Search Engine

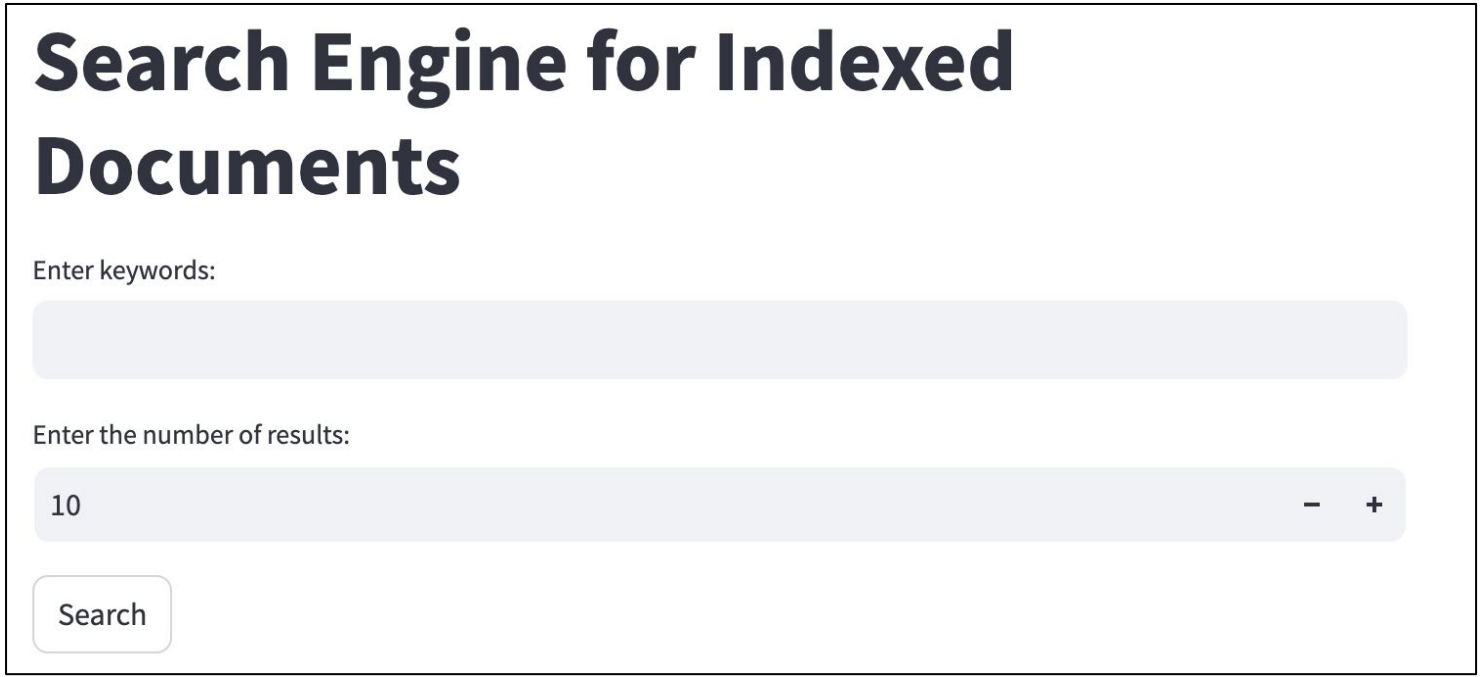
For CS Papers

Miaomiao Zhang

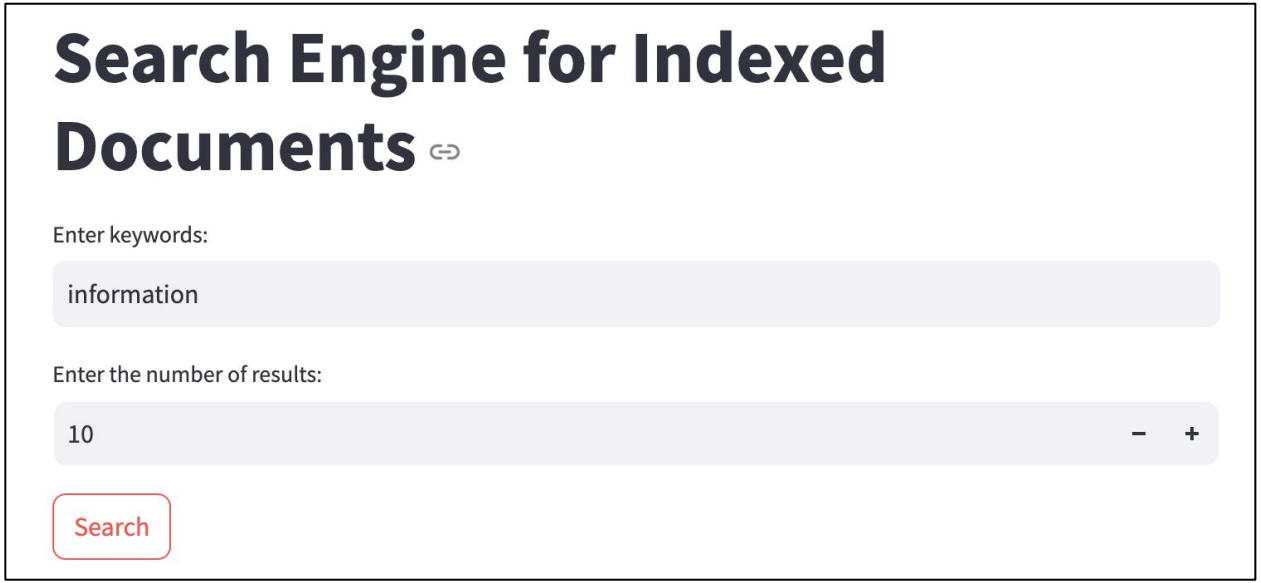
Asmita Prabhakar

Section Title

Output Screen



Output screen- 2



Result

Found 10 matching documents.

1. C:/Users/atulp/Downloads/Dataset_IRS\2\9958\10.1.1.2.9958.txt

Title: What Is Information Discovery About?

