
AN INFORMATION RETRIEVAL PROJECT
HEALTH
net

DOMAIN.

🏥 *Healthnet aims to develop a search engine capable of retrieving accurate responses to natural language queries related to health.*

🏥 *By leveraging PyTerrier, the system indexes medical documents and evaluates its performance using NFCorpus, a benchmark dataset for medical information retrieval.*

🏥 *The corpus includes relevance judgments for 169,756 query-document pairs.*

DOMAIN.

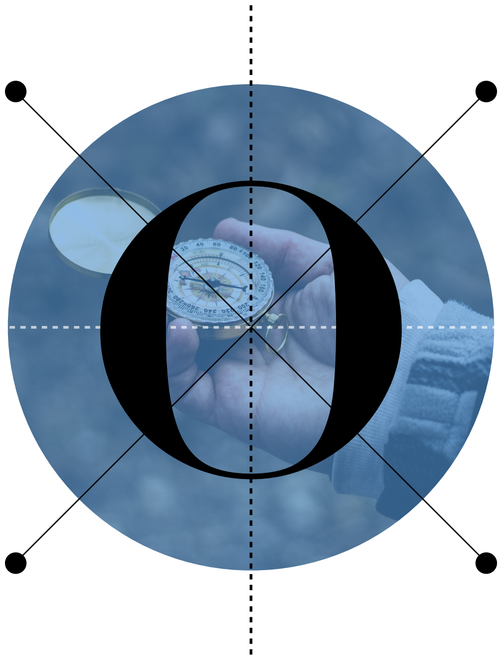
🏥 *The primary goal is to bridge the semantic gap between the natural language used in user queries and the technical language of medical documents. This involves addressing challenges such as vocabulary mismatch and ensuring meaningful ranking.*

🏥 *This can be seen as a mapping task: translating user questions into complex query representations that align with the vocabulary and semantics of medical documents.*

OBJECTIVES

Dataset Analysis

We want to analyze the dataset and its components, trying to extract interesting insights from it.



Retrieval Pipelines

We will describe the indexing process, retrieval models, and evaluation of different retrieval pipelines.

Improving Performance

We want to explore techniques to improve the retrieval performance of the pipelines we built.

OBJECTIVES

Dataset Analysis

The data is organized into three main DataFrames with specific roles and dimensions:

The **data** DataFrame (3633x3) represents the document collection, where each row corresponds to a document identified by `doc_id`, containing its `text` and `title`.

	text	title	doc_id
1911	Smoking is an established risk factor for panc...	Cigarette Smoking and Pancreatic Cancer: A Poo...	MED-3318
2847	The cytotoxic effects of Triphala (TPL), an In...	Cytotoxic response of breast cancer cell lines...	MED-4532
2142	Using the infrastructure of the National Atmos...	Wet deposition of fission-product isotopes to ...	MED-3635
666	Ciguatera is an important form of human poison...	Ciguatera: recent advances but the risk remains.	MED-1570
1702	BACKGROUND: Elevated levels of lipids, such as...	Strawberry modulates LDL oxidation and postpra...	MED-2972
2347	BACKGROUND: Mammalian lignans, enterolactone (...	Effect of mammalian lignans on the growth of p...	MED-3875
488	This paper documents the recent (1976-1995) li...	The acute effects of exercise on mood state.	MED-1358
1461	There are now extensive scientific data sugges...	Prostate cancer and inositol hexaphosphate: ef...	MED-2579
593	PURPOSE: Plant-based nutrition achieved corona...	A way to reverse CAD?	MED-1489
2955	Endemic cretinism includes two syndromes: a mo...	Cretinism revisited.	MED-4668
3265	AV119 is a patented blend of two sugars from a...	Effects of AV119, a natural sugar from avocado...	MED-5011
1502	Over the last 40 years there have been constan...	Inadvertent exposure to xenoestrogens.	MED-2650
1925	Background: Strategies are needed to increase ...	Hiding vegetables to reduce energy density: an...	MED-3369
2422	Various specific and non-specific environmenta...	Immune potentiation of ultrafine dietary parti...	MED-3970
1808	Anecdotal, survey, and epidemiological data su...	Is infection risk linked to exercise workload?	MED-3154
142	INTRODUCTION: Vegetarian diets are considered ...	The effect of vegetarian diet on selected esse...	MED-922
782	There are currently approximately 33.9 million...	The Projected Impact of Risk Factor Reduction ...	MED-1703
630	The purpose of this study was to determine whe...	The effect of inhaling peppermint odor and eth...	MED-1526
1616	OBJECTIVE: The objective of this study was to ...	Efficacy and safety of Curcuma domestica extra...	MED-2802
3358	The effectiveness of high-temperature, short h...	Effective Heat Inactivation of Mycobacterium a...	MED-5108

OBJECTIVES

Dataset Analysis

The data is organized into three main DataFrames with specific roles and dimensions:

The **queries** DataFrame (323x2) represents input queries,
with each row including a unique `query_id` and the query's `text`.

	query_text	query_id
42	accidents	PLAIN-478
74	canker sores	PLAIN-817
156	norovirus	PLAIN-1731
154	neurocysticercosis	PLAIN-1710
151	myelopathy	PLAIN-1679
180	rapamycin	PLAIN-1983
243	Turmeric Curcumin and Osteoarthritis	PLAIN-2650
158	okra	PLAIN-1752
217	whiting	PLAIN-2375
8	Chronic Headaches and Pork Parasites	PLAIN-91
202	tempeh	PLAIN-2220
3	What's Driving America's Obesity Problem?	PLAIN-33
245	Is Caramel Color Carcinogenic?	PLAIN-2670
304	Are Multivitamins Good For You?	PLAIN-3292
31	Dioxins Stored in Our Own Fat May Increase Dia...	PLAIN-344
165	Peoria	PLAIN-1827

OBJECTIVES

Dataset Analysis

The data is organized into three main DataFrames with specific roles and dimensions:

The **qrels_df** DataFrame (12334x3) contains relevance judgments linking `query_id` to `doc_id`, with a `relevance` score indicating how well a document matches a query.

	query_id	doc_id	relevance
7051	PLAIN-1805	MED-2166	1
8785	PLAIN-2061	MED-3621	1
3530	PLAIN-966	MED-4853	1
1688	PLAIN-531	MED-4536	1
11728	PLAIN-3251	MED-4095	1
4300	PLAIN-1288	MED-2449	1
9647	PLAIN-2281	MED-2098	1
9390	PLAIN-2102	MED-4466	1
1953	PLAIN-660	MED-2799	1
4689	PLAIN-1409	MED-1765	1
6282	PLAIN-1635	MED-4941	1
2301	PLAIN-660	MED-3583	1
3218	PLAIN-934	MED-1645	1

OBJECTIVES

Dataset Analysis

Preliminary analysis of the **qrels_df** dataset reveals that relevance labels are limited to two values: 1️⃣ (more frequent) and 2️⃣ (less frequent).

The dataset contains 12,334 explicit query-document relationships, a small fraction of the 1,173,459 possible pairs, implying most pairs are implicitly non-relevant (relevance value 0).

