PRI Report - Delivery 2

Group 04
96656 - Joaquim Bação
99258 - Jorge Santos
110946 - Francisco Abreu

1 Introduction

In the second stage of the project, the focus shifts from basic retrieval to leveraging document organization and relationships to potentially enhance retrieval effectiveness. Two complementary strategies are explored [4].

First, unsupervised clustering techniques are applied to the collection, aiming to uncover its latent structure and topical organization. By grouping documents based on content similarity, it becomes possible to derive interpretable clusters that may reveal semantic groupings or thematic overlaps. This analysis is carried out with a strong emphasis on interpretability, using tools like Kullback-Leibler divergence to highlight distinguishing terms across clusters [2].

Second, the project investigates the use of graph-based centrality metrics to support information retrieval. Specifically, document relationships are encoded into a similarity graph, and a variation of the PageRank algorithm is applied to compute centrality scores. These scores can then guide or re-rank the retrieval results, with the hypothesis that more central documents may serve as better entry points to the information space.

However, before discussing the aforementioned topics, an update summarization of the first delivery report (along with the notebook, both present in 2nd delivery zip file), with emphasis on the changes implemented according to received feedback.

2 Delivery 1 Summary

To address the first delivery objectives, it was asked to find a suitable model for information retrieval. Various combinations were tested, considering different preprocessing options as well as multiple Information Retrieval (IR) models, including TF-IDF, BM25, BM25f, and Language Models (LM) with their respective variants. Additionally, the effectiveness of Reciprocal Rank Fusion (RRF) will be discussed in this report.

To guide the iteration improvements, a baseline (TF-IDF model) was established and an evaluation system was put in place utilizing the *pytrec_eval* [6] framework. This utilized metrics such as MAP and NDCG@10 to evaluate the models' document ranking order according to each provided query.

2.1 Design Choices

As mentioned before in this work, decisions will be made that will impact the end results of this work, and this section will describe them.

2.2 Preprocessing

To determine which preprocessing method would provide the best results when evaluating the indexes, four different preprocessing options were tested: **Raw**, **Standard**, **Stemming**, **Language**.

2.3 Indexing

The first programming task involved developing a function to preprocess the collection and, using existing libraries, build an inverted index.

A Whoosh index is created to enable fast and efficient searching and retrieval of documents based on tokenized and processed text. The documents in the dataset are processed by iterating through each document, extracting its title and abstract, and adding them to the index. This structure allows for efficient indexing and retrieval of relevant documents.

2.4 IR Models

For Information Retrieval models, 6 algorithms were utilized: 4 based on the whoosh scoring framework (TF-IDF, BM25, BM25f, LM), 1 that does not provide ranking of results (Boolean Query), however it is present in the 1st delivery notebook and a results combining ranking algorithm (RRF). For this report, emphasis on the ranking schemes since these provide more palpable results to be evaluated.

2.5 Ranking

The ranking process was applied to retrieve and order documents based on their relevance to each query. Different retrieval models, including TF-IDF, BM25, BM25F and LM, were used to generate ranked lists of documents, ensuring a diverse evaluation of ranking strategies. The ranking results were then processed to extract the top 1000 documents per query, forming the basis for subsequent evaluation.

As an extra, a time-based boost [3] was applied for the year of the publication to improve the results of the IR models since most results are only relevant after 2019 (year in which the pandemic started).

2.6 Evaluation

The IR system's performance was assessed using `pytrec_eval` [6] with multiple evaluation metrics. A total of 50 queries with 1000 ranked documents per query were used, resulting in the evaluation of 50 000 documents, ultimately, ensuring a comprehensive performance analysis across diverse queries.

The performance of the IR model systems were compared using the following metrics: MAP@1000, NDCG@1000, NDCG@10, and P@10, which assess the

relevance and ranking quality of the retrieved documents. These metrics were chosen to provide a comprehensive analysis of the system's effectiveness in returning relevant results, particularly focusing on ranking quality, precision, and recall.

3 Questions to Explore - 1st Delivery

3.1 Characterize the Document Collection D and Topic Collection Q

The corpus D consists of 192,509 documents. The Fig. 9 in the appendices shows that the majority of documents were published between 1980 and 2021, with a peak during the pandemic.

The dataset captures a **temporal evolution of research**, which is useful for *time-based retrieval* (as will be seen later), providing insights into trends and emerging research areas over time.

3.1.1 Characterizing the Document Collection (D)

Titles have an average length of 12.6 words, with a maximum of 146 words. Abstracts, on the other hand, show a larger variance, with some being very concise while others contain extensive text up to 18,000 words.

3.1.2 Characterizing the Topic Collection (Q)

The topic collection contains 50 queries, structured into (title, description, and narrative). **Query Length Distribution:** The title field (i.e., the keyword query) varies in length, with most containing 2-5 words (Fig 1). Fig 8 in the appendices shows the most frequent terms in queries, with "coronavirus" and "COVID-19" dominating the distribution.

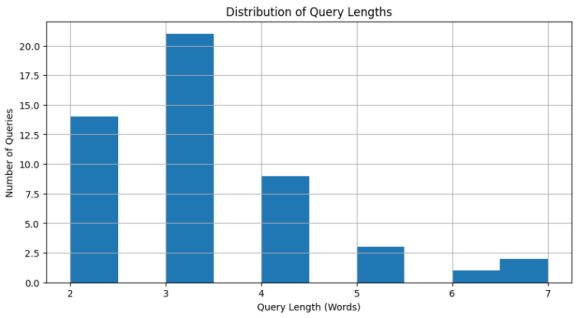

Figure 1: Distribution of Query Lengths.

