# Retrieval Evaluation for RAG Systems

Craig Franze, Baian Liu, Mohammad Nooranidoost,
Himanshu Raj, Anil Tokmak & Peter Williams

THE ERDŐS INSTITUTE
Helping PhDs get and create jobs they love at every stage of their career.

Aware

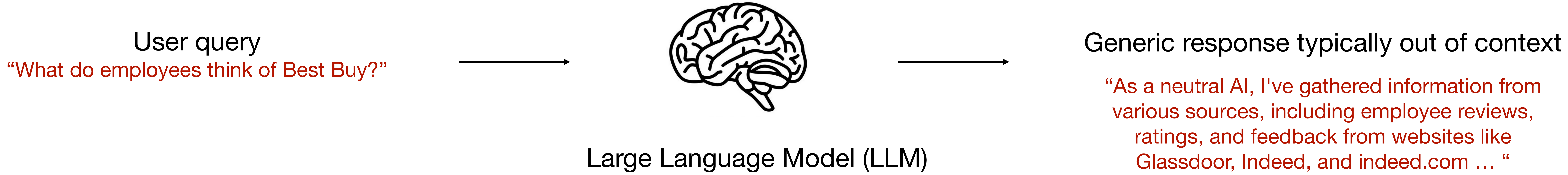
User query

“What do employees think of Best Buy?”

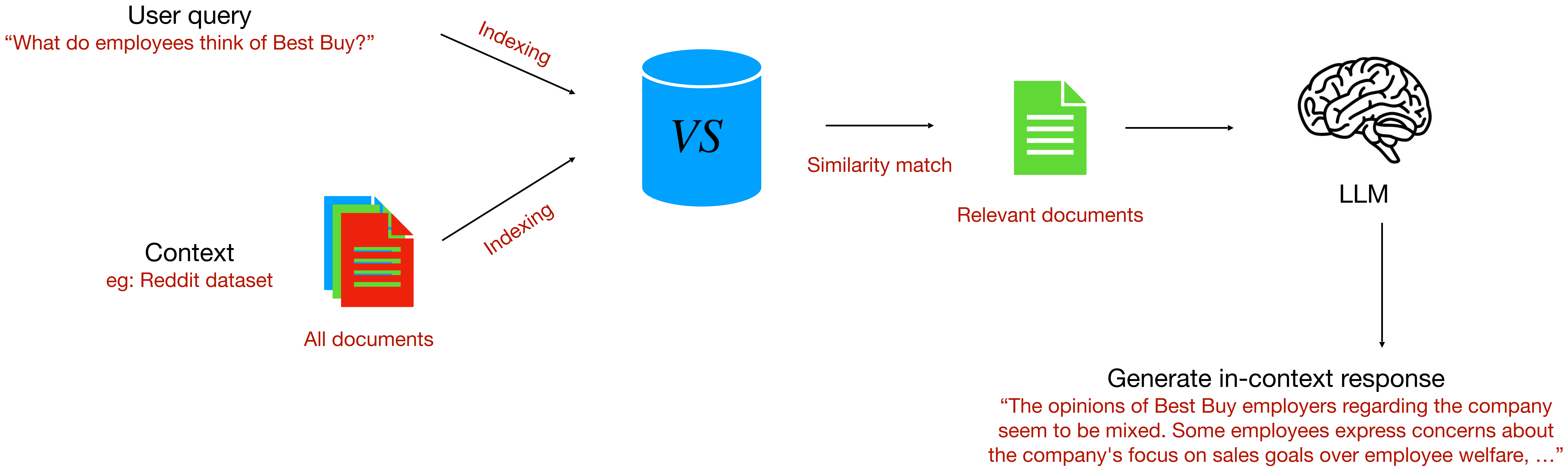
Large Language Model (LLM)

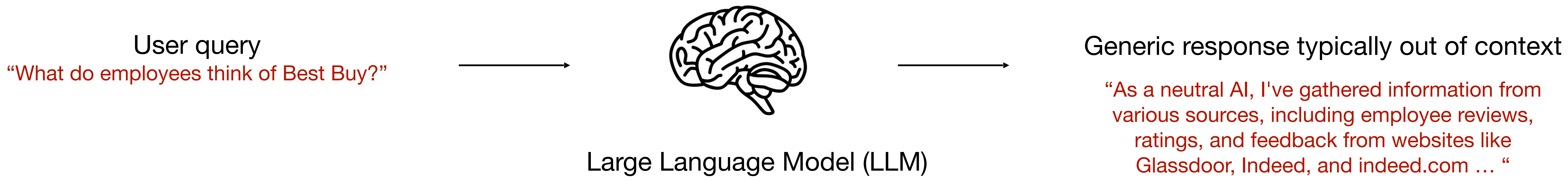
Generic response typically out of context

“As a neutral AI, I've gathered information from various sources, including employee reviews, ratings, and feedback from websites like Glassdoor, Indeed, and indeed.com … “




Explicitly provide context to a large language model:



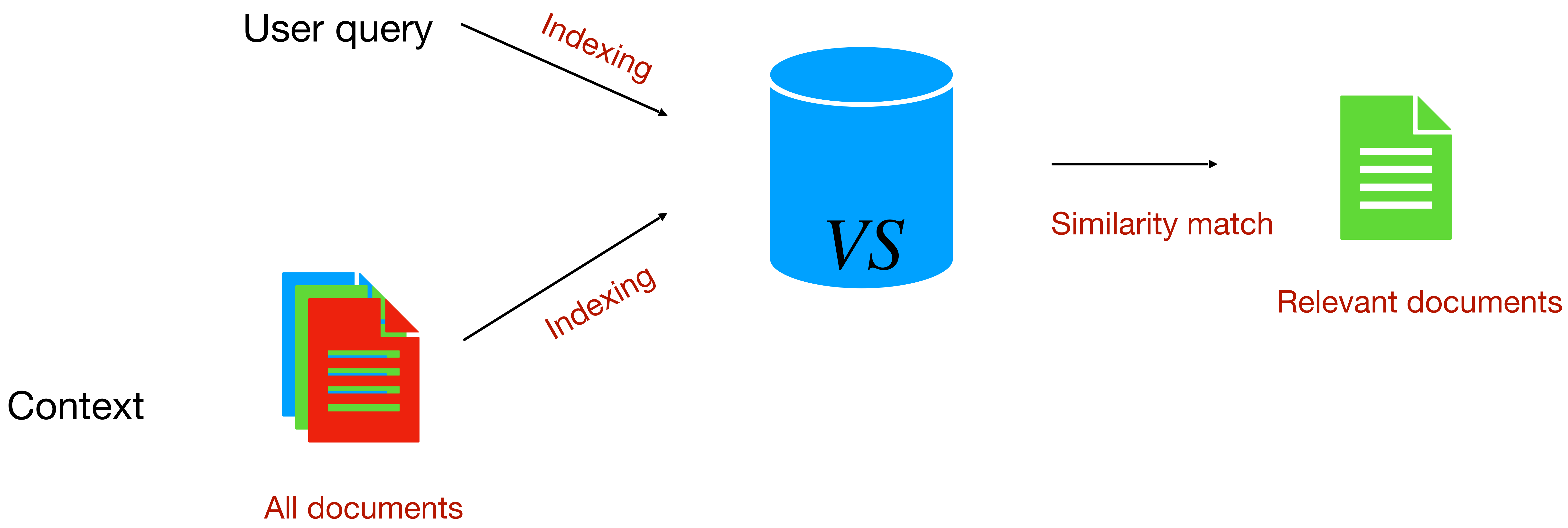



Explicitly provide context to a large language model:


**Retrieval Augmented Generation (RAG)**


User query
“What do employees think of Best Buy?”
Large Language Model (LLM)
Generic response typically out of context
“As a neutral AI, I've gathered information from various sources, including employee reviews, ratings, and feedback from websites like Glassdoor, Indeed, and indeed.com … “
Explicitly provide context to a large language model:
User query
“What do employees think of Best Buy?”
Indexing
Context
eg: Reddit dataset
All documents
Indexing
VS
Similarity match
Relevant documents
LLM
Information Retrieval System
Generate in-context response
“The opinions of Best Buy employers regarding the company seem to be mixed. Some employees express concerns about the company's focus on sales goals over employee welfare, …”

# Objective




Build an information retrieval system that can, given a user's query,

- Identify the most relevant content in the provided Reddit dataset
- Rank the retrieved content

Assessment of the retrieval system

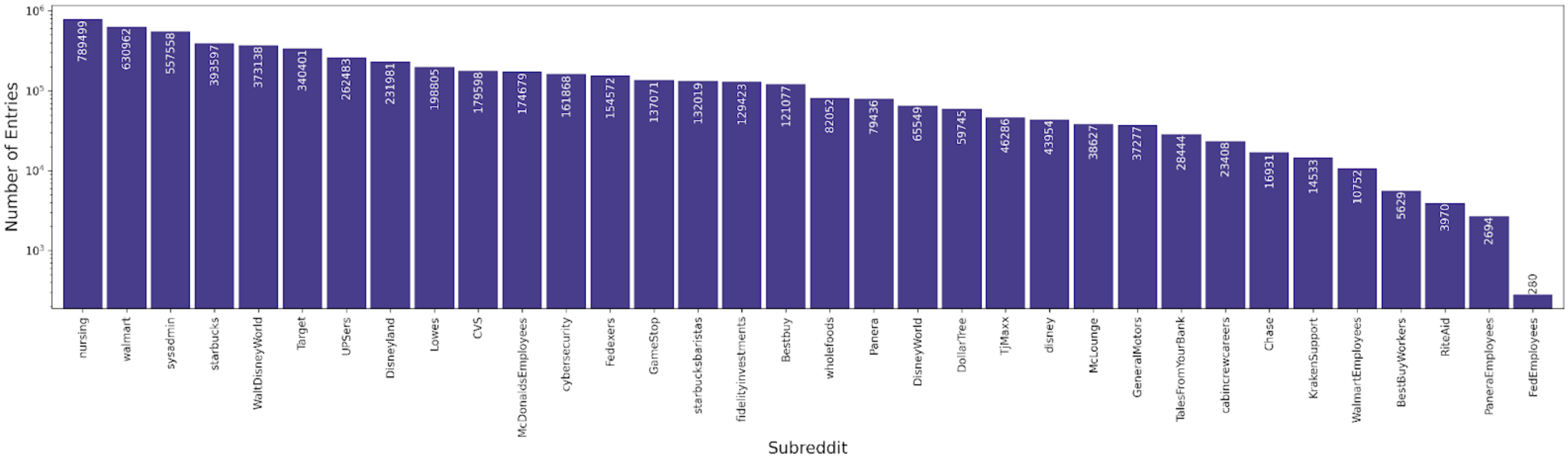
- A quantitative measure of the performance of the retrieval methods