2. C:/Users/atulp/Downloads/Dataset_IRS\2\8274\10.1.1.2.8274.txt

Title: Co-Constructive Information Systems

3. C:/Users/atulp/Downloads/Dataset_IRS\2\7252\10.1.1.2.7252.txt

Title: IMPACT OF INFORMATION REUSABILITY ON INFORMATION SYSTEMS

4. C:/Users/atulp/Downloads/Dataset_IRS\2\2550\10.1.1.2.2550.txt

Title: Information Retrieval and Situation Theory

5. C:/Users/atulp/Downloads/Dataset_IRS\2\8368\10.1.1.2.8368.txt

Title: Consultative

6. C:/Users/atulp/Downloads/Dataset_IRS\2\4989\10.1.1.2.4989.txt

Title: From Data to Actionable Knowledge and Decision 1

7. C:/Users/atulp/Downloads/Dataset_IRS\2\9379\10.1.1.2.9379.txt

Title: A Firm's Optimal Number of Bank Relationships and

8. C:/Users/atulp/Downloads/Dataset_IRS\2\4589\10.1.1.2.4589.txt

Title: Multi-Stage Information Acquisition in Auction Design

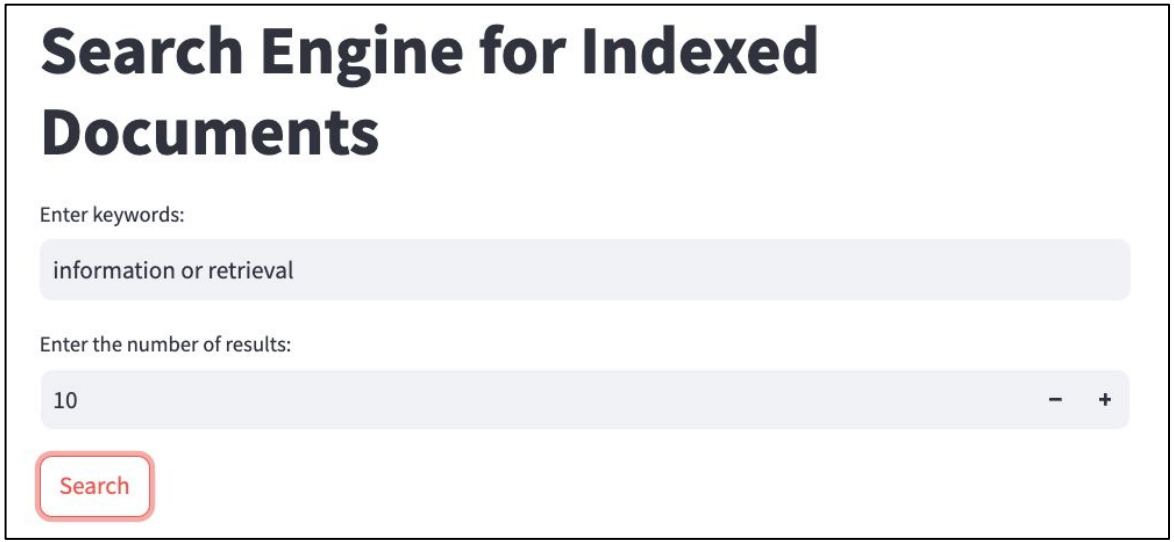
9. C:/Users/atulp/Downloads/Dataset_IRS\2\3539\10.1.1.2.3539.txt

Title: Information Systems Frontiers 5:1, 47–61, 2003

10. C:/Users/atulp/Downloads/Dataset_IRS\2\123\10.1.1.2.123.txt

Title: District Level Spatial Information:

Output Screen-3



Result

1. C:/Users/atulp/Downloads/Dataset_IRS\2\8485\10.1.1.2.8485.txt

Title: A Risk Minimization Framework for Information Retrieval

2. C:/Users/atulp/Downloads/Dataset_IRS\2\3303\10.1.1.2.3303.txt

Title: ELECTRONIC WORKSHOPS IN COMPUTING

3. C:/Users/atulp/Downloads/Dataset_IRS\2\4009\10.1.1.2.4009.txt

Title: Parallel Visual Information Retrieval in VizIR

4. C:/Users/atulp/Downloads/Dataset_IRS\2\2550\10.1.1.2.2550.txt

Title: Information Retrieval and Situation Theory

5. C:/Users/atulp/Downloads/Dataset_IRS\2\6485\10.1.1.2.6485.txt

Title: ABSTRACT

6. C:/Users/atulp/Downloads/Dataset_IRS\2\1601\10.1.1.2.1601.txt

Title: Evaluating High Accuracy Retrieval Techniques

7. C:/Users/atulp/Downloads/Dataset_IRS\2\6363\10.1.1.2.6363.txt

Title: Language-dependent and Language-independent

8. C:/Users/atulp/Downloads/Dataset_IRS\2\8055\10.1.1.2.8055.txt

Title: CLARITY: CROSS LANGUAGE INFORMATION

9. C:/Users/atulp/Downloads/Dataset_IRS\2\5260\10.1.1.2.5260.txt

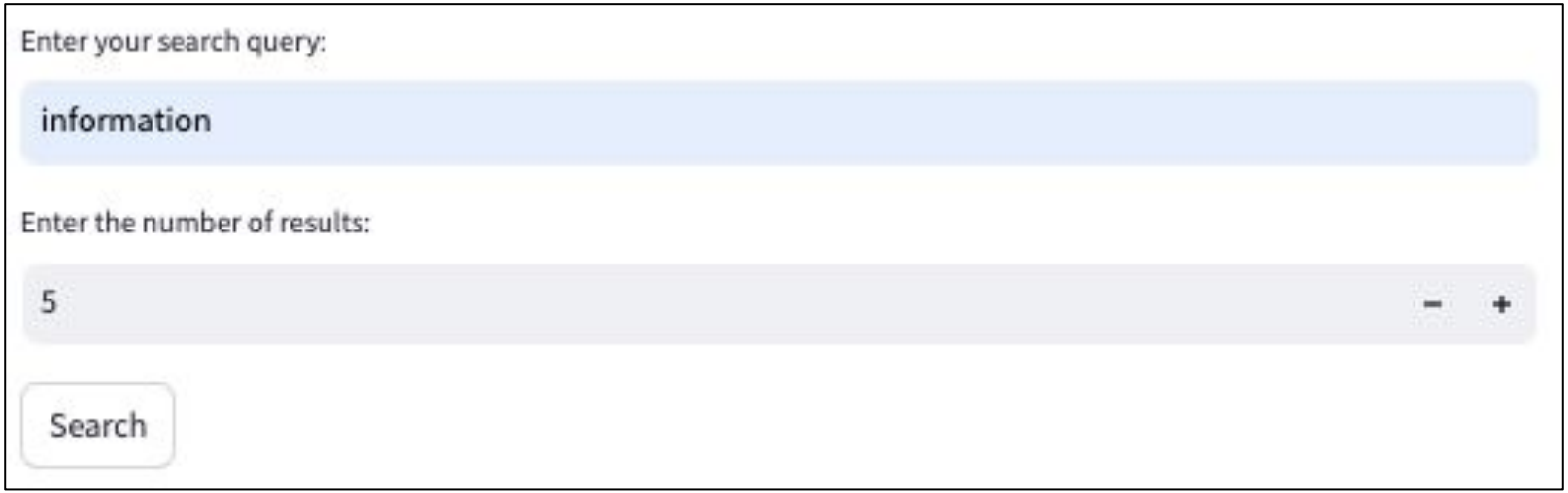
Title: The Importance of Morphological Normalization

10. C:/Users/atulp/Downloads/Dataset_IRS\2\2492\10.1.1.2.2492.txt

Title: Computational Semantics and

Output Screen -4

Output Screen -5



Result

Top 5 results for 'information':

Document 4251: (Score: 0.4395)

File Path: citeseer2/8592/10.1.1.2.8592.txt

A Contribution to the Theory of Information Acquisition in Financial Markets Marc-Andreas Muendler • University of California, Berkeley September 16, 2000 Abstract In order to explore the incentives for information acquisition in financial markets, a model of the joint information and portfolio choice is developed. Investors are allowed to acquire a number of signals that inform about a risky asset's dividend, and "informational efficiency" is defined as a social planner's preferred signal allo