3.1.3 Text Processing Steps

The text processing workflow consisted of multiple steps, aimed at standardizing and cleaning the dataset for more effective analysis and including **Tokenization**, **Lowercasing**, **Stopword Removal** and **Punctuation Removal**

Before Processing: The raw term frequency is shown in Fig. 10 in the appendices. The most frequent terms include stopwords and punctuation, which contribute to noise in retrieval tasks.

After Processing: After removing stopwords and punctuation, the dataset became more refined and focused on biomedical concepts. The most frequent words now relate directly to the core subject matter as seen in Fig. 11 in the appendices.

3.1.4 Overlapping Top Terms Among Q Topics

Lexical Overlaps: Queries and documents share key terms, including "biomarkers", "drug", "transmission", "symptoms", and "serological".

Semantic Overlaps: Using word embeddings based on Word2Vec reveals that query terms have meaningful similarities within the dataset (e.g, biomarkers: biomarker, signatures; analysis: analyses, study)

This suggests that query expansion techniques could significantly enhance retrieval performance by incorporating semantically related terms.

3.2 Characterize the performance of the IR system

3.2.1 Indexing Performance Across Preprocessing Methods

To evaluate the space and time requirements for processing and indexing, the four preprocessing methods were tested again: raw (no analyzer), standard (removes stop words and converts all tokens to lowercase), stemming (which adds stemming to the standard method using Porter) and language (which adds stemming to the standard method using Snowball). The indexing times and index sizes were recorded and are presented in the graph below (Figure 2).

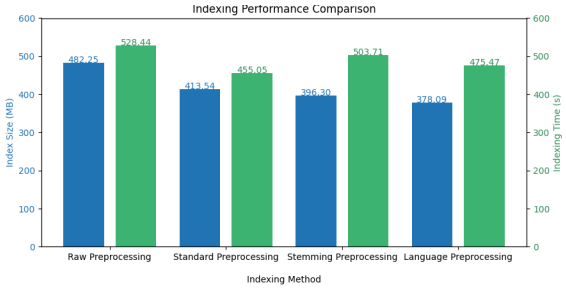

Figure 2: Indexing Performance Across Preprocessing Methods.

Raw indexing requires the most space and time because every word is indexed as a separate token without any filtering, leading to higher storage and processing costs. Standard indexing improves efficiency by applying basic preprocessing, which reduces the index size compared to raw indexing and makes processing faster.

Stemming further reduces space usage by converting words to their root forms, eliminating variations of the same word. However, this extra computational step makes it slower than standard indexing. Language-specific preprocessing achieves the smallest index size by applying more advanced linguistic rules but requires slightly more processing time than standard indexing, though it remains faster than stemming.

3.2.2 Ranking Performance Across Retrieval Models

To evaluate the space and time required for the retrieval stage, memory usage and execution time of different ranking models were measured. Fig. 3 presents these metrics for TF-IDF, BM25, BM25F and LM.

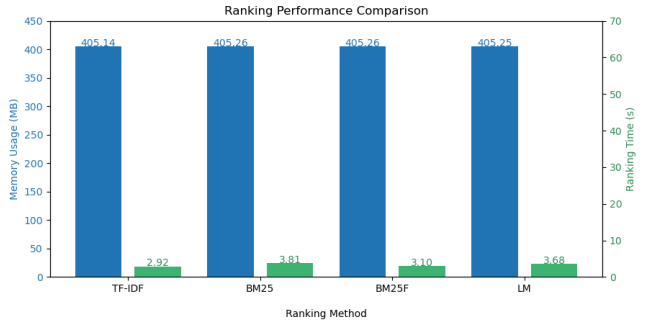

Figure 3: Ranking Performance Across Retrieval Models.

Memory usage remained constant across all models, indicating that the ranking function's RAM consumption was not significantly impacted by the choice of retrieval model. However, execution time showed slight variations. TF-IDF completed the ranking process faster than BM25 models and LM, which took about the same amount of time. This difference can be attributed to the additional computations of these models, such as document length normalization, which introduce more complexity compared to the simpler frequency-based calculations of TF-IDF.

3.3 Given a specific p, is the implemented IR system better at providing recall or precision guarantees?

To determine whether the system is better at promoting precision or recall with a given number of p documents, the results of the execution must be compared with the referenced ones by *qrels*. Fig. 4 shows that the recall increases with the number of documents being considered in the ranking (the bigger the number of documents considered, the bigger the probability a relevant document is considered, assuming the system is not perfect), whilst the precision presents the opposite trend, with the number of documents considered increasing reflecting a decrease in the system precision (the bigger the number of documents, the higher the probability more irrelevant documents are considered ranked). Overall, these results are in accordance with the expected trends for these metrics.

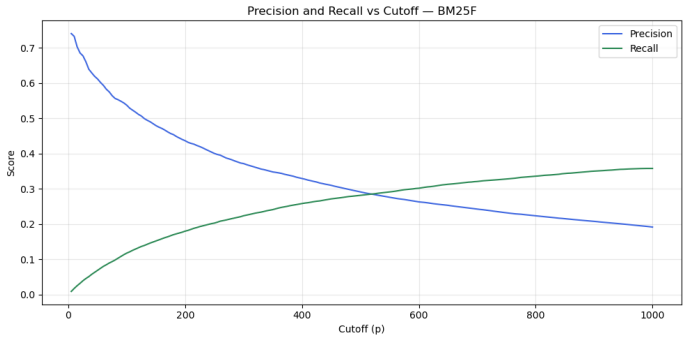

Figure 4: Precision and Recall values as p increases.