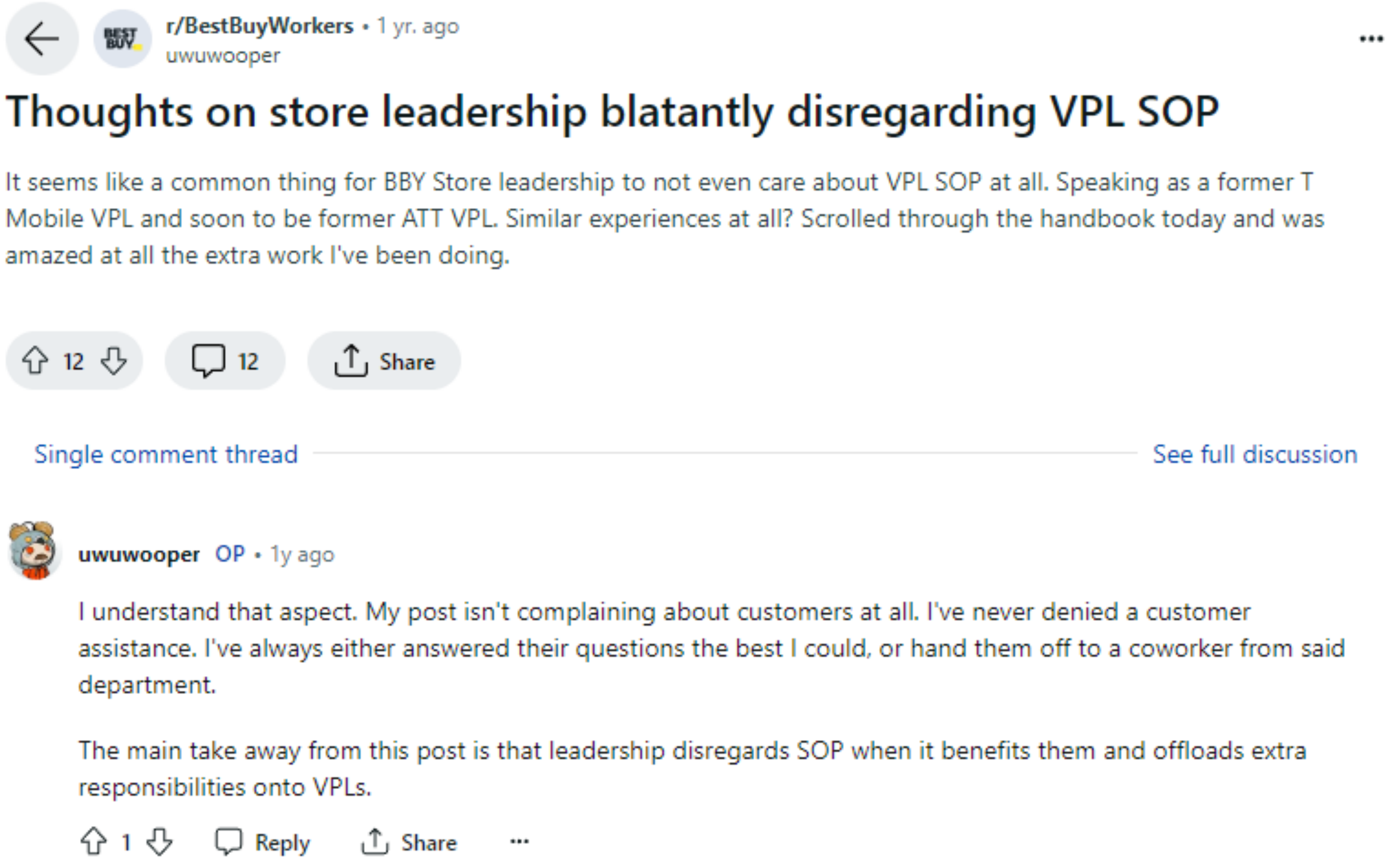
# Dataset

Context

All documents

Reddit dataset comprising of > 5.5M posts covering 34 subreddits of submissions and comments



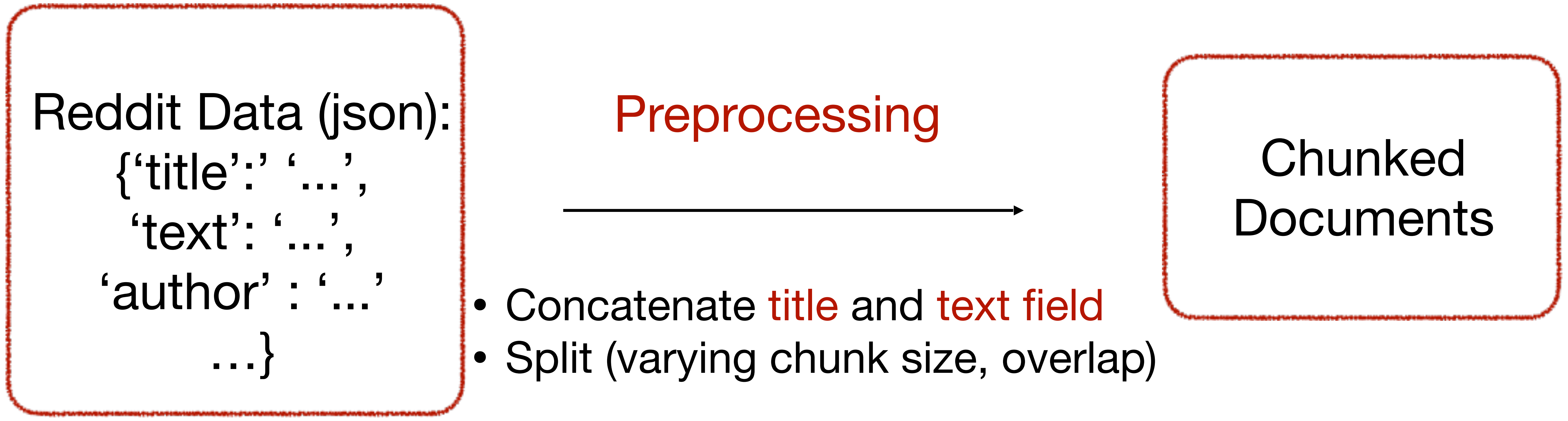



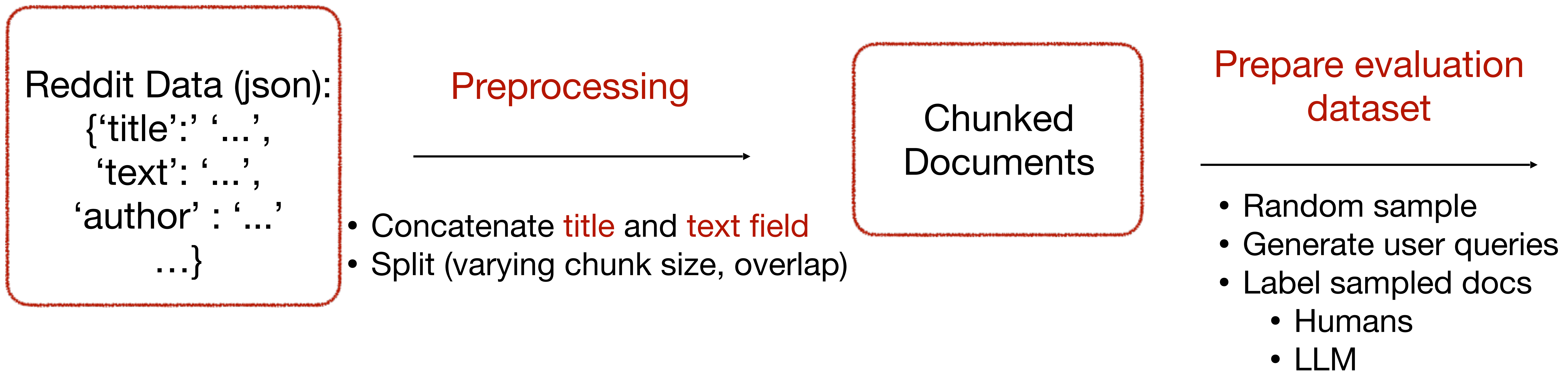

Relevant context: **BestBuyWorkers** subreddit (5629 entries comprised of submissions and comments + metadata)