These labels can be interpreted as follows:

0️⃣ (not relevant)

1️⃣ (slightly relevant)

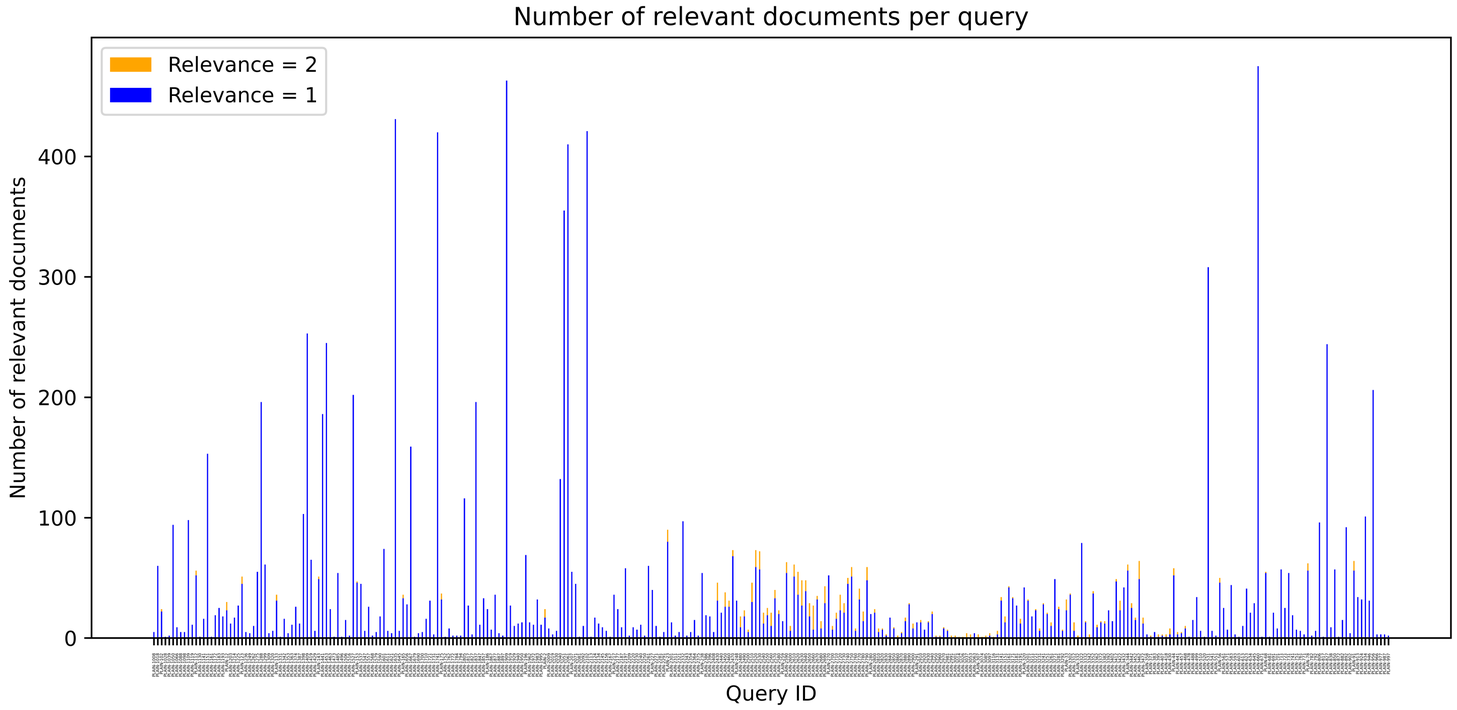
2️⃣ (highly relevant)

OBJECTIVES

Dataset Analysis

The analysis of **number of relevant documents per query** reveals an uneven dataset, with some queries covering many relevant documents and others very few, making it challenging for retrieval algorithms to balance these differences.

Moderately relevant documents (1️⃣) dominate the results, while highly relevant ones (2️⃣) are rarer. This imbalance impacts some evaluation metrics, as metrics like **Precision@K** penalise systems that retrieve many moderately relevant documents. **The prevalence of Relevance = 1️⃣** underscores the need to fine-tune algorithms to prioritise highly relevant documents, in order to try and enhance retrieval quality.



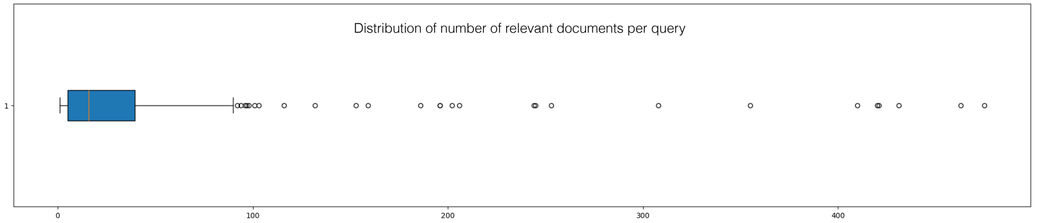
OBJECTIVES

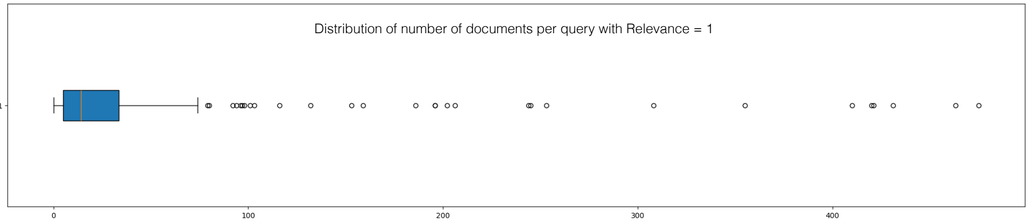
Dataset Analysis

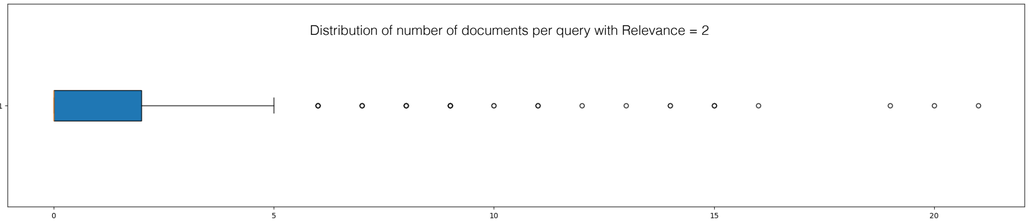
The box-plots reveal a clear hierarchy in document relevance. Queries with many relevant documents vary, with an average of 38.18 documents, a median of 16 and an IQR of 34.5. Outliers, defined as values above 91.25, include 27 queries, causing a right-skewed pattern.

Documents with a relevance of 1 are more evenly distributed but still skewed, with an average of 36.40, a median of 14, and an IQR of 28.5. Outliers occur above 76.25, with 29 queries falling into this range, reflecting slightly reduced variability.

In contrast, documents with a relevance of 2 are rare, with an average of 1.78, a median of 0, and an IQR of just 2. Most queries have no such documents, and only 36 queries exceed the outlier threshold of 5, underscoring the scarcity of highly relevant documents. Overall, highly relevant documents are scarce, and lower relevance documents dominate the dataset.



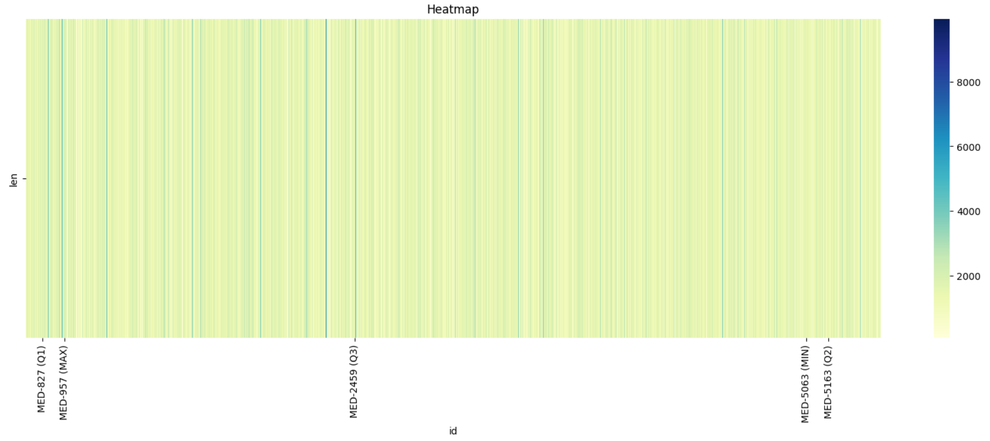




OBJECTIVES

Dataset Analysis

Document lengths show wide variation, with most falling in a moderate range, though a few very long outliers create a skewed distribution. The average document length is 1496.9 characters, with a median of 1517. Lengths range from a minimum of 90 to a maximum of 9939 characters, with 50% of the data concentrated between 1185 and 1769 characters. Notably, 56 documents exceed 2645 characters, and 17 fall below 309, totaling 73 outliers.