3.4 Which p should be used by the IR system if the user has a preference towards: minimizing false positives, minimizing false negatives, or maximizing true positives?

The choice of p (the number of retrieved documents) significantly impacts the trade-off between **precision** and **recall**. Depending on the user's objective, the IR system should adjust p accordingly:

Minimizing False Positives (FP) – Prioritizing Precision: False positives occur when an *irrelevant* document is retrieved. Users who cannot tolerate incorrect documents should prioritize precision. Using **smaller values of p (e.g., 10, 20, 50)** ensures that only the most relevant documents are retrieved.

The best evaluation metrics for this scenario are **Precision@10 (P@10) and nDCG@10**, which measure the relevance of top-ranked results. Retrieval models such as **BM25F and TF-IDF** are best suited for precision-focused ranking.

Minimizing False Negatives (FN) – Prioritizing Recall: False negatives occur when a *relevant* document is not retrieved. To avoid missing relevant documents, the system should use **larger values of p (e.g., 500, 1000)**.

Mean Average Precision (MAP) is a useful metric for balancing recall and ranking effectiveness, while **Recall@1000** measures the proportion of relevant documents retrieved. Retrieval models such as **PL2 (Language Model) and BM25F** work well for recall-oriented retrieval.

Maximizing True Positives (TP) – Balancing Precision and Recall: True positives refer to *retrieved documents that are actually relevant*. To maximize TP, the IR system should balance both precision and recall, using an **intermediate value of p (e.g., 100, 200, 500)**.

nDCG@10 and MAP are the best evaluation metrics for this trade-off. A hybrid approach using **BM25F combined with Reciprocal Rank Fusion (RRF)** can further enhance the ranking by integrating multiple retrieval models.

3.5 Performance Variation Across Queries

Evaluating the retrieval effectiveness of the system requires not only assessing overall performance but also understanding how performance varies across different queries. Some queries may consistently yield high-quality results, while others perform poorly due to factors such as ambiguity, topic complexity, or limitations in the ranking function. To investigate this, per-query performance was analyzed using key evaluation metrics.

The goal of this analysis is to determine whether performance significantly varies across queries and, if so, to identify which queries or topics exhibit lower retrieval effectiveness. By computing metrics like MAP@1000, NDCG@1000, NDCG@10, and P@10 on a per-query basis, the degree of variability in retrieval effectiveness can be measured. Additionally, computing summary

statistics such as mean, standard deviation, and minimum/maximum values allows us to quantify the consistency of the ranking function.

After running evaluation per query, these were the results:

Model	Mean	Std Dev	Min	Max
TF-IDF	0.5020	0.3228	0.0000	1.0000
BM25	0.7080	0.3104	0.0000	1.0000
BM25F	0.7320	0.2908	0.0000	1.0000
LM	0.7100	0.3048	0.0000	1.0000

Table 1: P@10 - Model Evaluation Metrics

Model	Mean	Std Dev	Min	Max
TF-IDF	0.4413	0.2974	0.0000	0.8611
BM25	0.6631	0.3016	0.0000	1.000
BM25F	0.6687	0.2936	0.0000	1.0000
LM	0.6588	0.3006	0.0000	1.0000

Table 2: nDCG@10 - Model Evaluation Metrics

Model	Mean	Std Dev	Min	Max
TF-IDF	0.2615	0.1437	0.0142	0.5696
BM25	0.3773	0.1845	0.0199	0.7176
BM25F	0.3890	0.1824	0.0180	0.6915
LM	0.3823	0.1972	0.0102	0.7587

Table 3: nDCG@1000 - Model Evaluation Metrics

Model	Mean	Std Dev	Min	Max
TF-IDF	0.0960	0.0885	0.0010	0.3129
BM25	0.1916	0.1427	0.0008	0.5247
BM25F	0.2021	0.1453	0.0006	0.5130
LM	0.1958	0.1513	0.0003	0.5419

Table 4: MAP@1000 - Model Evaluation Metrics

From the results, BM25, BM25F and LM consistently outperform TF-IDF across all metrics, with higher mean values and better worst-case performances, reflected by their higher minimum values. All three, the BM25, BM25F and LM achieve similar high maximum values, indicating their potential for high performance on specific queries, while TF-IDF has consistently lower maximum values, which aligns with its overall weaker performance.

In general, BM25, BM25F and LM are more reliable and consistent, making them preferable choices for most use cases, while TF-IDF can still be useful for certain high-performance queries despite its inconsistency.

3.6 How different text processing and scoring options affect retrieval? Is reciprocal rank fusion useful to place ensemble decisions?

How different text processing and scoring options affect retrieval? From the several configurations tested, it was found that the model that maximized the MAP and nDCG@10 metric was the BM25f with language preprocessing (utilizing English) and 3 times the weight of the abstract over the weight of the title (boost="title":1, "abstract": 3). To allow for comparisons, Fig. **??** shows the evaluation results for different models according to the established metrics.