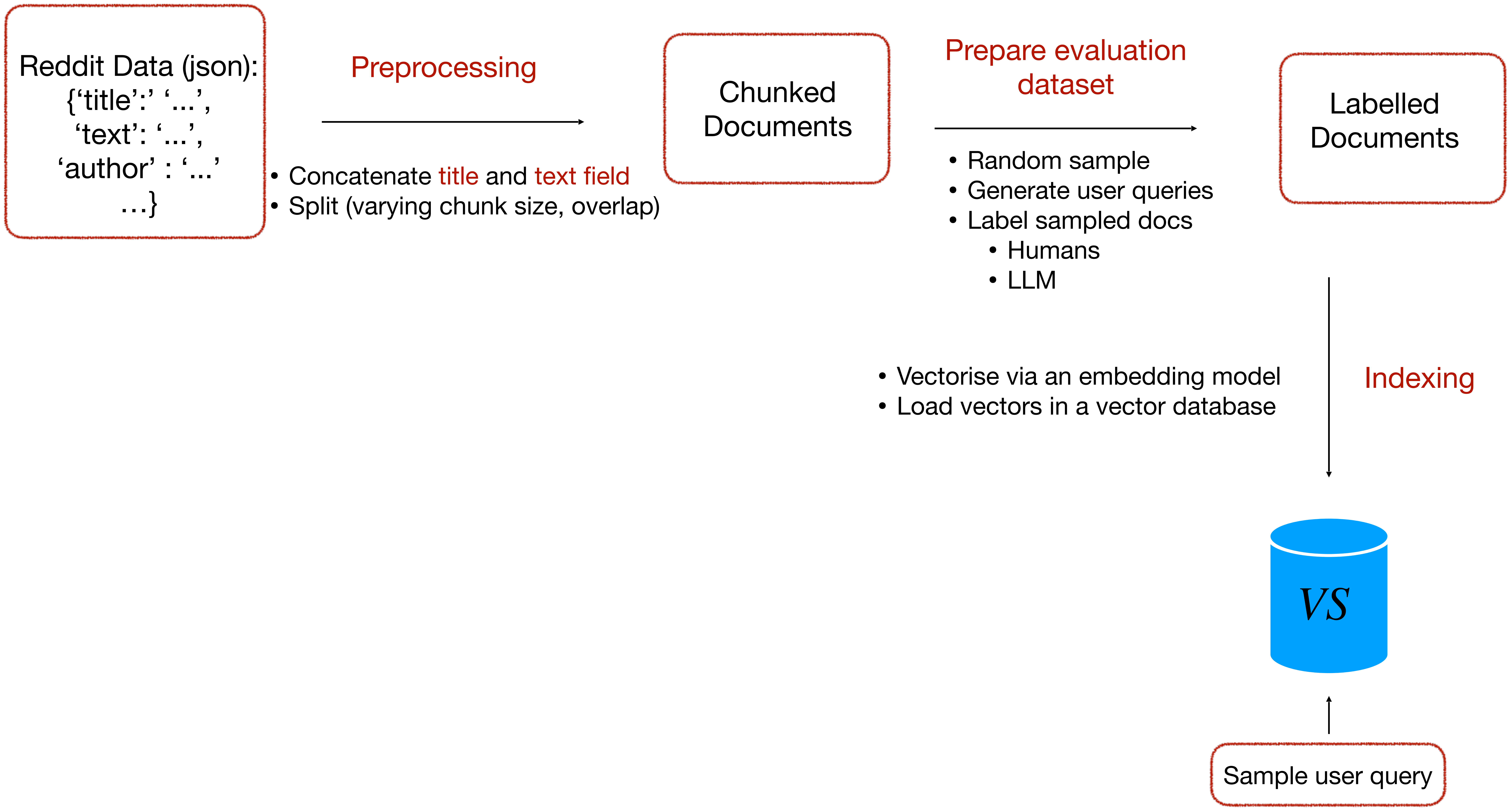

```
{'aware_post_type': 'comment',
 'aware_created_ts': '2023-01-23T11:20:15',
 'reddit_id': 'j5k65rh',
 'reddit_name': 't1_j5k65rh',
 'reddit_created_utc': 1674490815,
 'reddit_author': 'uwuwooper',
 'reddit_text': "I understand that aspect. My post isn't complaining about customers at all. I've never denied
a customer assistance. I've always either answered their questions the best I could, or hand them off to a
coworker from said department.\n\nThe main take away from this post is that leadership disregards SOP when it
benefits them and offloads extra responsibilities onto VPLs.",
 'reddit_permalink': '/r/BestBuyWorkers/comments/10i9ao0/thoughts_on_store_leadership_blatantly/j5k65rh/',
 'reddit_title': None,
 'reddit_url': None,
 'reddit_subreddit': 'BestBuyWorkers',
 'reddit_link_id': 't3_10i9ao0',
 'reddit_parent_id': 't1_j5jweo5',
 'reddit_submission': '10i9ao0'}
```

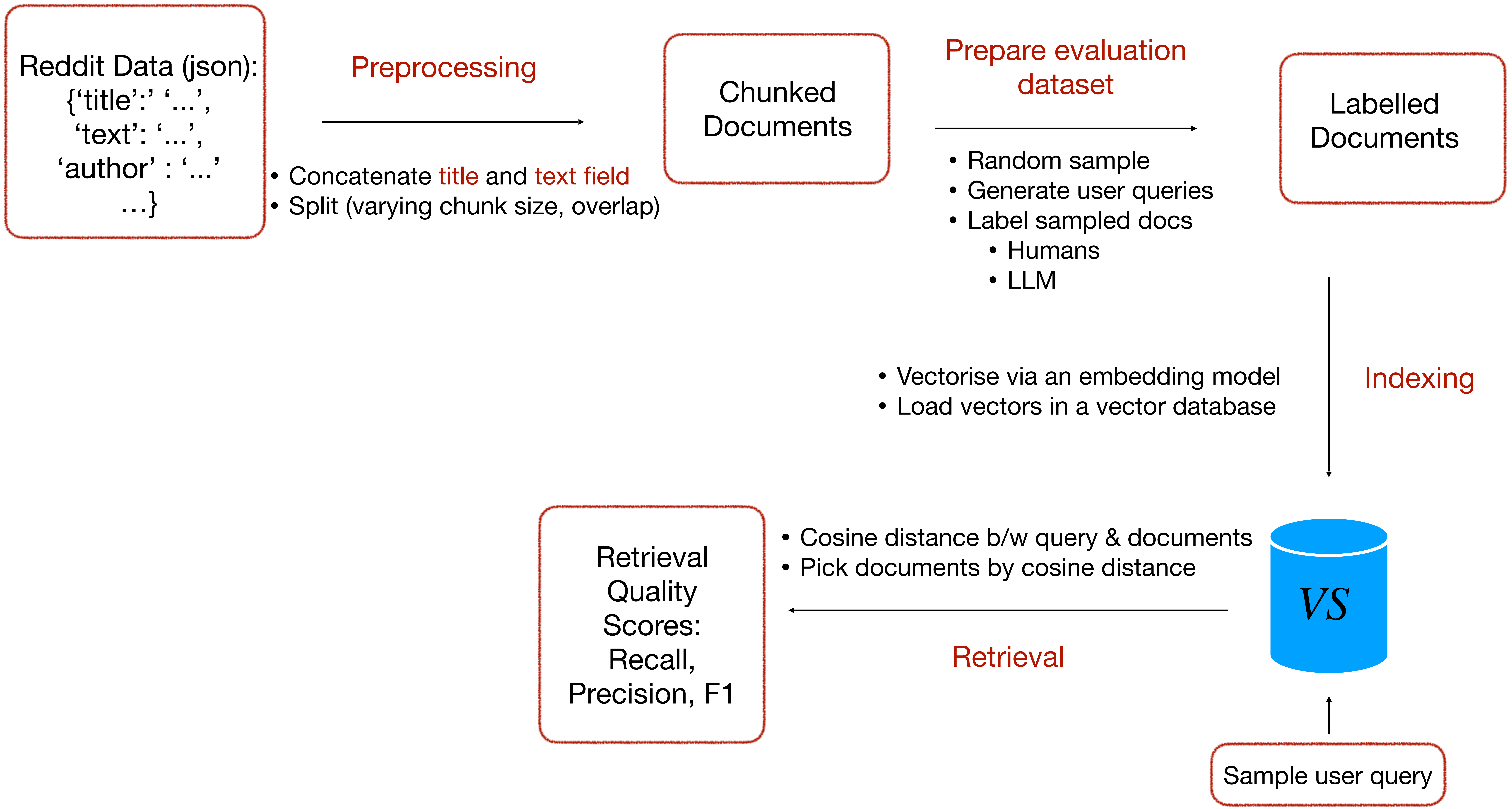
# Retrieval Pipeline and the Evaluation Dataset