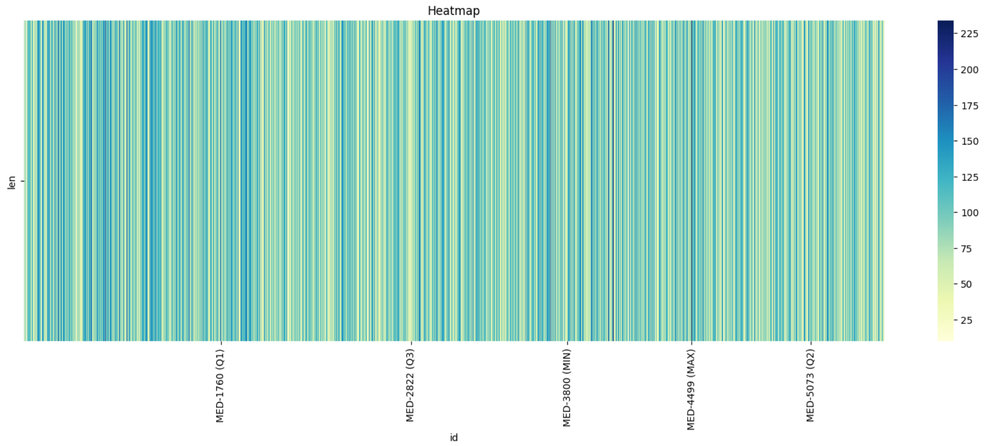
OBJECTIVES

Dataset Analysis

OBJECTIVES

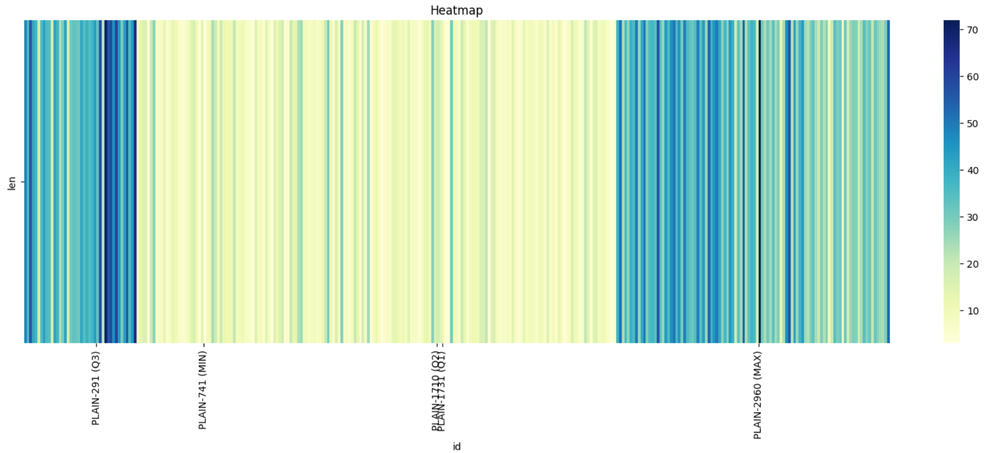
OBJECTIVES

Titles are generally short and compact, but a small number of longer ones add slight asymmetry. Most titles range between 68 and 117 characters, with a mean of 92.87 and a median of 91. The minimum title length is 10, and the maximum is 234. Eight outliers exceed 190 characters, further emphasising the variability.



Dataset Analysis

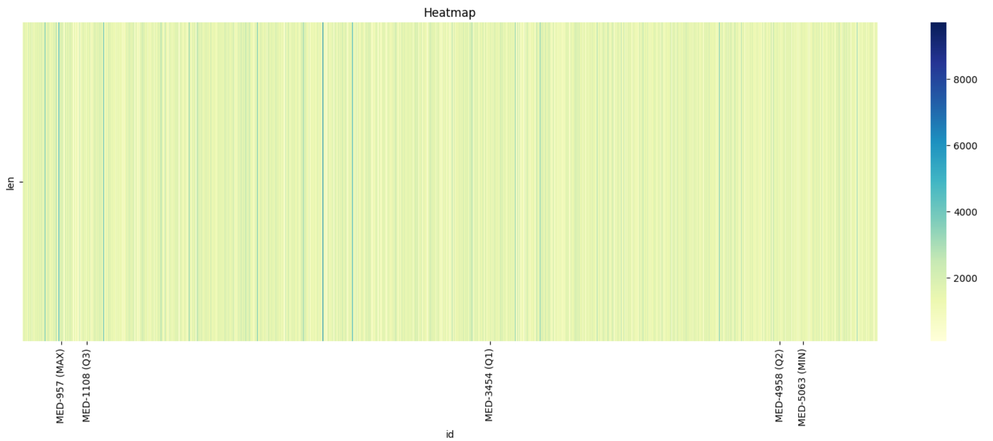
Queries are typically brief and concentrated, with occasional longer ones slightly skewing the average. Most queries range between 9 and 33 characters, with a mean of 21.76 and a median of 18. Lengths vary from a minimum of 3 to a maximum of 72, with two outliers exceeding 69 characters.