Is reciprocal rank fusion useful to place ensemble decisions? The goal of RRF is to combine rankings from multiple retrieval models, ensuring that documents ranked highly by any model are promoted in the final list. It is important to notice that Reciprocal Rank Fusion (RRF) operates on a different scale from LM and BM25 because it is a rank-based fusion method, whereas LM and BM25 are score-based retrieval models.
To further evaluate the effectiveness of RRF, a comparison of its ranking outputs with those of BM25F and LM (the best models) is required. Fig. 12 in the appendices presents the document rankings for one of the queries.
From the results, it was observed that RRF scores are **closely aligned with BM25F and LM ranks**, indicating that it effectively integrates information from both models. The RRF ranking ensures that documents highly ranked by any model maintain strong positions in the final list.
The equation governing Reciprocal Rank Fusion [1] with k=60 ensures that documents ranked highly by any individual retrieval model receive a boost in the final fused ranking. This is particularly useful when different models highlight different relevant documents. By summing the reciprocal rank contributions across models, RRF helps establish consensus in retrieval, reducing the reliance on a single retrieval method and promoting a balanced ranking.
If the fused ranking consistently outperforms individual models in retrieval quality metrics such as NDCG and MAP, it is possible to conclude that **fusion is beneficial**, as seen in Fig. 6.

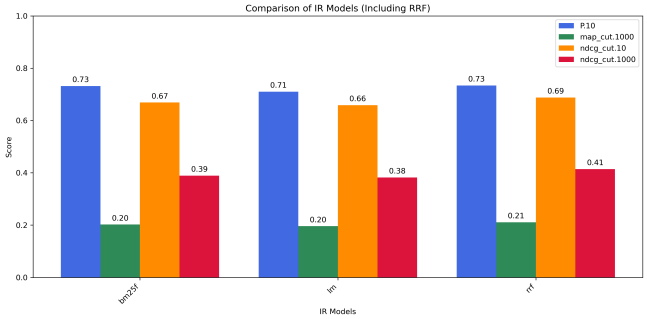

Figure 6: Reciprocal Rank Fusion results compared with BM25F and LM models.

4 Clustering approach

4.1 Clustering

To group documents in the collection, clustering algorithms were applied using different approaches and distance metrics. The choice of algorithm and distance metric was crucial.
To determine the optimal number of clusters, different

values were evaluated of k using internal metrics (as explained in 4.6.1). A detailed explanation of the preprocessing steps applied prior to clustering can be found below.

Embedding TF-IDF was used to embed the documents into a sparse, high-dimensional matrix. This representation works better with cosine similarity, and for that reason, vectors were normalized to unit length.

4.2 Interpret

Although the task suggests using the medoid — the most central actual document in each cluster — the centroid was opted instead. The centroid represents the mean vector of the cluster and serves as a good abstraction of the central theme. A global centroid for the entire collection also exists and is used in some comparative analyses.

Cluster interpretation is primarily keyword-based. The most representative terms in each cluster were analyzed, using the term scores from the centroid to infer the cluster's topic.

Strategy: The TF-IDF score of a term tends to be similar within a cluster and across the whole collection because the cluster is extracted from that same collection. To compute the average TF-IDF of a term across the collection, it was needed to average its value across all documents.

In Section 4.5, our goal is to identify terms whose mean TF-IDF in a cluster significantly differs from the overall collection. For this, using the **absolute difference** was not the most appropriate strategy to compare histograms or term distributions — a more robust method was required.

4.3 Evaluate

To assess the quality of the clustering results, *internal clustering metrics* were used, as the dataset is unlabeled and no ground-truth cluster assignments are available. Since this is an unsupervised setting, external evaluation is not possible.

Primarly, Silhouette Coefficient was used, which captures both cohesion (how close points in the same cluster are) and separation (how distinct clusters are from each other). This metric allowed comparing different clustering solutions and select the most meaningful configuration.

4.4 Preprocessing

The objective was to reduce the number of terms in the document-term matrix. Since, fifty thousand documents were clustered, the overall vocabulary size turned out to be very large.

Lemmatization was applied to group together variations of the same word, such as *virus/viruses* and *disease/diseases*. Stemming was not applied because it can lead to overly aggressive reductions and less interpretable terms.

Takeaways: Preprocessing helped balance the sum of divergences between clusters, improving the quality of the overall clustering structure.

4.5 Labelling Clusters

One of the most challenging aspects of the clustering process was how to label the resulting clusters in a meaningful way. The goal was to describe each cluster by assigning it a representative category label.

The initial approach was to extract the top terms by sorting features with the highest TF-IDF values. In a simple bag-of-words model, these terms were expected to carry the most importance. However, it was observed that the top TF-IDF terms were often too similar across clusters — for example, the term *virus* appears in 7 different clusters (see Figure 13). This redundancy reduces the discriminative power of TF-IDF for meaningful cluster labeling. The first workaround was to remove the most common terms shared across all clusters, but this method was still limited.

To improve this, The term distributions of each cluster were compared to the entire collection using **Kullback-Leibler divergence**. The goal was to identify the terms where the cluster distribution diverges most from the global one — especially in the lower-frequency regions, which often contain more distinctive terms [7].

Firstly, the term histograms for both the collection and each cluster were compared. If the global distribution is smooth (e.g., Gaussian) and a cluster's distribution is shifted, KL divergence highlights where and how they differ. Importantly, it does not treat all differences equally — small changes in low-frequency areas weigh more than small changes in high-frequency areas.

A further challenge was deciding the direction of divergence in the KL formula (Equation 1): whether P represents the global collection and Q the cluster, or vice versa. Since results vary depending on the order, P was chosen as the global distribution [5].

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{1}$$

To conclude, once the documents are assigned to centroids (or labeled), the distance between each cluster's mean and the overall collection mean can be calculated.

