al. (2005) "An improved VSM based information retrieval system and fuzzy query expansion," Fuzzy Systems and Knowledge Discovery, pp. 537–546. Available at: https://doi.org/10.1007/11539506_68.
Dumais, S.T. (2005) "Latent semantic analysis," Annual Review of Information Science and Technology, 38(1), pp. 188–230. Available at: https://doi.org/10.1002/aris.1440380105.