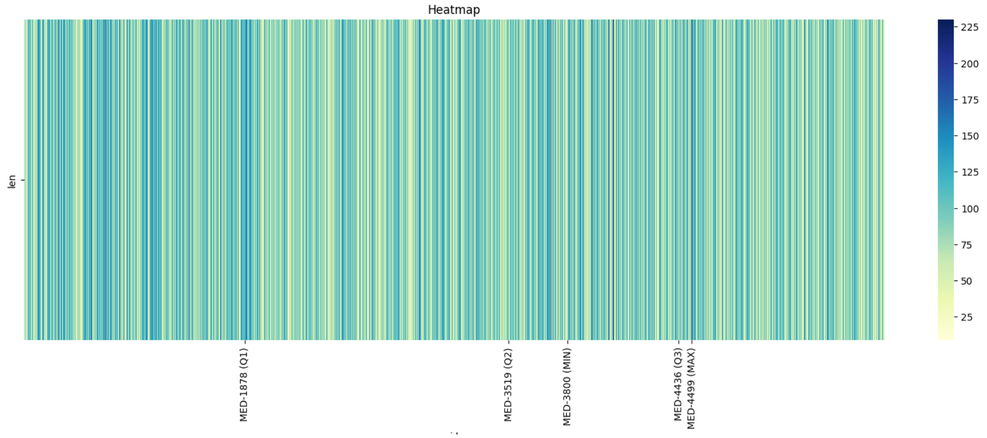


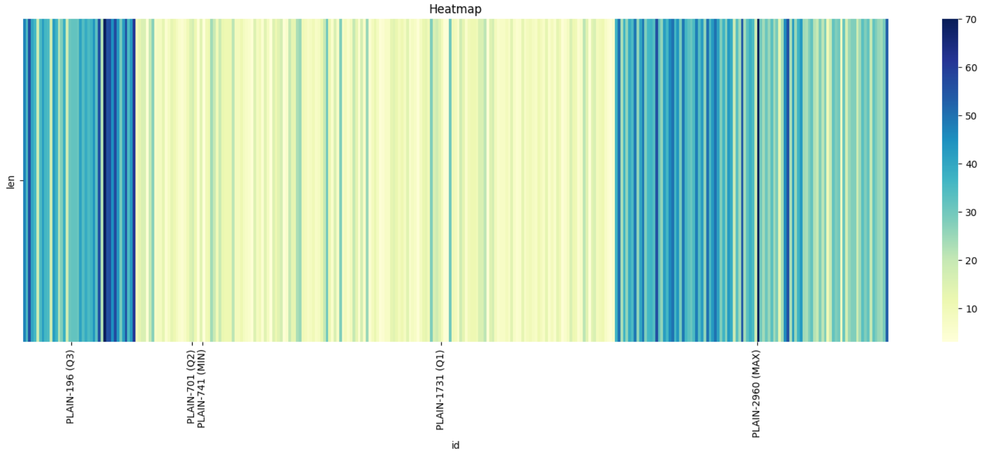
OBJECTIVES

Dataset Analysis

Normalization improves text consistency by removing URLs, emojis, HTML tags, extra spaces, and non-alphabetic characters, while preserving numbers. This reduces noise, aiding machine learning models. Its impact was minimal on titles and queries, but for document texts, it slightly shortened lengths and increased outliers. Overall, normalization made text more uniform, with subtle changes in distribution.







OBJECTIVES

Dataset Analysis

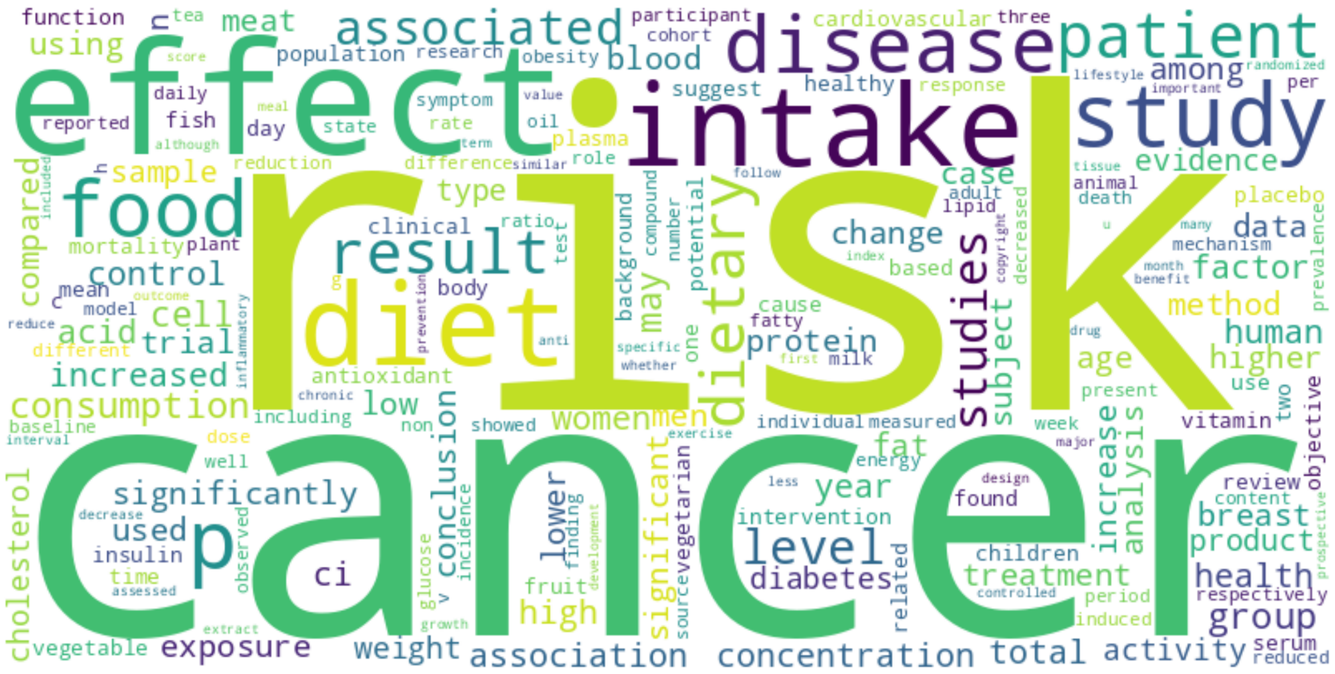
Tokenisation breaks text into smaller units, or "tokens," such as words or symbols, to simplify text analysis. In this case, tokenisation is applied to normalised datasets (text, titles, and queries) using the word_tokenise function from NLTK. This function splits text by spaces and punctuation, following rules to account for language-specific nuances. The corpus contains 31,881 unique tokens and 872,028 total tokens, showing significant repetition.

OBJECTIVES

```
Large Language Models (LLMs), such as GPT-3 and GPT-4, util
ize a process called tokenization. Tokenization involves br
eaking down text into smaller units, known as tokens, which
the model can process and understand. These tokens can rang
e from individual characters to entire words or even larger
chunks, depending on the model. For GPT-3 and GPT-4, a Byte
Pair Encoding (BPE) tokenizer is used. BPE is a subword tok
enization technique that allows the model to dynamically bu
ild a vocabulary during training, efficiently representing
common words and word fragments. Although the core tokeniza
tion process remains similar across different versions of t
hese models, the specific implementation can vary based on
the model's architecture and training objectives.
```