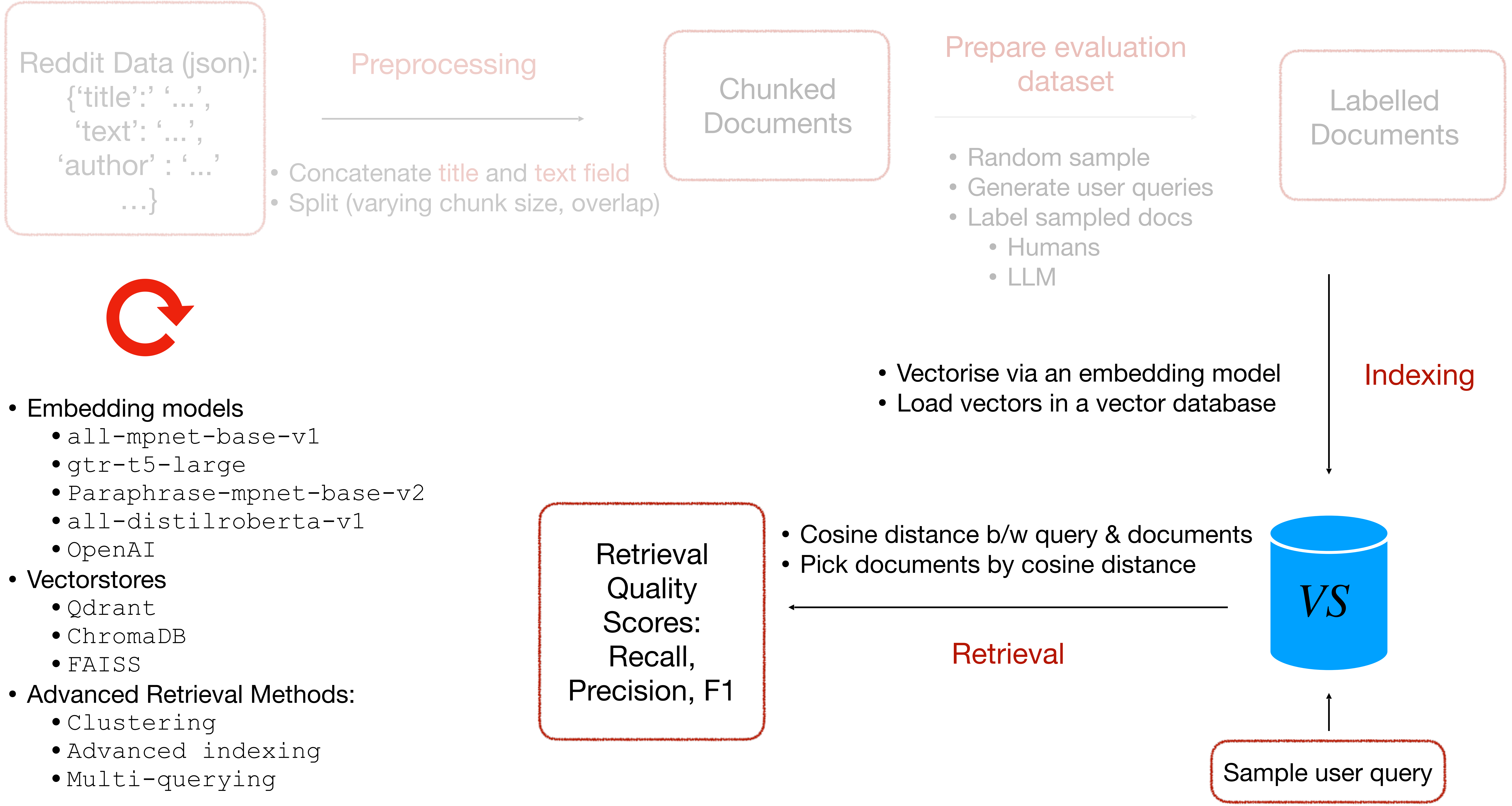
Reddit Data (json):
{‘title’:’ ‘...’,
‘text’: ‘...’,
‘author’ : ‘...’
…}


Reddit Data (json):
{‘title’:’ ‘…’,
‘text’: ‘…’,
‘author’ : ‘…’
…}
Preprocessing
• Concatenate title and text field
• Split (varying chunk size, overlap)
Chunked Documents


Reddit Data (json):
{‘title’:’ ‘…’,
‘text’: ‘…’,
‘author’ : ‘…’
…}
Preprocessing
• Concatenate title and text field
• Split (varying chunk size, overlap)
Chunked Documents
Prepare evaluation dataset
• Random sample
• Generate user queries
• Label sampled docs
• Humans
• LLM
Labelled Documents

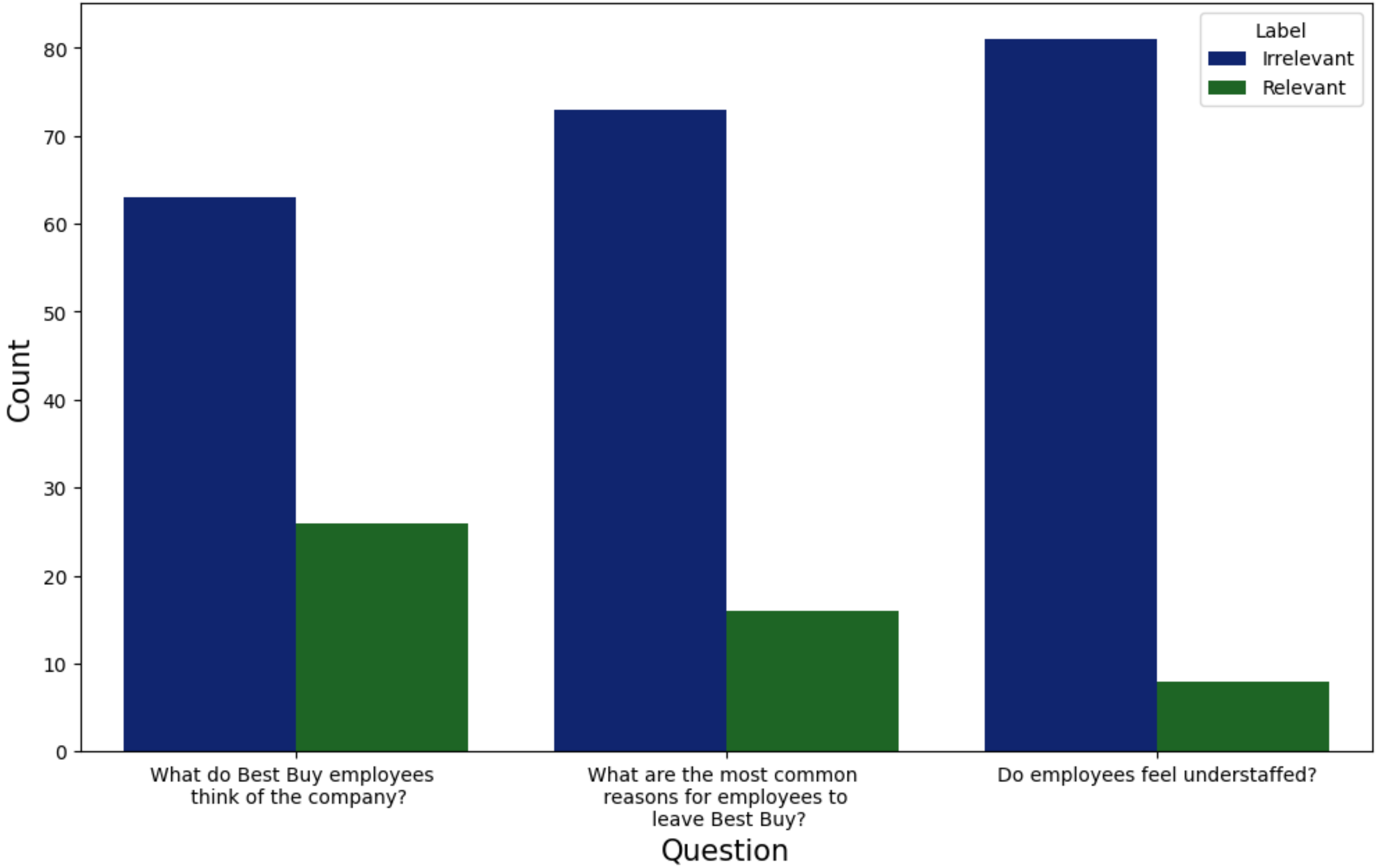

Reddit Data (json):
{'title':' '...',
'text': '...',
'author' : '...'
...}
Preprocessing
• Concatenate title and text field
• Split (varying chunk size, overlap)
Chunked Documents
Prepare evaluation dataset
• Random sample
• Generate user queries
• Label sampled docs
• Humans
• LLM
Labelled Documents
• Vectorise via an embedding model
• Load vectors in a vector database
Indexing
VS
Sample user query


Reddit Data (json): {‘title’:’ ‘…’, ‘text’: ‘…’, ‘author’ : ‘…’ …}
Preprocessing
• Concatenate title and text field
• Split (varying chunk size, overlap)
Chunked Documents
Prepare evaluation dataset
• Random sample
• Generate user queries
• Label sampled docs
• Humans
• LLM
Labelled Documents
Indexing
• Vectorise via an embedding model
• Load vectors in a vector database
VS
Sample user query
• Cosine distance b/w query & documents
• Pick documents by cosine distance
Retrieval
Retrieval Quality Scores: Recall, Precision, F1



Reddit Data (json): {'title':' '...', 'text': '...', 'author' : '...' ...}
Preprocessing
• Concatenate title and text field
• Split (varying chunk size, overlap)
Chunked Documents
Prepare evaluation dataset
• Random sample
• Generate user queries
• Label sampled docs
• Humans
• LLM
Labelled Documents
• Vectorise via an embedding model
• Load vectors in a vector database
Indexing
• Embedding models
• all-mpnet-base-v1
• gtr-t5-large
• Paraphrase-mpnet-base-v2
• all-distilroberta-v1
• OpenAI
• Vectorstores
• Qdrant
• ChromaDB
• FAISS
• Advanced Retrieval Methods:
• Clustering
• Advanced indexing
• Multi-querying
Retrieval Quality Scores: Recall, Precision, F1
• Cosine distance b/w query & documents
• Pick documents by cosine distance
VS
Retrieval
Sample user query

