First Stage Review

What is the problem?

- A large amount of the Woo-requests is not being fulfilled in time
  - You need to fulfil a request in 28 (max. 42) days.
  - Yet, in 2023, it took the dutch ministries on average 172 days.
- Fulfilling a Woo-request is VERY time consuming
  - Manually look for documents/emails, and manually check what pieces are relevant.

- Van Huffelen (2024): The government needs to make information more available on one central place

“Just put it in a database” - Problems and issue that arise

- Data is very messy
  - Printed → Scan → OCR
  - Unrelated content (e.g. logo’s, contact information, page footers/headers etc.)
- Lots of different types of data (e.g. documents, images, emails)
- Data is spread out everywhere (e.g. a combination of emails), randomly sent documents
- Not annotated

Research Questions

- How can state-of-the-art information retrieval methods be used to effectively identify and categorize Woo-requested information within Dutch Ministries?

- What technological improvements can Dutch Ministries implement to enhance the accuracy and efficiency of document retrieval for Woo-requests?

Experiment - Part 1/2

- Quantitative experiment using data from Woogle*.
- Everything will be compared by using BM25 vs BERTje vs Llama3

- Part 1: Retrieving 1 relevant page
  - Goal: How well can you find the corresponding dossier.
- Part 1.5: Retrieving 1-n relevant pages
  - Goal: How well can you find corresponding pages/documents.
- Part 2: Showing similarity between documents
  - Goal: How well can you categorize different documents.

*This is the most accurate database that I have access to right now. Even though WoogleDumps are about requests that have already been finished, it represents the data quite accurately.

Experiment - Part 2/2

- Ground Truth == 12 ministries (12 dossiers)
- Testing Ground truth vs 1 dossier of every ministry (12 dossiers)
- Testing Ground truth vs 5 dossier of every ministry (60 dossiers)

- Querying entire request file
- Querying Extracted request using GPT-4
- Querying 5 Extracted words using GPT-4

Experiment - Assumptions

- In my data, the *exact* original request is actually often not available. But a “Publicatieversie” is available, which includes a lot more.
- We are working with a database that has the content saved in there.
- Pages in a dossier is considered “Ground Truth”
  - All the content in a dossier is relevant.
  - Data is related to only 1 dossier.

Part 1: Retrieving 1 relevant page

Subset == 1 dossier per ministry

Prompt	BM25Okapi	BERTje	Llama3
Entire Request File	0.182	0.273	0.091
Extracted & Paraphrased Request	0.364	0.455	0.545
5 Extracted Keywords	0.455	0.545	**0.636**

Subset == 5 dossiers per ministry

Prompt	BM25Okapi	BERTje	Llama3
Entire Request File	0	0	0
Extracted & Paraphrased Request	0.091	0.273	0
5 Extracted Keywords	**0.455**	0.364	0.273

Part 1.5: Retrieving 1-n pages

Subset == Ground Truth Size

Prompt	BM25Okapi		BERTje		Llama3	
Entire Request File	P@1 = 0.182 P@20 = 0.114	R@1 = 0.009 R@20 = 0.113	P@1 = 0.273 P@20 = 0.091	R@1 = 0.014 R@20 = 0.090	P@1 = 0.091 P@20 = 0.068	R@1 = 0.005 R@20 = 0.068
	MAP@20 = 0.023		MAP@20 = 0.022		MAP@20 = 0.013	
Extracted & Paraphrased Request	P@1 = 0.364 P@20 = 0.159	R@1 = 0.018 R@20 = 0.158	P@1 = 0.455 P@20 = 0.182	R@1 = 0.023 R@20 = 0.180	P@1 = 0.545 P@20 = 0.132	R@1 = 0.027 R@20 = 0.131
	MAP@20 = 0.041		MAP@20 = 0.071		MAP@20 = 0.057	
5 Extracted Keywords	P@1 = 0.455 **P@20 = 0.223**	R@1 = 0.023 **R@20 = 0.221**	P@1 = 0.545 P@20 = 0.191	R@1 = 0.026 R@20 = 0.183	**P@1 = 0.636** P@20 = 0.177	**R@1 = 0.032** R@20 = 0.176
	MAP@20 = 0.070		**MAP@20 = 0.073**		MAP@20 = 0.062	

Part 1.5: Retrieving 1-n pages

Subset == ~5 times as big (i.e. 5 dossiers per ministry)

Prompt	BM25Okapi		BERTje		Llama3	
Entire Request File	P@1 = 0.000 P@20 = 0.009	R@1 = 0.000 R@20 = 0.009	P@1 = 0.000 P@20 = 0.014	R@1 = 0.000 R@20 = 0.014	P@1 = 0.000 P@20 = 0.000	R@1 = 0.000 R@20 = 0.000
	MAP@20 = 0.001		MAP@20 = 0.003		MAP@20 = 0.000	
Extracted & Paraphrased Request	P@1 = 0.091 P@20 = 0.095	R@1 = 0.005 R@20 = 0.095	P@1 = 0.273 P@20 = 0.059	R@1 = 0.014 R@20 = 0.059	P@1 = 0.000 P@20 = 0.059	R@1 = 0.000 R@20 = 0.059
	MAP@20 = 0.022		MAP@20 = 0.031		MAP@20 = 0.017	
5 Extracted Keywords	**P@1 = 0.455** **P@20 = 0.205**	**R@1 = 0.023** **R@20 = 0.203**	P@1 = 0.364 P@20 = 0.114	R@1 = 0.018 R@20 = 0.113	P@1 = 0.273 P@20 = 0.068	R@1 = 0.014 R@20 = 0.068
	MAP@20 = 0.063		MAP@20 = 0.050		MAP@20 = 0.029	