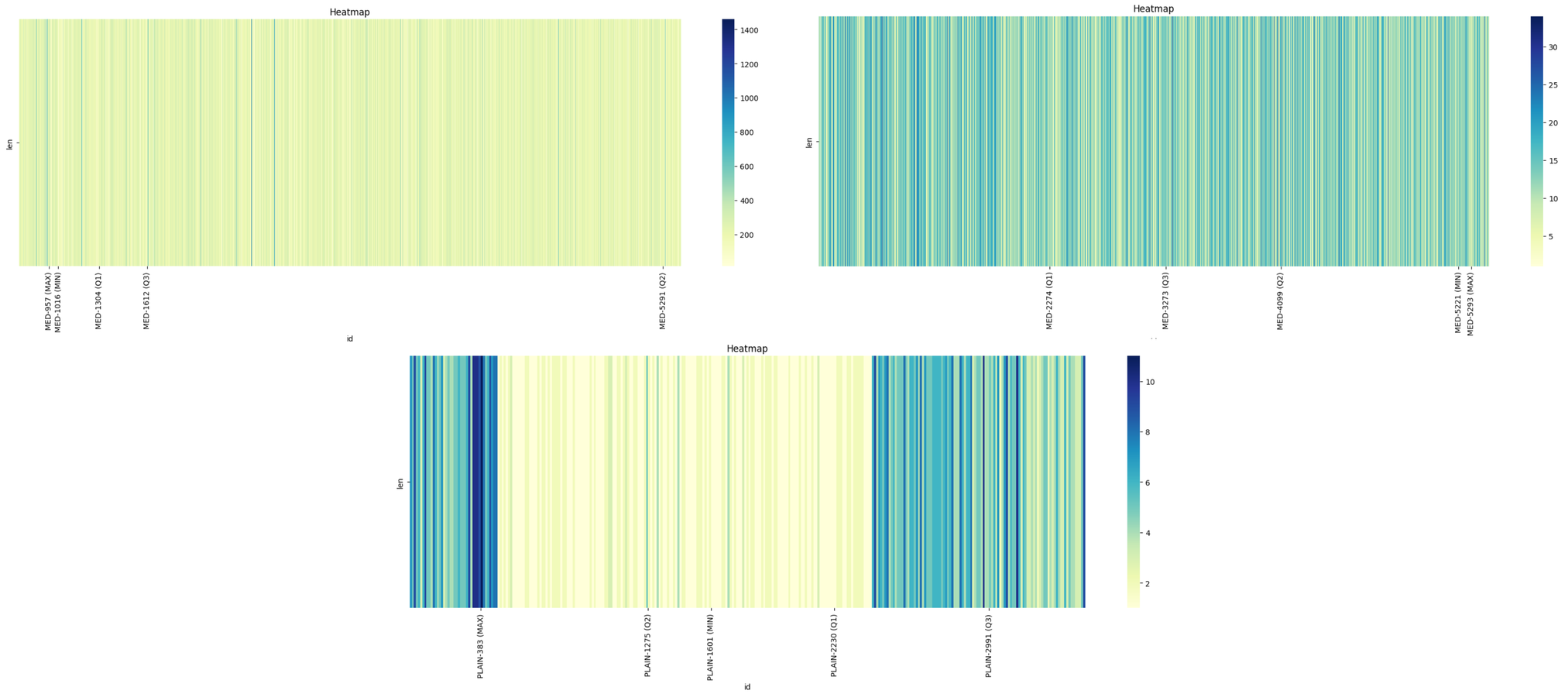
Dataset Analysis

The word-clouds highlight terms like "cancer", "risk", "treatment", and "effect", indicating a focus on treatment impacts. Secondary words such as "study", "patient", and "result" suggest a focus on research. The raw-token word-cloud reveals word frequency but also shows redundancy from morphological variations.



Dataset Analysis

Token distribution is similar to normalised text lengths, with tokenisation increasing outliers in documents (especially long ones) and slightly reducing them in titles. In queries, tokenisation removes outliers entirely. On average, tokens contain 6.5-7.0 characters, indicating they represent longer text segments rather than short elements.

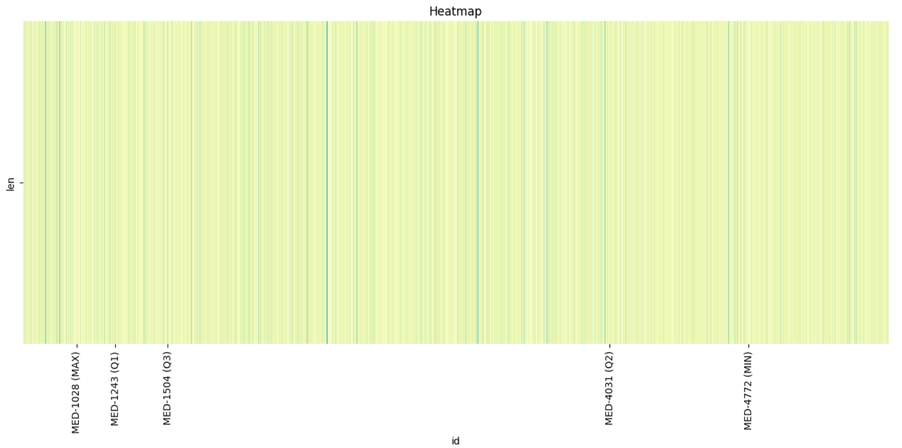


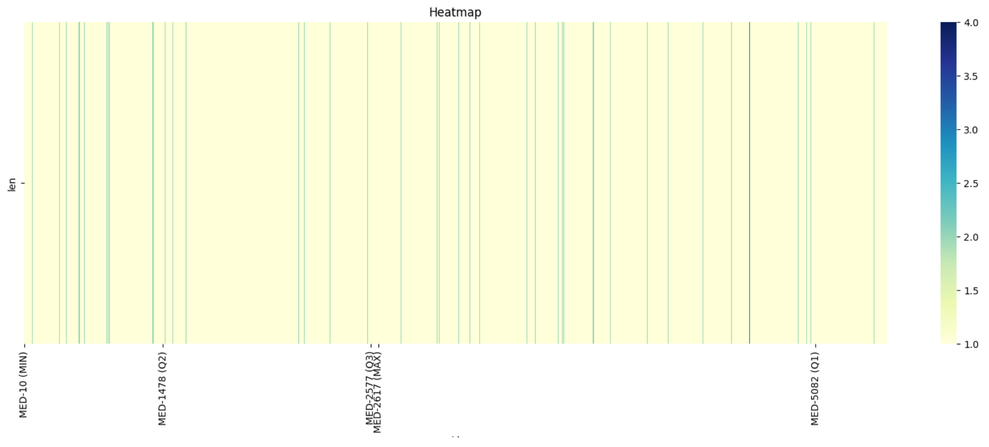
OBJECTIVES

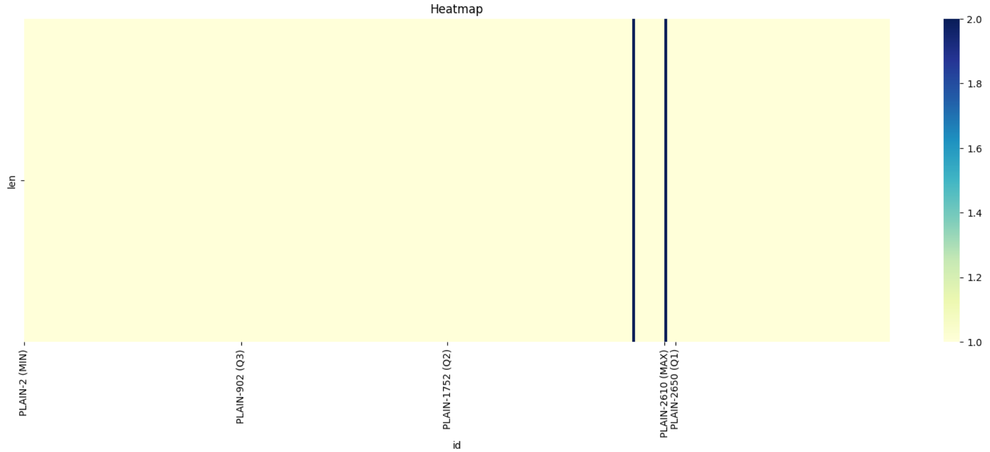
Dataset Analysis

Sentence tokenisation breaks text into sentences, helping analyse meaning, context, and relationships. Document lengths vary widely, with a median of 10 sentences, reflecting a mix of short and long texts. Titles are typically short, but longer ones appear in academic contexts. Queries are usually brief, though some outliers occur due to tokenisation errors.