Compared to the TF-IDF-based approach, KL-divergence (Figure 14) yields a more diverse set of distinguishing terms across clusters, with significantly fewer terms repeated. This suggests that KL-divergence provides a better signal for uniquely characterizing cluster semantics.

4.6 Questions to Explore

4.6.1 What is the (hypothesized) number of topic clusters? And document clusters in the D train collection?

To better understand how to represent the clusters, an analysis was conducted to determine the optimal number of clusters in the D_{train} collection. Values of k ranging from 5 to 30 were experimented using both K-Means and Agglomerative clustering algorithms. The evaluation was based on three standard metrics: Silhouette Score (higher

is better), Calinski-Harabasz Index (higher is better), and Davies-Bouldin Index (lower is better).
The results for K-Means clustering are summarized in the following table:

k	Silhouette	Calinski-Harabasz	Davies-Bouldin
5	0.0041	175.93	10.4506
10	0.0070	137.52	8.9299
15	0.0118	124.71	7.1836
20	0.0133	112.92	7.2953
25	0.0147	102.27	**6.9589**
30	**0.0156**	93.22	7.1068

Table 5: Evaluation metrics for different values of k using K-Means clustering.

In addition to the numerical evaluation, an elbow plot was generated (Figure 15) by plotting the Sum of Squared Errors (SSE) across increasing values of k. The curve shows a smooth decrease with a visible inflection point around $k = 25$, supporting our previous hypothesis. This suggests that $k = 25$ offers a good trade-off between underfitting and overfitting the clustering structure in the dataset.
Although the highest Silhouette score was obtained at $k = 30$, the Davies-Bouldin Index reaches its minimum at $k = 25$, indicating better separation and compactness between clusters. Taking all metrics into account, we hypothesize that the best number of topic (and document) clusters in the D_{train} collection is $k = 25$.
Agglomerative clustering was also tested and showed similar but slightly worse results across all metrics. Given K-Means' simplicity and overall better performance, it was opted to continue the analysis using K-Means with $k = 25$.

4.6.2 Are the clusters from previous solutions cohesive? And well separated?

To assess the quality of the clusters generated in the previous section, their cohesion was analyzed (how similar documents within a cluster are) and separation (how distinct clusters are from one another). Two clustering methods were evaluated using the Silhouette Score, which considers both intra-cluster similarity (cohesion) and inter-cluster difference (separation).
Using **K-Means with Euclidean distance** (which tends to give more weight to longer documents), a Silhouette Score was obtained of **0.0147** for $k = 30$.
With **Agglomerative Clustering and using cosine similarity and average linkage**, the Silhouette Score was **-0.0012** for $k = 10$ (due to computation limitations, it was not possible to extend the experiment to $k = 30$).
These values suggest that the clusters are not particularly cohesive nor well separated, regardless of the method used. While K-Means performs slightly better, the scores indicate that the document vectors in the D_{train} collection do not naturally form cohesive, well-separated clusters. This may be due to overlapping topics or high dimensionality and sparsity in the dataset.

4.6.3 What does the clustering of topic documents, Q, reveal regarding their conceptual organization and independence? Are there highly similar or overlapping topics?

The dataset used is unlabeled and lack any ground-truth topic annotations, it was not possible to directly assess the conceptual independence or validity of the resulting clusters. Furthermore, the clustering approach did not rely on predefined topic labels, but instead grouped documents based solely on their content similarity.
To investigate potential topic overlap between clusters, the *interpret* method was used. This method highlights the most distinctive terms in each cluster by measuring their divergence from the overall term distribution of the collection using Kullback-Leibler (KL) divergence. By identifying terms that are highly specific to a single cluster, it was possible to gain insights into how conceptually distinct or overlapping different clusters were.
In Figure 18, we show a binary heatmap of KL-divergent terms across all clusters. The presence of shared terms such as *sequence*, *protein*, and *strain* across multiple clusters suggests semantic overlap in those regions of the document space. However, terms like *porcine*, *swine*, and *diarrhea* are uniquely associated with individual clusters, reinforcing their conceptual independence.

4.6.4 Given a specific cluster of topics, check whether the medoid (a sort of prototype topic for the cluster) adequately represents the remaining topics in the given cluster.

To visualize how topics are grouped after clustering, t-SNE was applied over the TF-IDF feature space. Since TF-IDF vectors are typically high-dimensional and sparse, dimensionality was first reduced using **Truncated Singular Value Decomposition (TruncatedSVD)**. This technique projects the data into a 50-dimensional space that preserves most of the variance while minimizing noise, thereby improving the effectiveness of the subsequent t-SNE projection. The resulting 2D embedding, shown in Figure 17, illustrates the spatial distribution of clusters, where each point corresponds to a document and each color represents a different cluster assignment.
Although the medoid is selected as the most representative document within a cluster, analysis suggests that it often fails to fully capture the thematic diversity present across all documents in the cluster.
As a more comprehensive alternative, the use of **Kullback-Leibler (KL) divergence** was adopted to compare each cluster's term distribution against the global collection-wide distribution. This method highlights terms that are most distinctive to a given cluster. Unlike TF-IDF, KL divergence places more emphasis on relative differences, particularly in low-frequency terms, which are often more informative.
Figure 19 presents the most frequently occurring divergent terms across all clusters. While some overlap is

present — for instance, *respiratory* and *patient* appear in three clusters — the majority of top-ranked KL-divergent terms are unique or appear in at most two clusters. This indicates stronger topical separation and greater specificity when compared to TF-IDF-based top terms (discussed in Section 4.6.5).