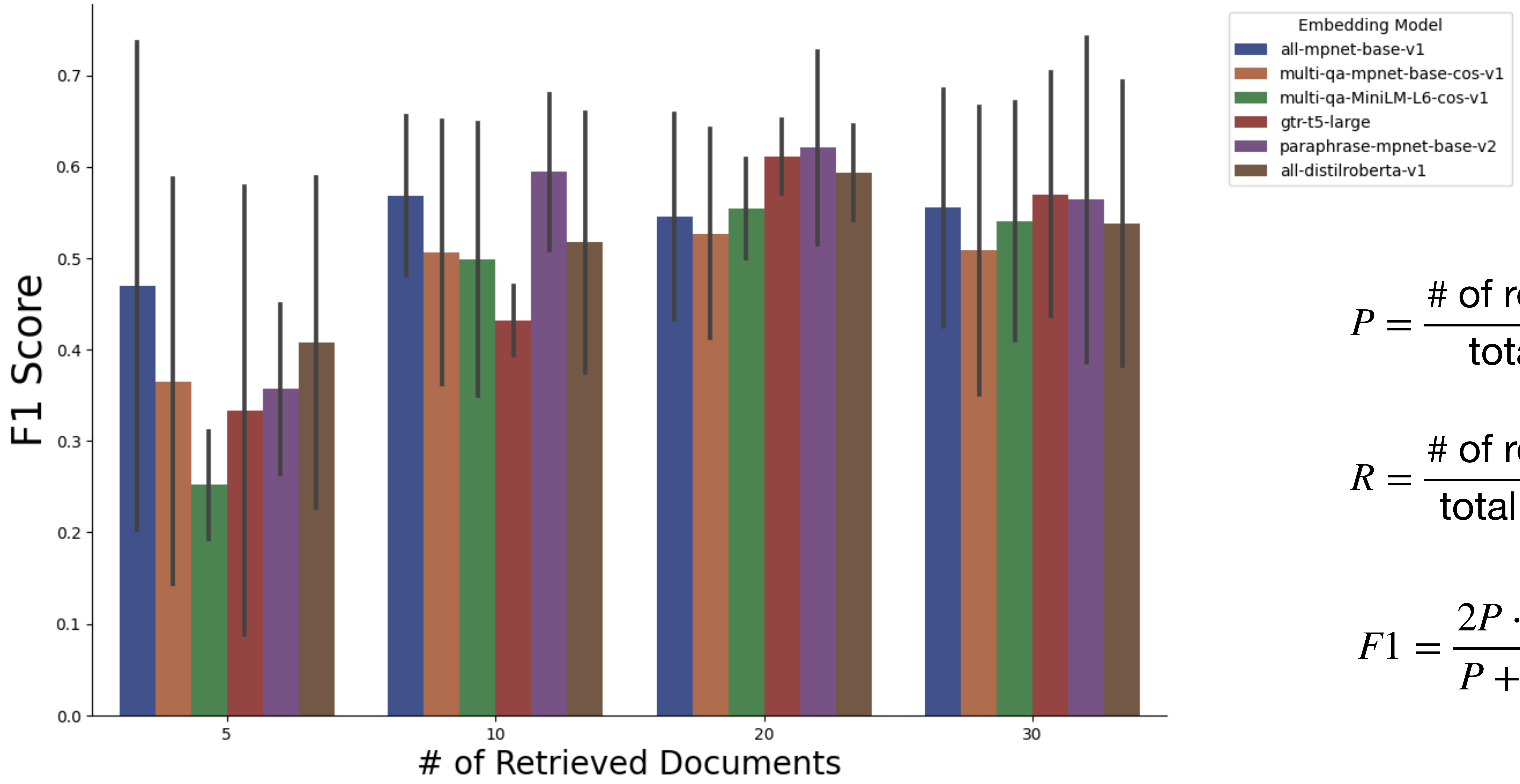
# Evaluation Dataset




- 90 Randomly Sampled Comments/Submissions
- Human Labeled Relevance for 3 Sample Questions

# Quality Evaluation for Different Embeddings

Baseline retriever



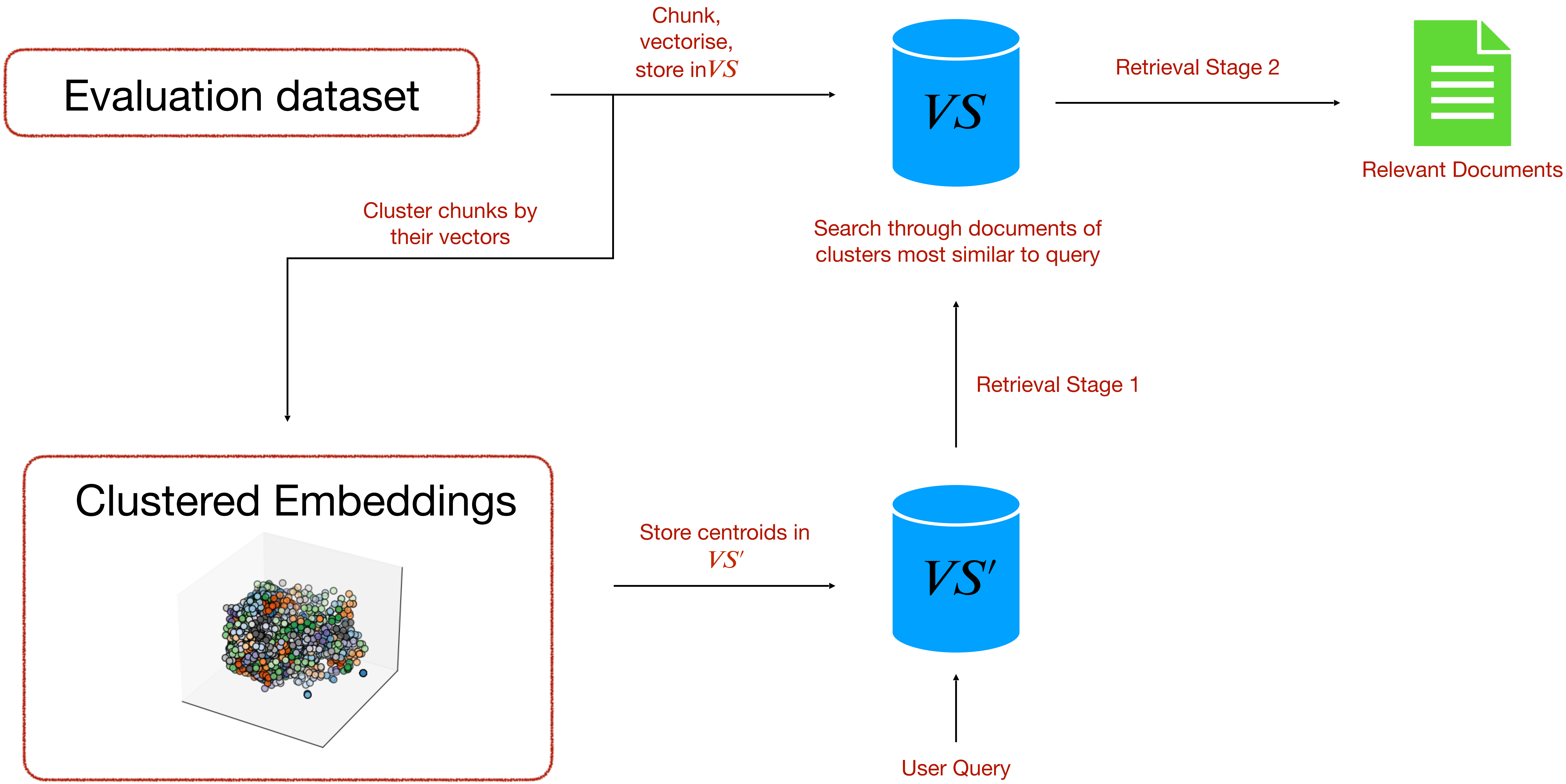

$$P = \frac{\text{\# of relevant retrievals}}{\text{total no. retrievals}}$$

$$R = \frac{\text{\# of relevant retrievals}}{\text{total no. relevant doc}}$$

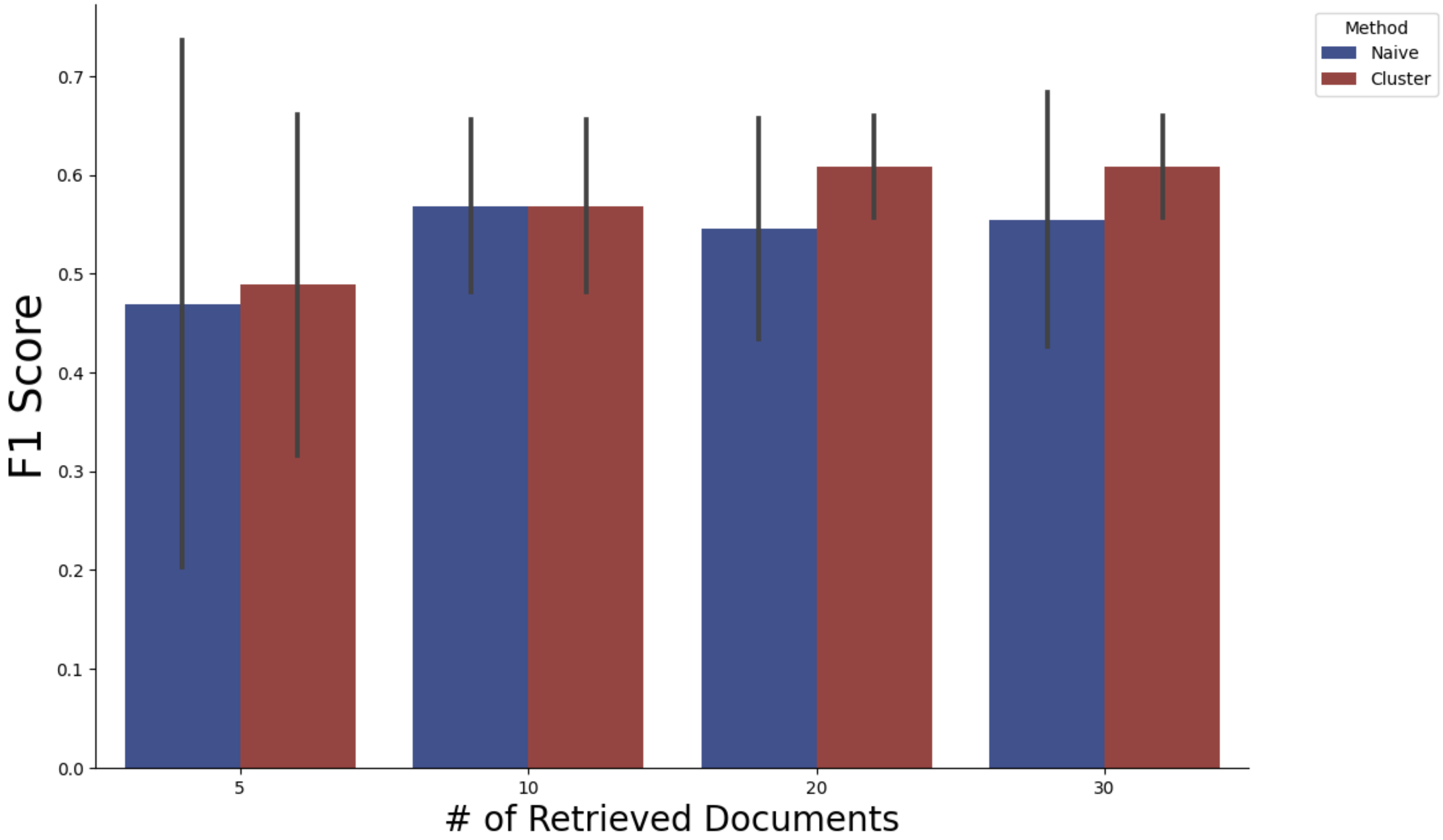
$$F1 = \frac{2P \cdot R}{P + R}$$

# Some Advanced Pipelines and Explorations

# Clustering


Evaluation dataset
Chunk, vectorise, store in $VS$
$VS$
Retrieval Stage 2
Relevant Documents
Cluster chunks by their vectors
Search through documents of clusters most similar to query
Retrieval Stage 1
Clustered Embeddings
Store centroids in $VS'$
$VS'$
User Query