OBJECTIVES



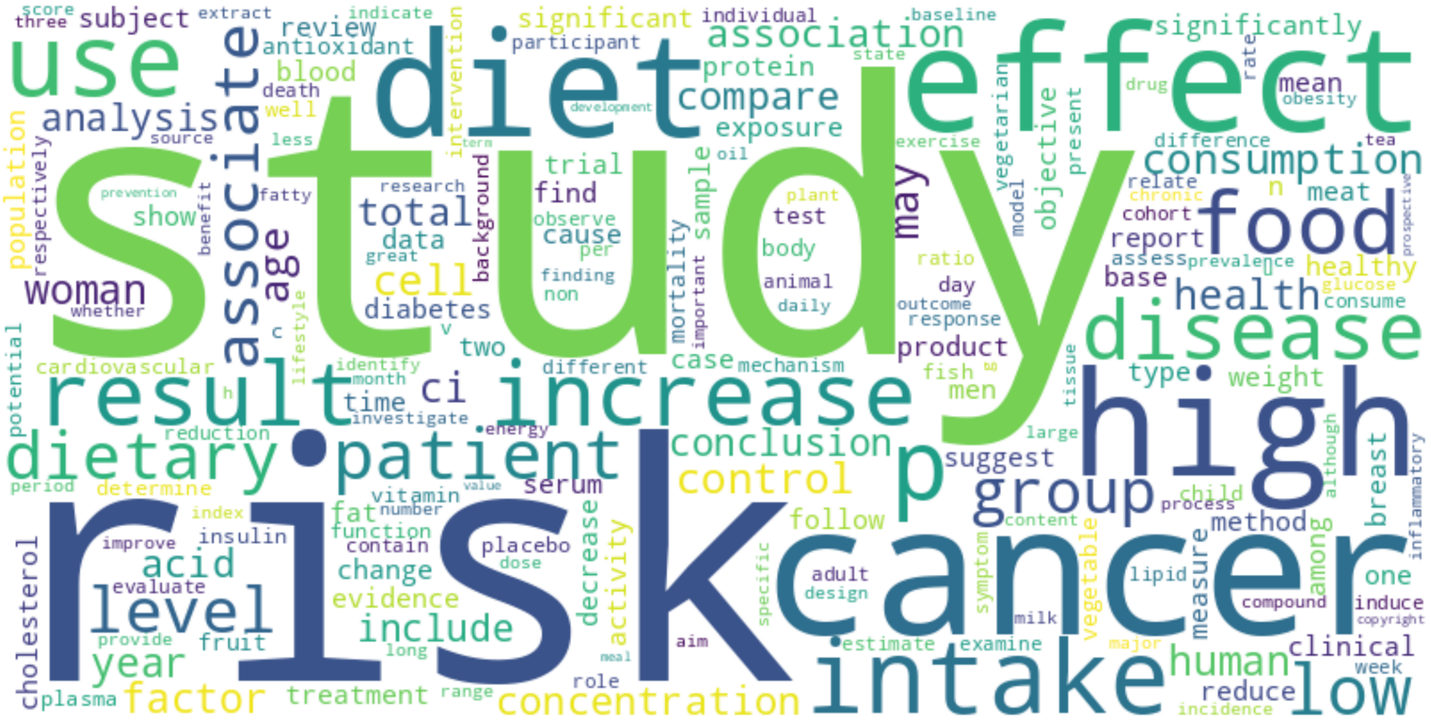




OBJECTIVES

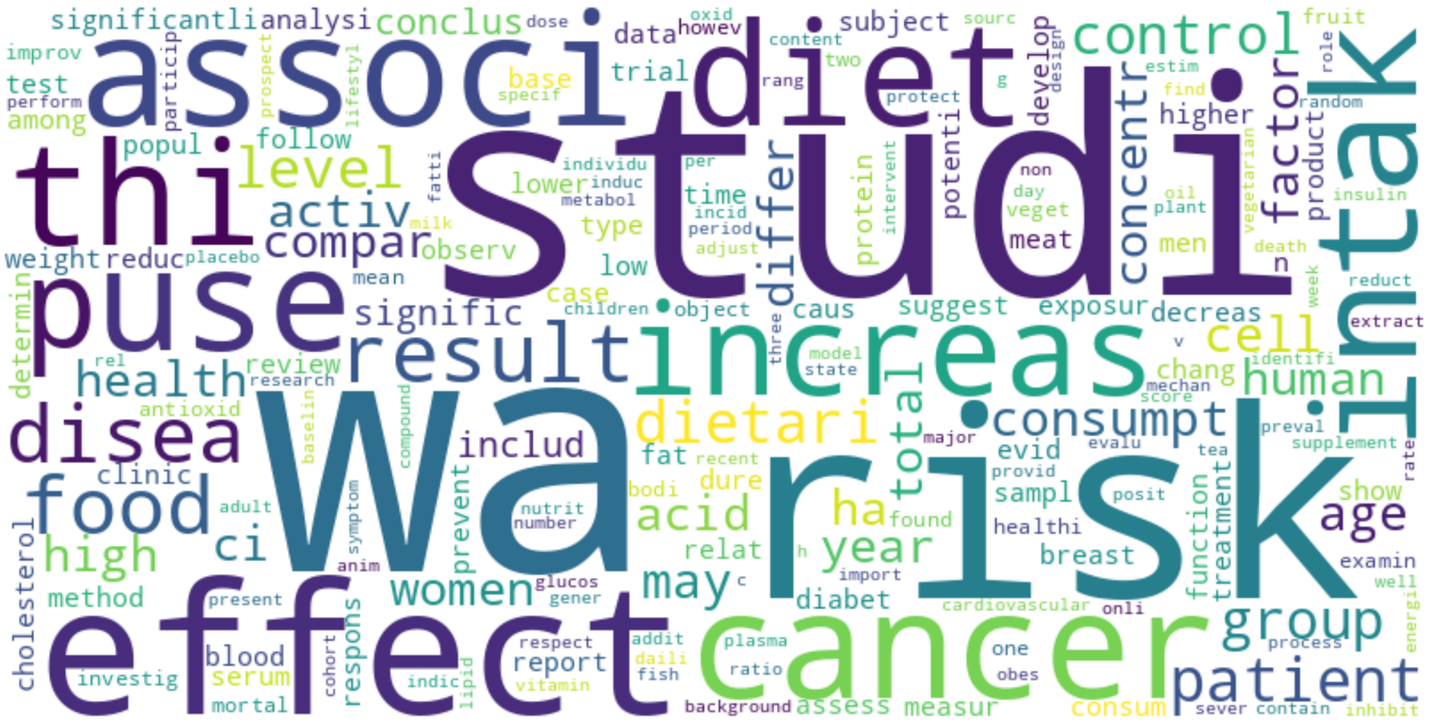
Dataset Analysis

Lemmatisation simplifies words by reducing them to their base forms, reducing linguistic variability, and improving text analysis by treating words with similar meanings as the same entity. After lemmatisation, the corpus consists of 28,786 unique lemmas, showing a significant reduction in word variants. This unification helps focus on core vocabulary, making analysis more concentrated. The lemmatised word cloud is clearer.



Dataset Analysis

Stemming goes a step further by reducing words to their root forms, such as turning "associated" and "associating" into "associ." The corpus contains 24,480 unique stems, which further simplifies the vocabulary, though at the cost of nuance. The resulting word cloud is even more simplified, showing roots like "studi" and “intak." While stemming reduces dimensionality, it can hinder interpretation due to the loss of context.



OBJECTIVES

Dataset Analysis

The analysis of tokens, lemmas, and stems shows a moderate but noticeable reduction in unique elements as we move from tokens to lemmas and then to stems. Tokens have the widest range, while lemmas and stems show slight reductions. This compression of unique elements is more evident in documents with larger vocabularies, though variability remains high. For titles and queries, the reduction has minimal impact due to their short length.

OBJECTIVES

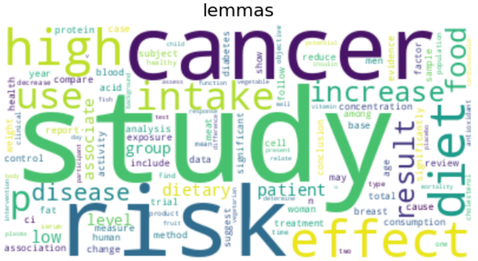
Dataset Analysis

Removing stop words streamlines text by eliminating common, insignificant words, improving the efficiency and accuracy of NLP models. Removing stop-words reduced the token count in documents by 35%, and similar reductions were seen in titles and queries. The removal had minimal impact on unique tokens but increased clarity in word clouds, making key concepts like "risk" and “study" more prominent. This process helps better highlight the main themes in the text.