Part 2: Showing similarity between documents

Subset == Ground Truth Size

Prompt	BM25Okapi		BERTje		Llama3	
Every page in every dossier	P@1 = 0.789 P@20 = 0.408	R@1 = 0.042 R@20 = 0.393	P@1 = 0.826 P@20 = 0.412	R@1 = 0.045 R@20 = 0.415	**P@1 = 0.860** **P@20 = 0.441**	**R@1 = 0.047** **R@20 = 0.429**
	MAP@20 = 0.106		MAP@20 = 0.114		**MAP@20 = 0.119**	

Part 2: Showing similarity between documents

Subset == ~5 times as big (i.e. 5 dossiers per ministry)

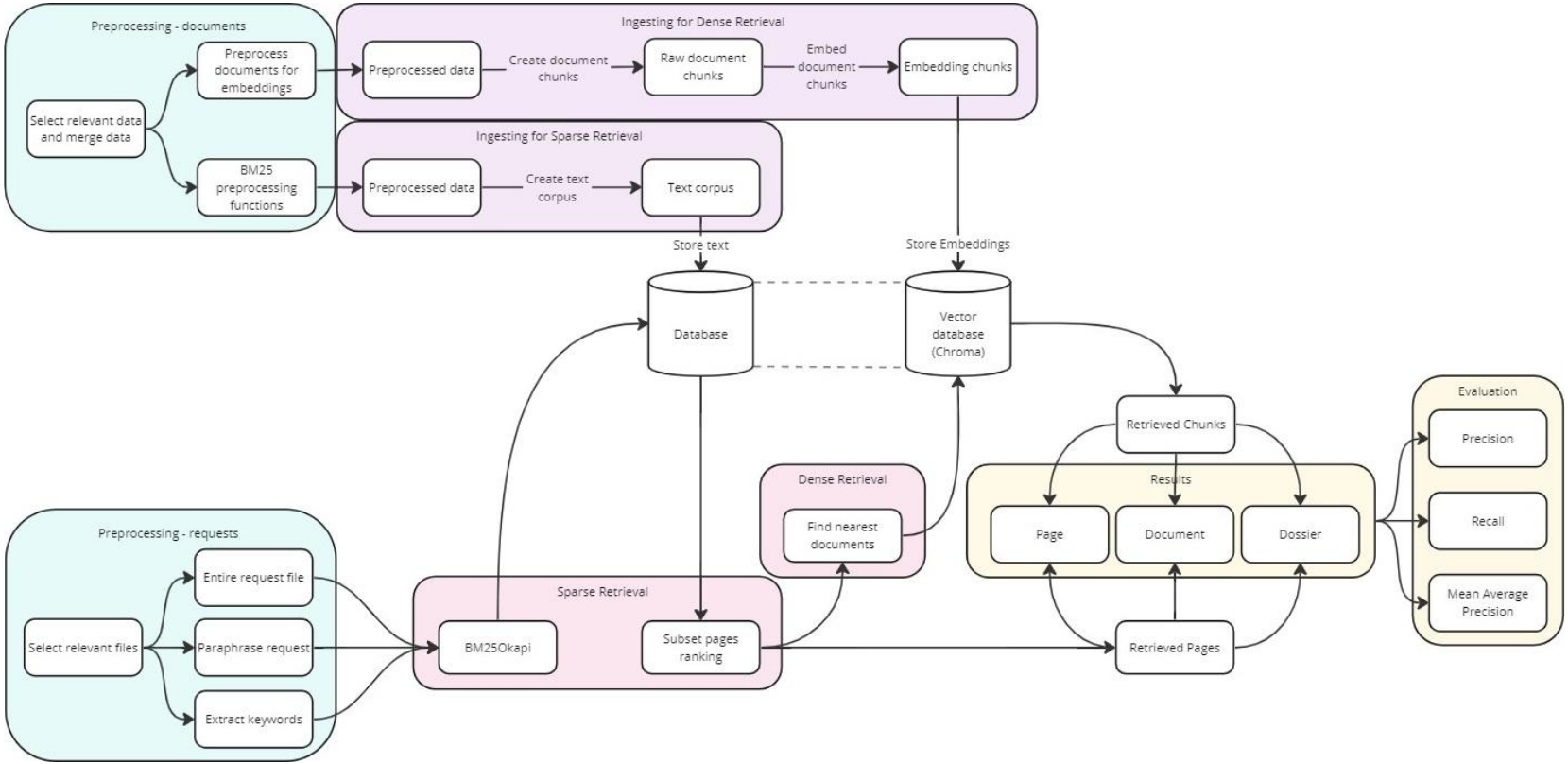
Prompt	BM25Okapi		BERTje		Llama3	
Every page in every dossier	P@1 = 0.220 P@20 = 0.172	R@1 = 0.014 R@20 = 0.205	P@1 = 0.764 P@20 = 0.339	R@1 = 0.041 R@20 = 0.332	**P@1 = 0.800** **P@20 = 0.364**	**R@1 = 0.044** **R@20 = 0.352**
	MAP@20 = 0.044		MAP@20 = 0.096		**MAP@20 = 0.104**	

Document Embeddings on 2D Vector Space

BERTje

Llama3

Experiment - Graph



Temporary conclusion

- Extracted keywords + BM25 works best as a “start location”, but not for finding all related pages.

- When having a page that is considered relevant, an approach with document similarity with dense retrieval works best for finding more pages.

Additions/directions to make it (potentially) better

- Automatic label/keyword extraction for pages.
- Metadata about: date, topic, ministry.
- Hybrid search (combining best of both worlds)

- Testing with more ground truth data.
- Testing with bigger dataset.
- Testing with other embeddings algorithms.

Future - SSC-ICT

- Usable for optimizing the retrieval in RAG.
- Usable for optimizing the retrieval in DMS.