Therefore, KL divergence provides a more interpretable and fine-grained understanding of each cluster's thematic structure, making it a preferable strategy for cluster interpretation and labeling, especially in cases where the medoid alone is insufficient.

4.6.5 How are the documents in the target collection organized? Briefly discuss the importance of this information to understand the behavior of the target IR system.

To better understand the organization of the D_{train} collection, we used unsupervised clustering with $k = 25$ and interpreted the resulting clusters using KL divergence from a global centroid. Each cluster was then labeled based on the most divergent terms, which highlight the dominant topics in the group. This led to meaningful and interpretable groupings, such as "COVID-19 Cases", "Viral Proteins", "Vaccines", and "PCR Diagnostics". These labels were manually assigned by analyzing the top 10 divergent terms in each cluster.

For example, for cluster 22 the top divergent terms were words like health, public, global, so the label chosen was "Public Health"). As our organization constructed 25 clusters we present some of the chosen in the table 6 below:

Cluster ID	Cluster Label	Top 10 Divergent Terms
0	**Epidemic Models**	model, epidemic, estimate, transmission, network, dynamic, contact, outbreak, number, reproduction
5	**COVID-19 Cases**	covid19, sarscov2, 2019, 2020, case, wuhan, patient, china, february, coronavirus
6	**RNA Replication**	rna, mrna, frameshifting, genome, sequence, replication, synthesis, subgenomic, translation, structure
9	**Pediatric Infections**	child, respiratory, rsv, tract, rhinovirus, asthma, hmpv, infection, syncytial, hbov
10	**SARS-CoV (2003)**	sars, sarscov, severe, syndrome, acute, 2003, respiratory, patient, outbreak, hong
14	**Viral Proteins**	protein, fusion, membrane, domain, glycoprotein, sarscov, peptide, spike, structure, amino
16	**Antivirals**	compound, drug, antiviral, activity, inhibitor, protease, derivative, target, 3clpro, ic50
17	**Vaccines**	vaccine, adjuvant, vector, vaccination, dna, immunogenicity, response, immunization, immune, mucosal
21	**PCR Diagnostics**	detection, assay, pcr, amplification, sample, sensitivity, realtime, rtpcr, test, multiplex
24	**Clinical Patients**	patient, pneumonia, care, hospital, exacerbation, icu, treatment, antibiotic, cap, communityacquired

Table 6: Top 10 Divergent Terms for each cluster.

Additionally, KL divergence enables more nuanced analysis. In cluster 3, Figure 20 shows terms such as *merscov*, *camel*, and *arabia*, clearly pointing to the Middle East Respiratory Syndrome (MERS) topic. Furthermore, Figure 16 displays a co-occurrence network of divergent terms that appear in multiple clusters. This visualization highlights concept overlaps, such as terms like *coronavirus*, *wuhan*, and *china*, shared across clusters related to COVID-19.

5 Page Ranking

5.1 Graph ranking method based on document similarity

The graph-based ranking approach, which leverages document similarity to construct a document graph, was investigated to assess its potential for improving information retrieval performance. Specifically, the goal was to evaluate whether incorporating document centrality through PageRank can outperform the baseline BM25F model.

5.1.1 Graph Construction

The process began with collecting the top 1000 documents for each query using the BM25F model, which was previously established as the best-performing IR model. For each query and its 1000 documents, a similarity graph was constructed. The cosine similarity between each document and every other document was computed, and edges were created if the similarity exceeded a certain threshold (0.5, 0.7, or 0.9). These thresholds allowed the analysis to observe how varying similarity levels affected the ranking outcomes.

Cosine similarity was chosen for its effectiveness in comparing high-dimensional, sparse vectors—typical in textual data of BM25F. Unlike distance-based measures such as Manhattan or Euclidean, cosine similarity captures the orientation between vectors rather than their magnitude, making it better suited for measuring semantic closeness between documents.

5.1.2 Undirected PageRank

The PageRank algorithm was applied to the constructed graph using the equation:

$$\mathrm{PR}(d_i) = \frac{p}{N} + (1-p) \times \sum_{d_j \in \mathrm{Links}(d_i)} \frac{\mathrm{PR}(d_j)}{|\mathrm{Links}(d_j)|}$$

Here, $\mathrm{PR}(d_i)$ is the updated probability of document d_i, and $|\mathrm{Links}(d_j)|$ is the number of outbound links from document d_j. An initial uniform distribution was used, with a damping factor $p = 0.15$. The algorithm ran for 50 iterations, and documents were ranked based on their final probabilities.

5.1.3 Centrality Scores

Upon examining the top-ranked documents for each similarity threshold, it is observed that $\theta = 0.7$ and $\theta = 0.9$ consistently rank the same documents at the top across different queries. In contrast, $\theta = 0.5$ frequently returns

different documents. This behavior suggests that evaluation metrics for the higher thresholds are likely to be more similar to each other than to those obtained with the lower threshold.

5.1.4 Evaluation

The performance was assessed using MAP@1000, NDCG@1000, NDCG@10, and P@10.

Evaluation	BM25F	$\theta = 0.5$	$\theta = 0.7$	$\theta = 0.9$
P@10	0.7280	0.2180	0.2720	0.2680
nDCG@10	0.6765	0.1686	0.2353	0.2471
nDCG@1000	0.3975	0.3166	0.3245	0.3252
MAP@1000	0.2069	0.0955	0.1004	0.1001

Table 7: Evaluation Results for BM25F and PageRank with Different Thresholds

The results showed that PageRank did not outperform BM25F, as BM25F is optimized for content-based ranking by considering term frequency and inverse document frequency. PageRank, on the other hand, focuses on document structure rather than content relevance, making it less effective when BM25F's ranking is already highly optimized. However, higher similarity thresholds (0.7, 0.9) improved results, suggesting that stricter similarity criteria help identify more relevant documents.