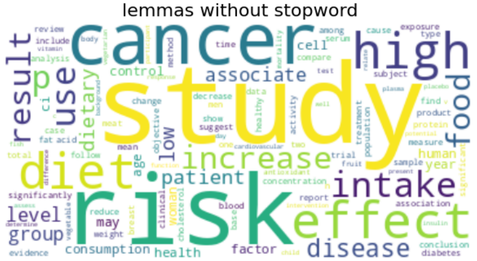
tokens

tokens without stopword

lemmas



lemmas without stopword



stems

stems without stopword

OBJECTIVES

OBJECTIVES

Dataset Analysis

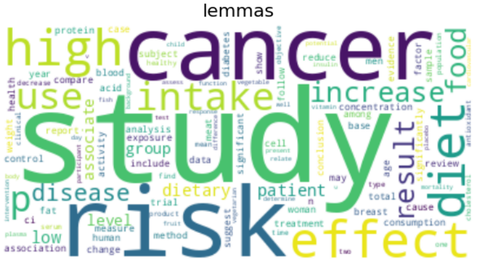
Removing stop words streamlines text by eliminating common, insignificant words, improving the efficiency and accuracy of NLP models. Removing stop-words reduced the token count in documents by 35%, and similar reductions were seen in titles and queries. The removal had minimal impact on unique tokens but increased clarity in word clouds, making key concepts like "risk" and “study" more prominent. This process helps better highlight the main themes in the text.

Note:
The WordCloud library already removes some of the most frequent stop words from images.

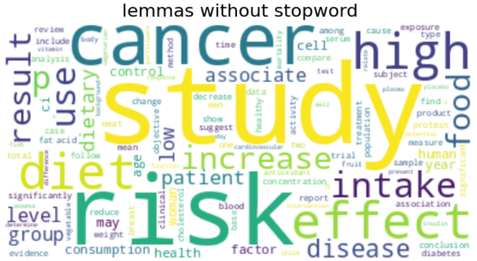
tokens

tokens without stopword

lemmas



lemmas without stopword



stems

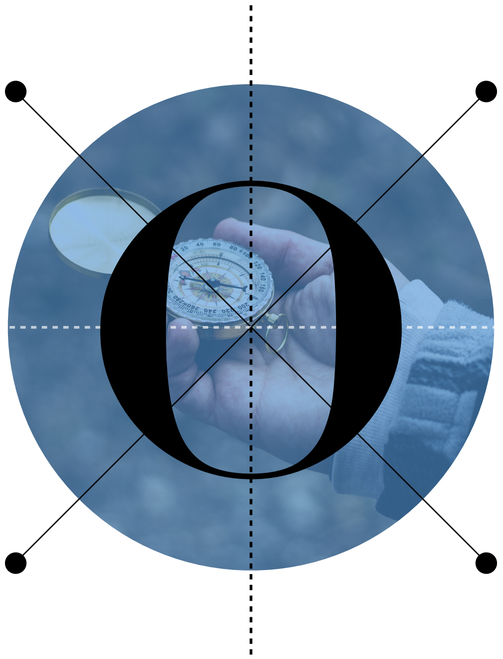
stems without stopword

OBJECTIVES

OBJECTIVES

Dataset Analysis

We want to analyze the dataset and its components, trying to extract interesting insights from it.



Retrieval Pipelines

We will describe the indexing process, retrieval models, and evaluation of different retrieval pipelines.

Improving Performance

We want to explore techniques to improve the retrieval performance of the pipelines we built.

OBJECTIVES

Retrieval Pipelines

OBJECTIVES

After defining several helper functions, four stemmers were analysed: the EnglishStemmer, a fast and simple tool for basic word reduction; the LancasterStemmer, which is more aggressive and can sometimes oversimplify; the KrovetzStemmer, less aggressive and more accurate thanks to its dictionary-based approach; and the PorterStemmer, widely used for its balance between simplicity and accuracy. The analysis showed that, in this specific context, all stemmers produced the same stems, indicating minimal differences in performance. However, results may vary with different datasets or contexts.

	Stemmer	Number of tokens/stems	Number of unique stems
0	EnglishStemmer	513630	20248
1	LancasterStemmer	513630	20248
2	KrovetzStemmer	513630	20248
3	PorterStemmer	513630	20248

OBJECTIVES

Retrieval Pipelines

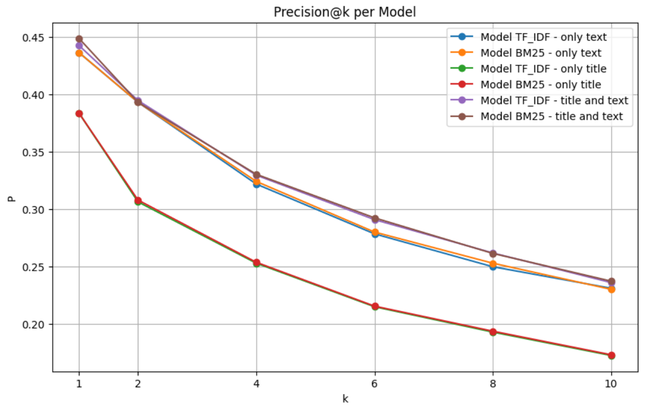
Two types of indexing are used in the analysis: **indexDoc,** which indexes based on document content, and **indexTitle,** which focuses on document titles, and **indexFull,** which considers both titles and texts in one string**.** The methods are kept separate for modularity and flexibility, with metadata storing unused document data for future analysis. In Pipeline 1, a retriever uses the indexDoc index and TF-IDF weighting to retrieve documents, considering term frequency and document importance. Pipeline 2 swaps TF-IDF for BM25, which also factors in term frequency but normalises for very frequent or rare words. Pipelines 3 and 4 are similar to the first two, but they focus on indexing titles instead of content. Pipeline 5 explores using titles plus text. Pipeline 6 does the same with BM25.

OBJECTIVES

Retrieval Pipelines

	name	map	ndcg	ndcg_cut.1	ndcg_cut.10	P.1	P.10	recall.1	recall.10
0	TF_IDF - only text	0.148	0.299	0.421	0.321	0.437	0.233	0.057	0.148
1	BM25 - only text	0.148	0.299	0.421	0.320	0.437	0.233	0.058	0.147
2	TF_IDF - only title	0.101	0.208	0.365	0.248	0.381	0.173	0.050	0.112
3	BM25 - only title	0.101	0.208	0.364	0.249	0.381	0.174	0.051	0.112
4	TF_IDF - title and text	0.151	0.303	0.426	0.325	0.440	0.237	0.057	0.152
5	BM25 - title and text	0.151	0.302	0.423	0.326	0.437	0.238	0.057	0.152