# Evaluation: Clustering



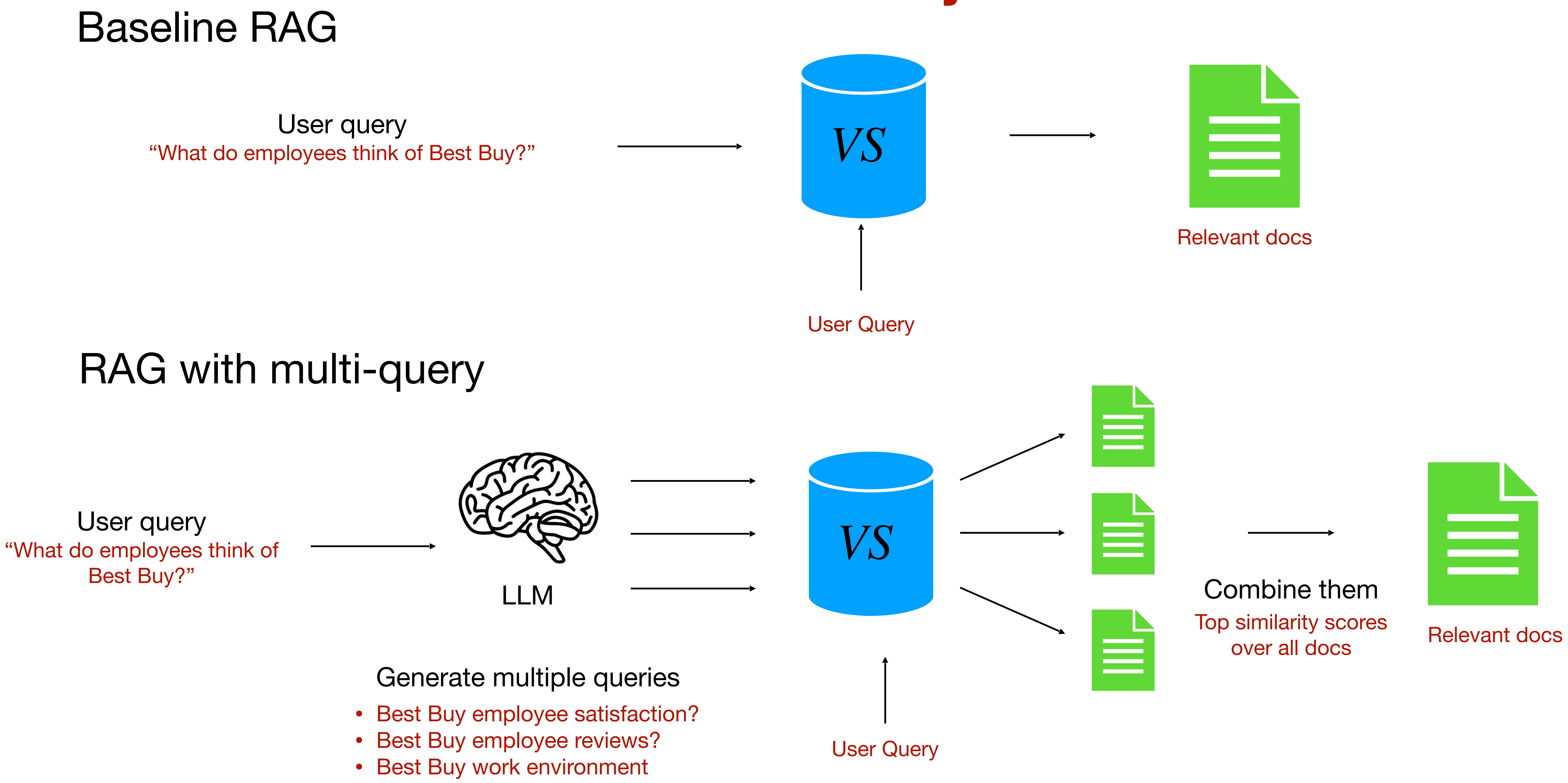

Pros:

- Slight improvements in Retrieval
- More efficient search

Cons:

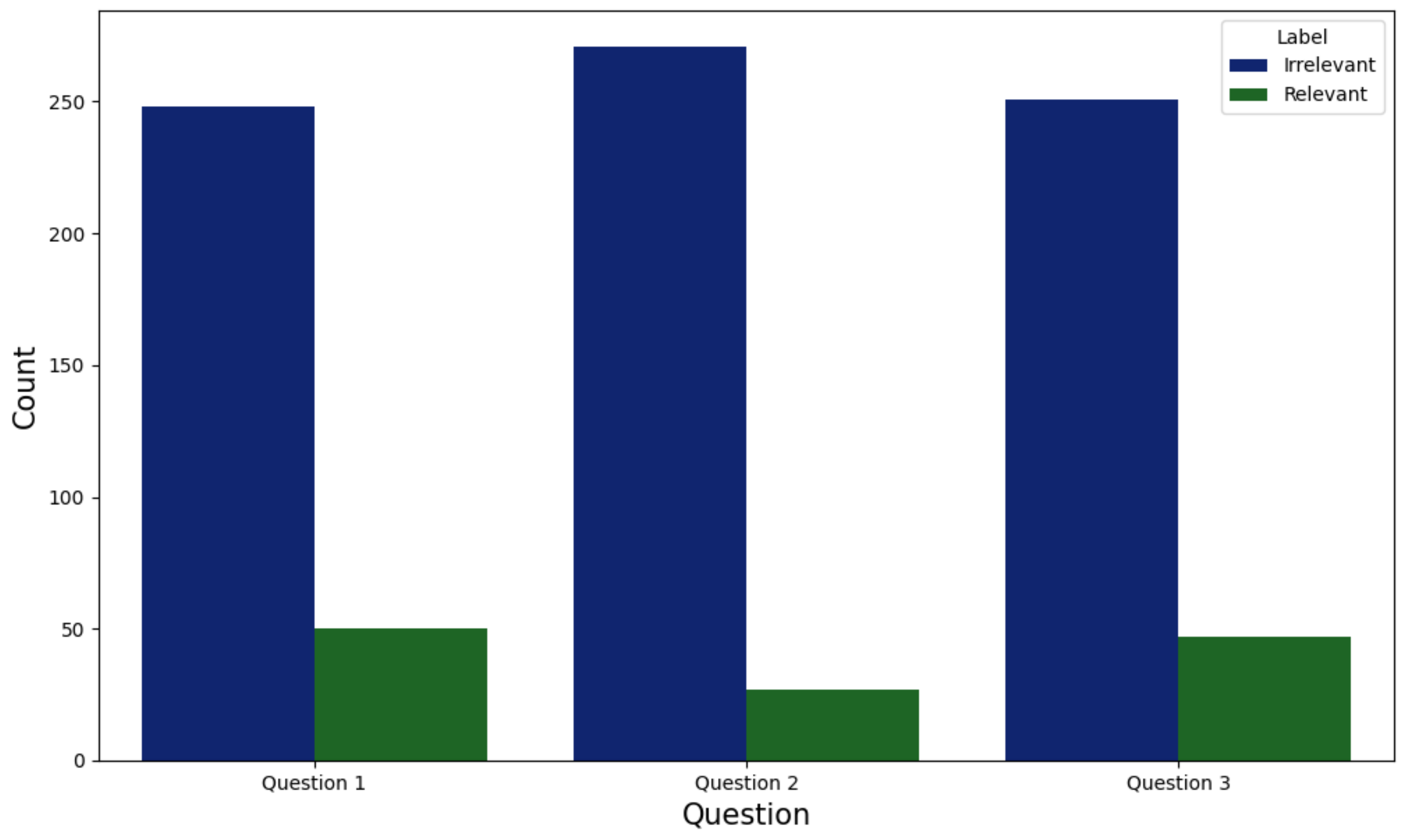
- Still miss relevant documents embedded farther from the query
- Requires regular updates to the clusters