5.2 Improved graph ranking method

An additional graph ranking method was explored, computing the PageRank through the following iterative procedure:

$$\mathrm{PR}(d_i) = p \times \frac{\mathrm{Prior}(d_i)}{\sum_{d_j} \mathrm{Prior}(d_j)} + (1-p) \times \sum_{d_j \in \mathrm{Links}(d_i)} \frac{\mathrm{PR}(d_j) \times \mathrm{Weight}(d_j, d_i)}{\sum_{d_k \in \mathrm{Links}(d_j)} \mathrm{Weight}(d_j, d_k)}$$

To enhance the PageRank algorithm, both the original BM25F ranking values and the link weights between documents were incorporated in each iteration, while keeping the number of iterations fixed at 50 and using the highest threshold value of 0.9. However, as shown in Table 8, this adjustment did not lead to improved evaluation performance.

Evaluation	BM25F	Original PR	Improved PR
P@10	0.7280	0.2680	0.2840
nDCG@10	0.6765	0.2471	0.2366
nDCG@1000	0.3975	0.3252	0.3246
MAP@1000	0.2069	0.1001	0.1019

Table 8: Evaluation Results for BM25F and both PageRanks

The results are very similar to those of the original PageRank, reinforcing the conclusion that BM25F outperforms both PageRank approaches.

6 Conclusion

The clustering analysis performed on the D_{train} collection led to a segmentation of documents into 25 groups, selected based on internal evaluation metrics and an elbow analysis over SSE. While clustering performance was limited due to the high dimensionality and sparsity of TF-IDF vectors, K-Means with $k = 25$ offered the most balanced trade-off.

To label and interpret each cluster, Kullback-Leibler (KL) divergence was used to compare the cluster-specific term distribution with the global distribution. This approach yielded more informative and distinct keywords than TF-IDF alone, leading to more meaningful cluster labels.

t-SNE projections showed partial topical separation but confirmed that some clusters had overlapping semantic content. Medoids were not always good representations of clusters; instead, KL-divergent terms provided a more interpretable summary of the cluster's theme.

The graph-based PageRank approach did not outperform the BM25F baseline, which already provides strong content-based relevance by leveraging term frequency and inverse document frequency. While higher similarity thresholds (θ=0.7 and θ=0.5) slightly improved PageRank's performance, the results remained well below those of BM25F. Even with enhancements such as incorporating prior BM25F scores and edge weights, the improved PageRank showed no significant gains, suggesting that BM25F's content-aware ranking leaves little room for improvement through graph-based re-ranking alone.

References

[1] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. "Reciprocal rank fusion outperforms condorcet and individual rank learning methods". In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 758–759. ISBN: 9781605584836. DOI: 10.1145/1571941.1572114. URL: https://doi.org/10.1145/1571941.1572114.

[2] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Available online. Pearson Education (free PDF version), 2015.

[3] Xiaoyan Li and W. Bruce Croft. "Time-based language models". In: *Proceedings of the Twelfth International Conference on Information and Knowledge Management*. CIKM '03. New Orleans, LA, USA: Association for Computing Machinery, 2003, pp. 469–475. ISBN: 1581137230. DOI: 10.1145/956863.956951. URL: https://doi.org/10.1145/956863.956951.

[4] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[5] SciPy Developers. *scipy.special.rel_entr*. Accessed: 2024-04-02. 2024. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.rel_entr.html.

[6] Christophe Van Gysel and Maarten de Rijke. "Pytrec$_e$val : $AnExtremelyFastPythonInterfacetotrec_eval$". In: *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*. SIGIR '18. ACM, June 2018. DOI: 10.1145/3209978.3210065. URL: http://dx.doi.org/10.1145/3209978.3210065.

[7] Wikipedia contributors. *Kullback–Leibler divergence*. Accessed: 2024-04-02. 2024. URL: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence.

Appendices

	title_length	abstract_length
count	192509	192509
mean	12.609	133.038
std	5.935	133.038
min	0	0
max	146	18000

Figure 7: Summary statistics of title and abstract lengths.

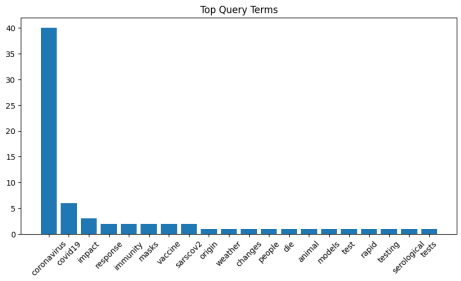

Figure 8: Top Query Terms.

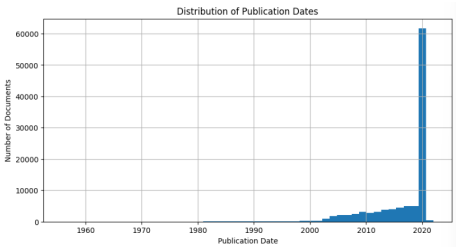

Figure 9: Document Publication Date Distribution.