Document 9540: (Score: 0.4215)

File Path: citeseer2/6404/10.1.1.2.6404.txt

An Information Product Approach for Total Information Awareness Richard Wang Thomas Allen Wesley Harris Stuart Madnick rwang@mit.edu tallen@mit.edu weslhar@mit.edu smadnick@mit.edu Abstract--To fight terrorism successfully, the quality of data must be considered to avoid garbage-in-garbage-out. Research has shown that data quality (DQ) goes beyond accuracy to include dimensions such as believability, timeliness, and accessibility. In collecting, processing, and analyzing a much broader array of

Document 5482: (Score: 0.4106)

File Path: citeseer2/3539/10.1.1.2.3539.txt

Information Systems Frontiers 5:1, 47–61, 2003 C○ 2003 Kluwer Academic Publishers. Manufactured in The Netherlands. Predicting the Future Abstract. We present a novel methodology for predicting future outcomes that uses small numbers of individuals participating in an imperfect information market. By determining their risk attitudes and performing a nonlinear aggregation of their predictions, we are able to assess the probability of the future outcome of an uncertain event and compare it to bot

Document 5383: (Score: 0.3998)

File Path: citeseer2/6421/10.1.1.2.6421.txt

in Artificial Life VIII, Standish, Abbass, Bedau (eds)(MIT Press) 2002. pp 345–349 1 Meaningful Information, Sensor Evolution, and the Temporal Horizon of Embodied Organisms Chrystopher L. Nehaniv 1,2 , Daniel Polani 1,2 , Kerstin Dautenhahn 1 , René te Boekhorst 1 , and Lola Cañamero 1 1 Adaptive Systems Research Group and 2 Algorithms Research Group Faculty of Engineering & Information Sciences, University of Hertfordshire, Hatfield Herts AL10 9AB, U.K. Abstract We survey and outline how an a