# Multi-Query

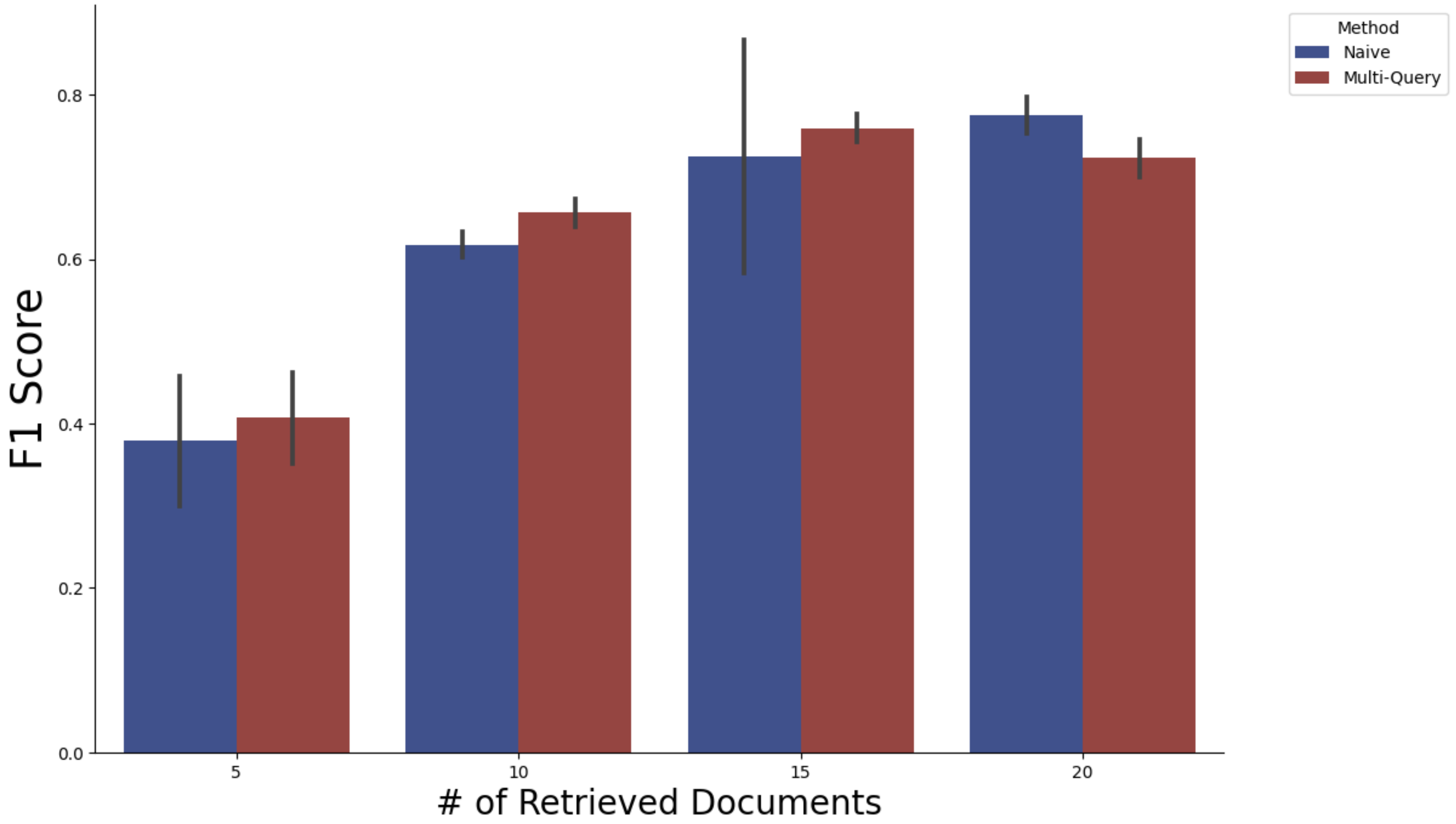


Baseline RAG
User query
"What do employees think of Best Buy?"
VS
Relevant docs
User Query
RAG with multi-query
User query
"What do employees think of Best Buy?"
LLM
Generate multiple queries
• Best Buy employee satisfaction?
• Best Buy employee reviews?
• Best Buy work environment
VS
User Query
Combine them
Top similarity scores over all docs
Relevant docs

# Evaluation: Multi-Query

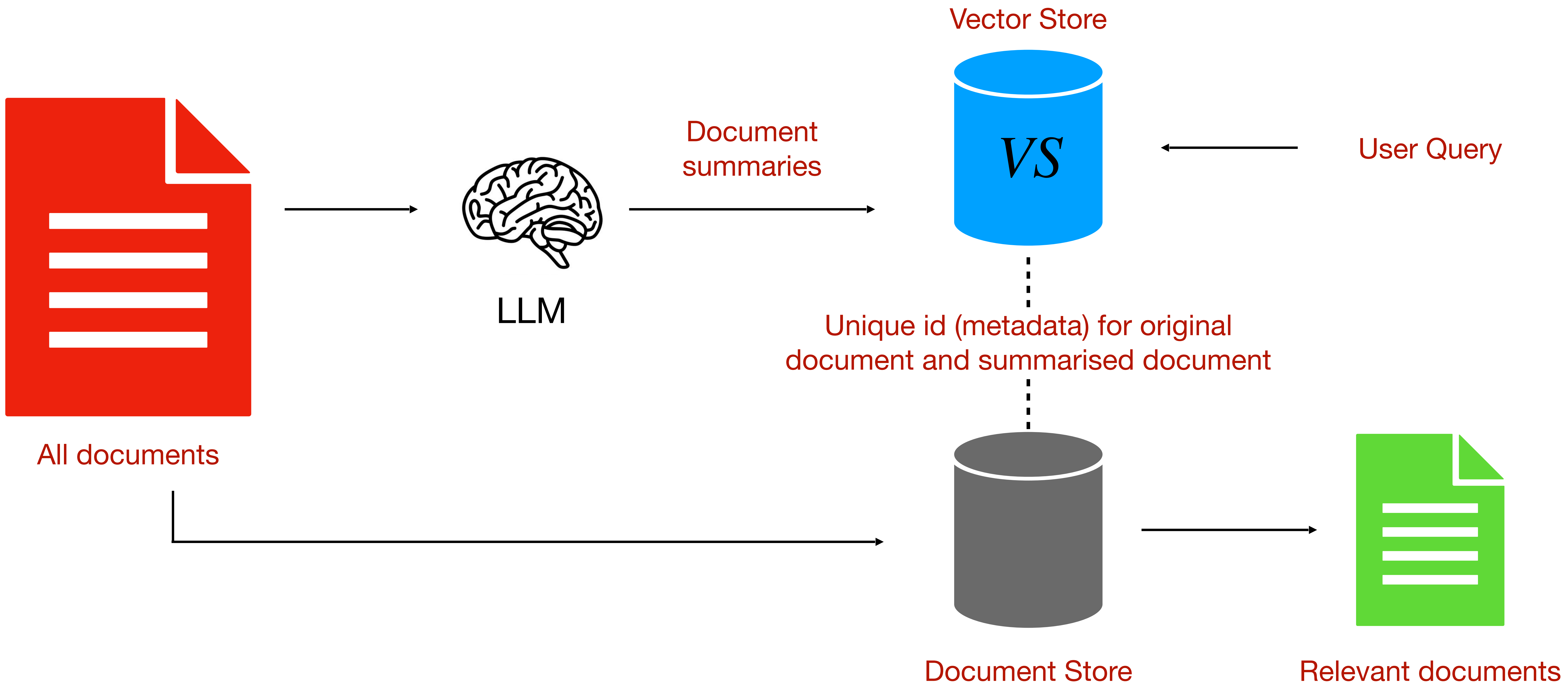



Evaluation dataset
(created using Llama3)

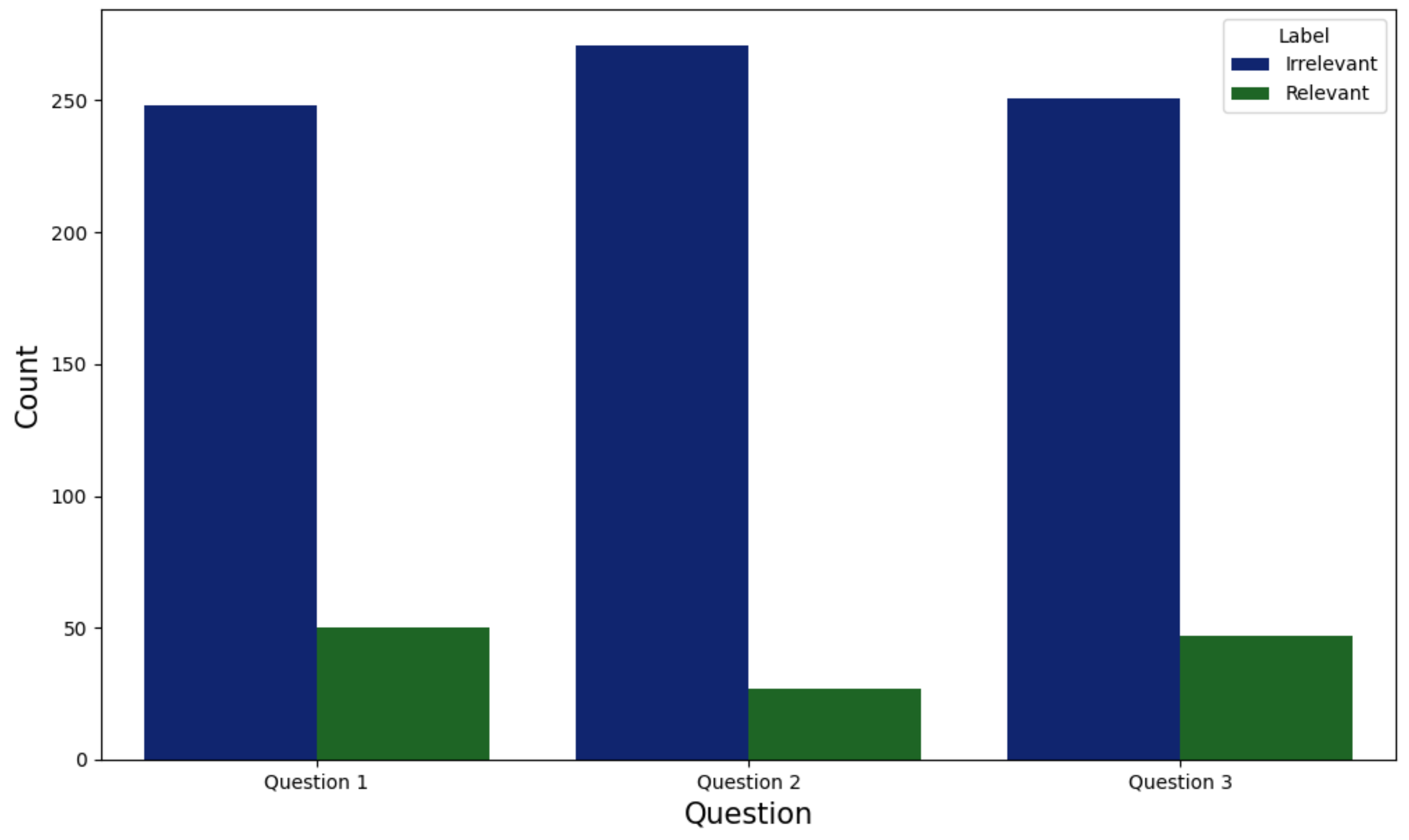



Multi-Query retrieval metric (F1 score)