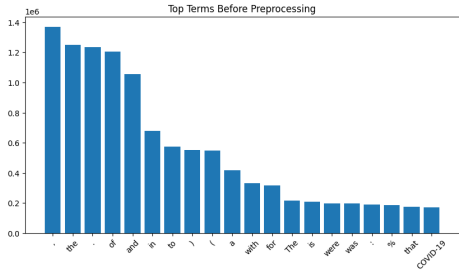

Figure 10: Top Terms Before Processing.

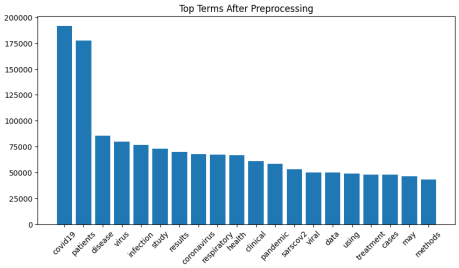

Figure 11: Top Terms After Processing.

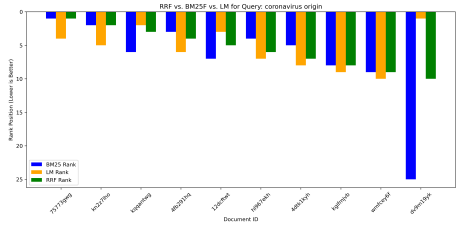

Figure 12: Comparison of ranking between IR models for one query.

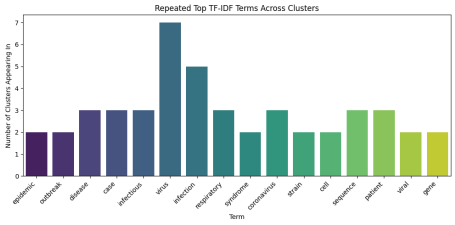

Figure 13: Repeated top TF-IDF Terms across all clusters.

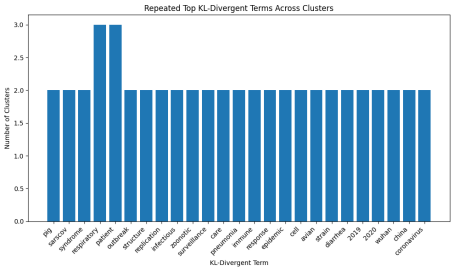

Figure 14: Repeated top KL Terms across all clusters.

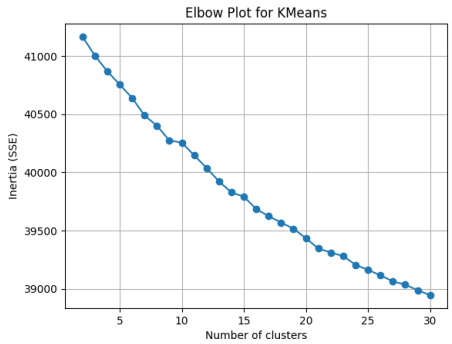

Figure 15: Elbow plot showing inertia (SSE) for K-Means clustering with different values of k.

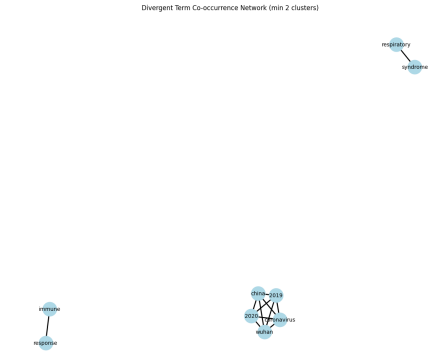

Figure 16: Co-occurrence network of shared divergent terms across at least 2 clusters.

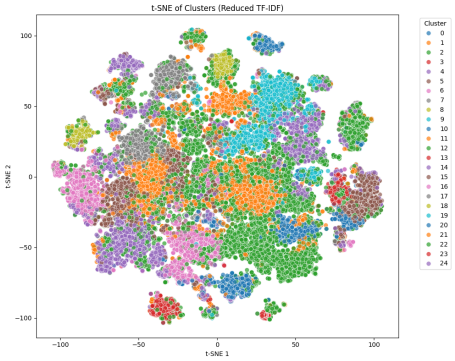

Figure 17: Cluster representation using t-SNE.

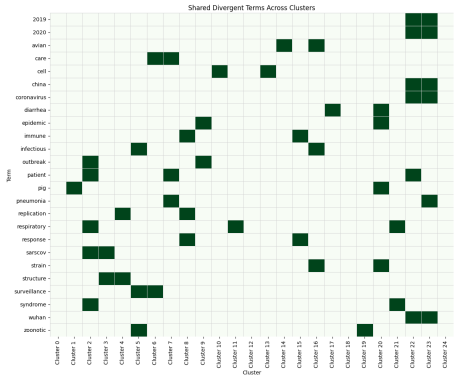

Figure 18: Shared terms across clusters.

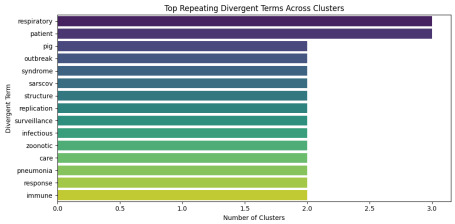

Figure 19: Most common Divergent Terms across clusters.

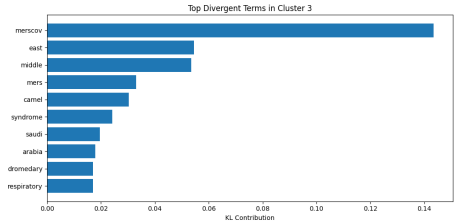

Figure 20: Cluster 3 terms divergence.