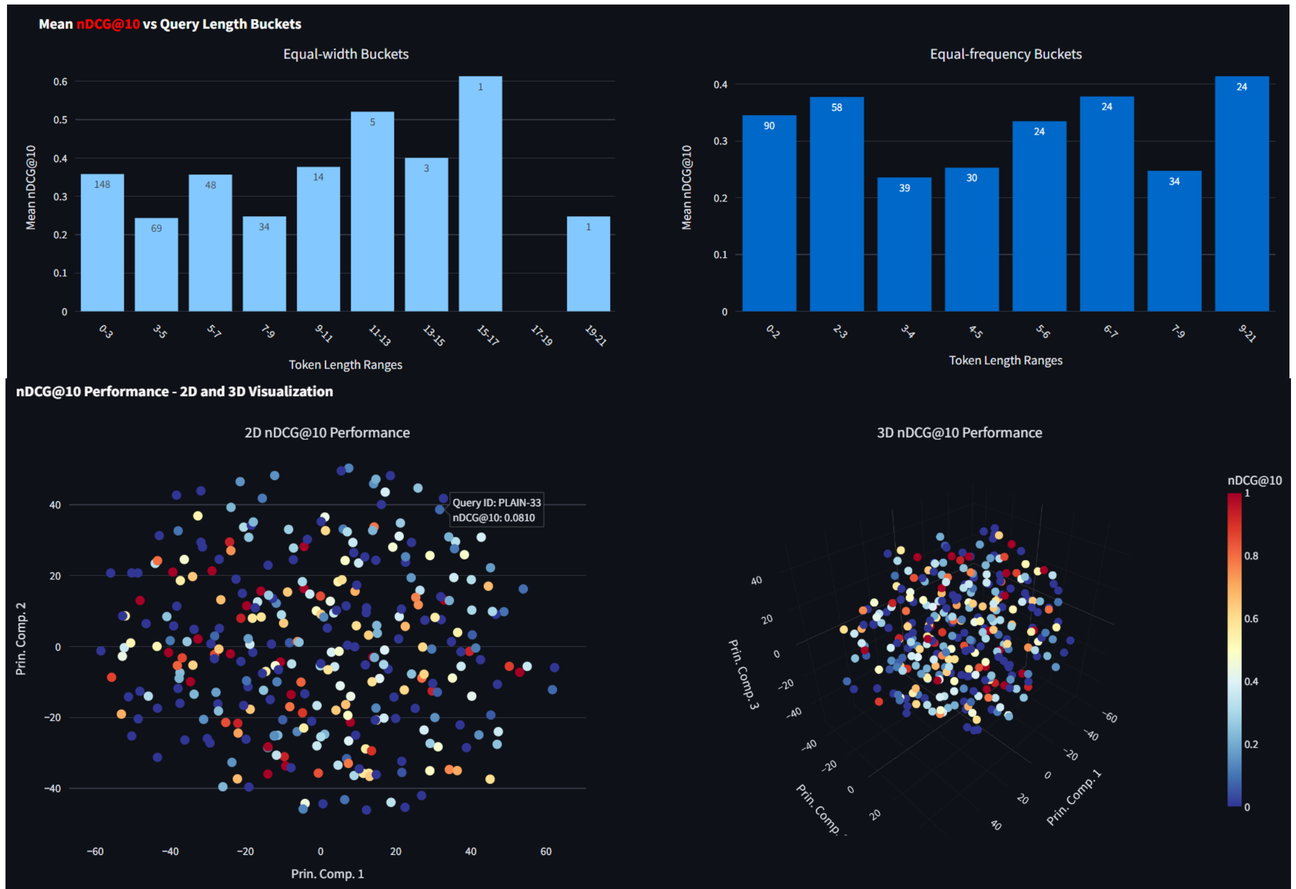
The evaluation shows that BM25 slightly outperforms TF-IDF across various metrics. BM25 achieves better precision, particularly in the "title and text" configuration, with a higher P.10 of 0.443. Recall values are similar for both models, with BM25 slightly ahead. Precision decreases as more results are retrieved, but BM25 still outperforms TF-IDF. NDCG shows similar trends, with BM25 consistently leading. The best results occur when both title and text are used, while "title only" performs the worst. Overall, BM25 is the more effective model, especially for applications prioritising top results.



OBJECTIVES

Retrieval Pipelines

Low-performing queries are often short, making matching difficult, especially with few relevant documents. Longer queries offer more matching opportunities. A challenge is when relevant documents lack query terms or stems, reducing the effectiveness of methods like TF-IDF and BM25. Improving performance requires better lexical alignment, query expansion, and advanced techniques like fuzzy matching or dense models. Despite model variations, the issues remain, with minimal performance improvement between models.

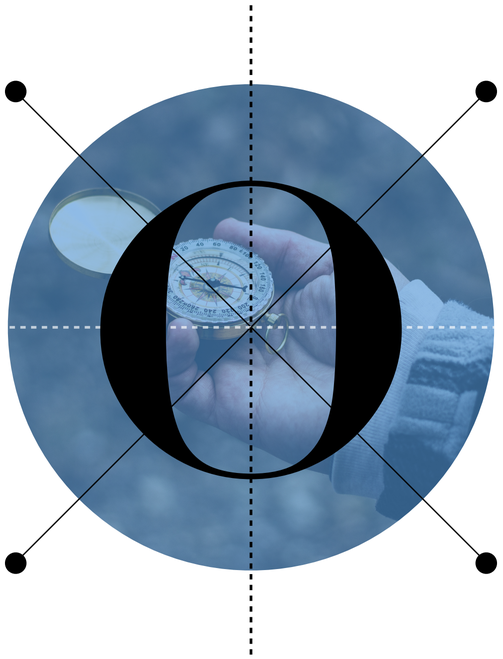


OBJECTIVES

OBJECTIVES

Dataset Analysis

We want to analyze the dataset and its components, trying to extract interesting insights from it.



Retrieval Pipelines

We will describe the indexing process, retrieval models, and evaluation of different retrieval pipelines.

Improving Performance

We want to explore techniques to improve the retrieval performance of the pipelines we built.

OBJECTIVES

Improving Performance

To enhance the model's performance, we implemented query expansion. This approach helps the system interpret user intent more effectively, broadening the range of relevant results and addressing issues with ambiguous, overly specific, or short queries. The goal is to improve precision by filtering irrelevant results and recall by capturing more relevant ones.

Using materials from "Lab 4," we processed a set of queries through the **Qwen** language model, which, guided by a tailored prompt, generated expanded versions of each query. These expansions were designed to reflect semantic and linguistic variations, boosting the system's accuracy. Two integration options are proposed: merging all expansions into a single string or applying a weighted sum of the scores from each expansion.

OBJECTIVES

Improving Performance

OBJECTIVES

OBJECTIVES

FIRST APPROACH

The first approach combines query expansions into a single string, evaluating scenarios with 0 (baseline), 1, 3, 5, and 7 expansions. Results show that the `map` metric improves from 0.151 to 0.166, and `ndcg` increases from 0.302 to 0.376, both peaking at 7 expansions. However, `ndcg_cut.1` and precision shows mixed results, with minor gains at the 10-document level but declines at the single-document level. While improvements in `map` and `ndcg` are measurable, they remain modest.

	name	map	ndcg	ndcg_cut.1	ndcg_cut.10	P.1	P.10	recall.1	recall.10
0	no expansions: baseline	0.151	0.302	0.423	0.326	0.437	0.238	0.057	0.152
1	1 expansions	0.158	0.339	0.399	0.324	0.415	0.233	0.059	0.159
2	3 expansions	0.152	0.357	0.399	0.318	0.412	0.231	0.053	0.159
3	5 expansions	0.165	0.375	0.426	0.334	0.433	0.242	0.063	0.159
4	7 expansions	0.166	0.376	0.447	0.340	0.455	0.246	0.059	0.165
5	9 expansions	0.160	0.371	0.432	0.336	0.446	0.245	0.058	0.165

Improving Performance

OBJECTIVES

OBJECTIVES

SECOND APPROACH

Query expansions are processed individually, with scores normalised and weighted—0.5 for the original query and 0.5/number-of-expansions for each expansion. Testing across various expansion counts shows consistent improvements over the baseline. This approach outperforms Option 1, offering more consistent improvements, though execution times are longer. The improvements, while measurable, remain moderate rather than transformative.

	name	map	ndcg	ndcg_cut.1	ndcg_cut.10	P.1	P.10	recall.1	recall.10
0	no expansions: baseline	0.151	0.302	0.423	0.326	0.437	0.238	0.057	0.152
1	1 expansions	0.156	0.342	0.387	0.319	0.402	0.229	0.057	0.159
2	3 expansions	0.170	0.398	0.446	0.343	0.458	0.247	0.058	0.168
3	5 expansions	0.173	0.421	0.433	0.347	0.443	0.252	0.059	0.171
4	7 expansions	0.174	0.438	0.449	0.347	0.458	0.250	0.058	0.165
5	9 expansions	0.174	0.448	0.460	0.352	0.471	0.256	0.059	0.173

THANKS FOR YOUR ATTENTION

Andrea and Filippo