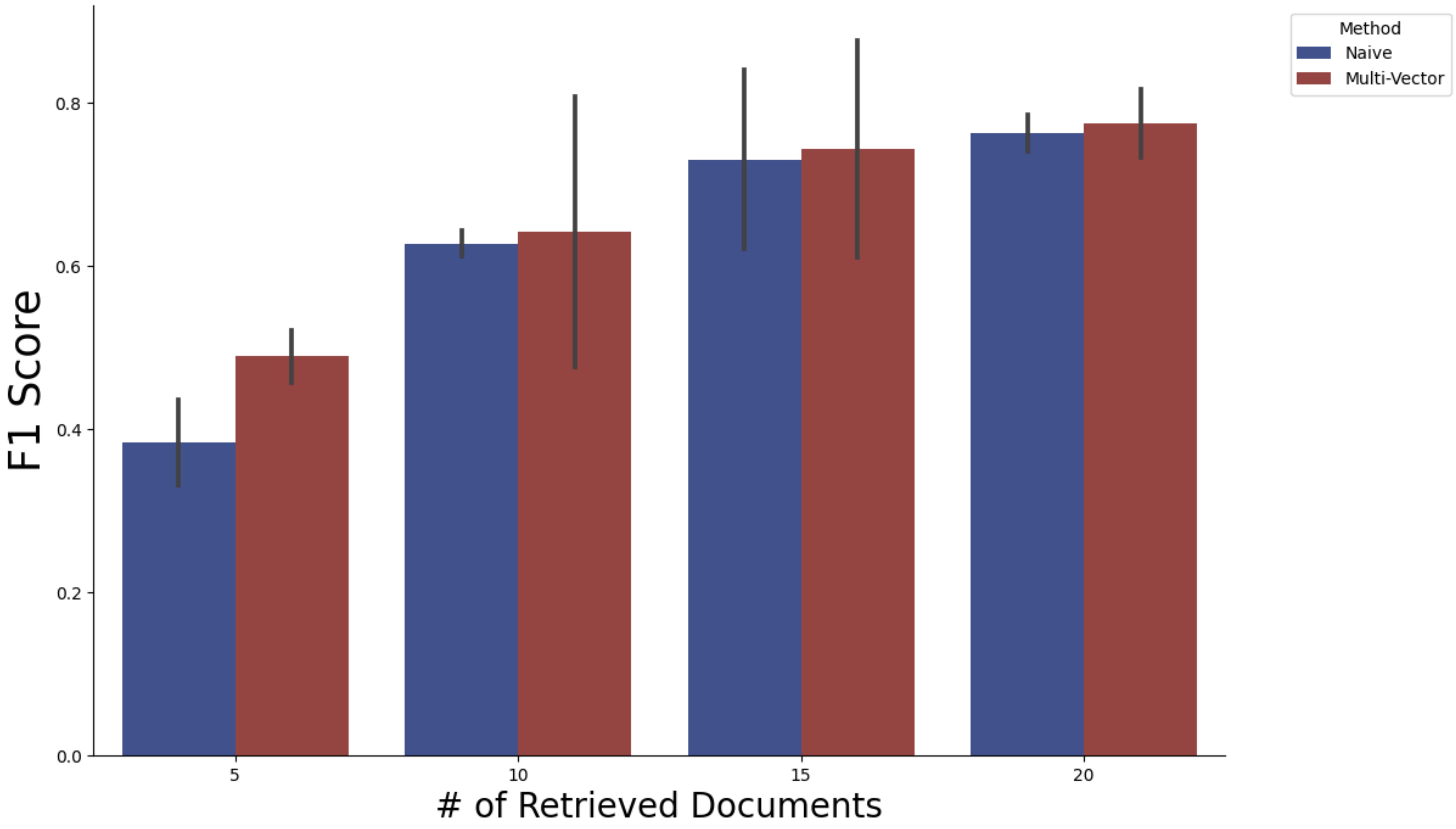
# Multi-Vector Indexing



# Evaluation: Multi-Vector Indexing




Evaluation dataset
(created using Llama3)




Multi-Vector retrieval metric (F1 score)

# Conclusions

1. Created an end-to-end Retrieval Augmented Generation pipeline for Reddit data

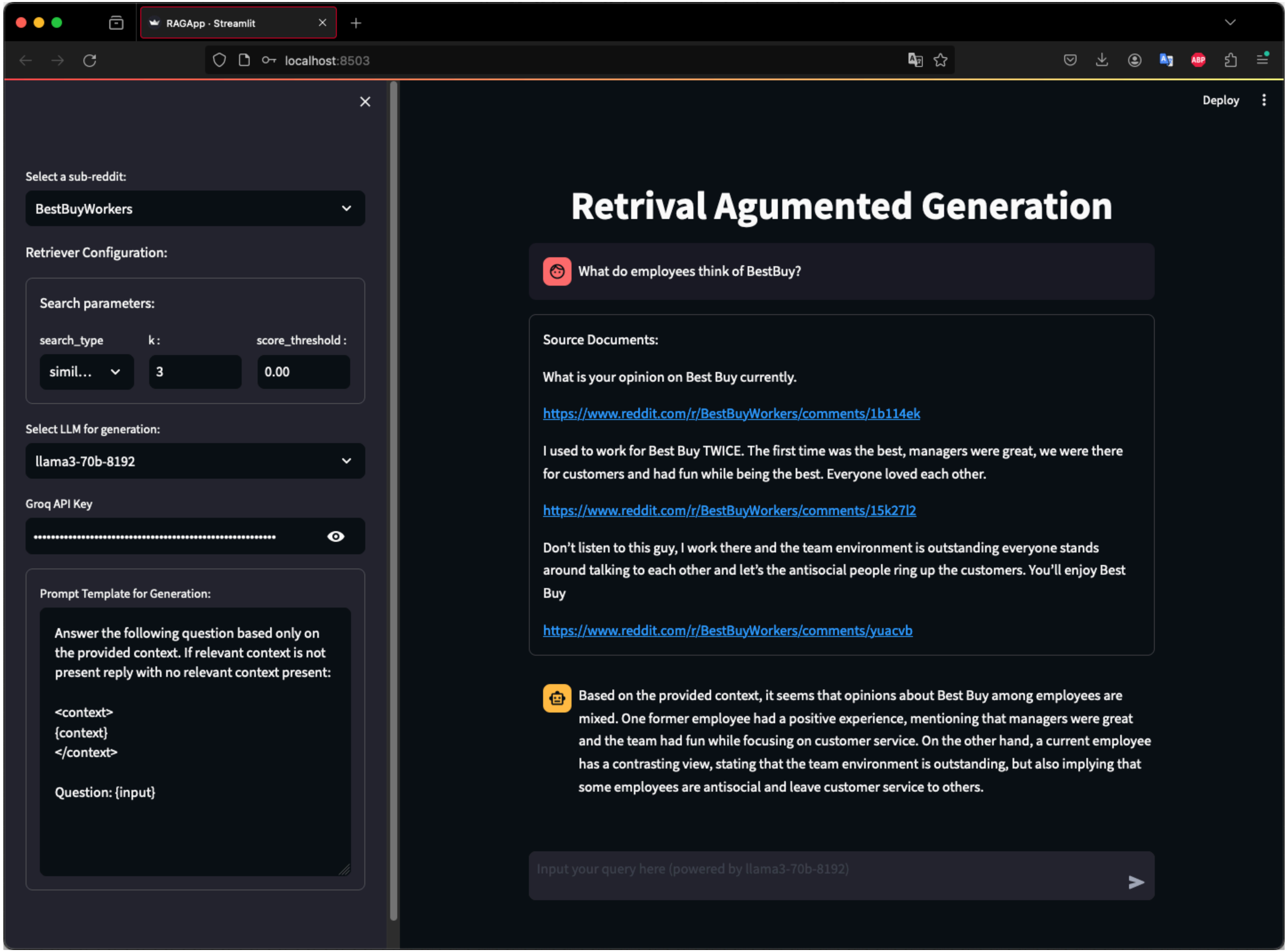
2. Devised methodologies to evaluate different retrieval pipelines

    - Prepared evaluation dataset from raw data (Human and LLM labels)

    - Evaluation metrics: precision, recall, F1

3. The different pipelines included: Naive RAG (baseline retriever), Cluster embeddings, Multi-query and Multi-indexing

    - Evaluation metrics indicate advanced retrieval methods perform better than the baseline




# Streamlit app

Chat interface

Option to select from distinct subreddits

Flexible retreiver configurations

Option to choose from best open source LLMs (including the latest Llama3) hosted on *Groq cloud*

Custom prompt template option for generating summarised answers

# Acknowledgements

We would like to thank:

Steven Gubkin and Roman Holowinsky @ Erdos Institute

Jason Morgan @ Aware