Document 3522: (Score: 0.3921)

File Path: citeseer2/9339/10.1.1.2.9339.txt

DUNEDIN NEW ZEALAND Assessing Prediction Systems Barbara Kitchenham Stephen MacDonell Lesley Pickard Martin Shepperd The Information Science Discussion Paper Series Number 99/14 June 1999 ISSN 1172-6024 University of Otago Department of Information Science The Department of Information Science is one of six departments that make up the Division of Commerce at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc de

Implementation & explanation

- Indexed search
- Semantic search

Preparation

- Machine: University server delta.cs(GPU)
- Dataset: citeseer2

  Size: 9999 documents

  Type: scientific publications
- Packages:

  whoosh: similar to Lucene for full-text search

  streamlit: user interface in python

  SentenceTransformer: a pre-trained model to encode documents for semantic search

  os: open files, read files, create directory for indexed files

SentenceTransformer

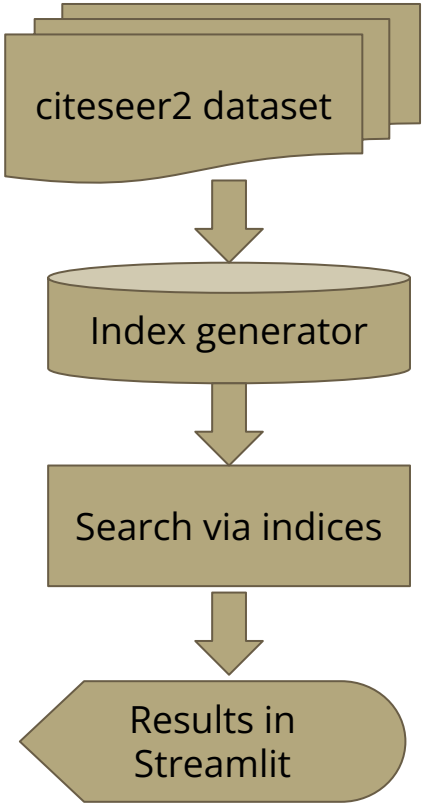
- **One Embedding per Document**

  The model outputs a single embedding that represents the entire document
- **Semantic Representation:**

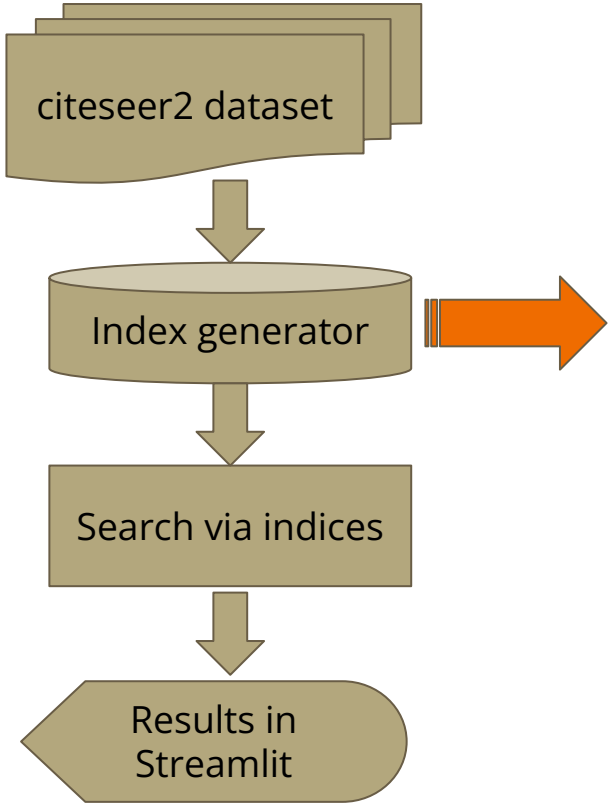
  This embedding captures the context and meaning of the document, not just individual words.
- **Fixed Size:**

  384 dimensions in the “all-MiniLM-L6-v2” model

Indexed search



Indexed search




```python
#Define the schema for the index
def create_schema():
    return Schema(
        path=ID(stored=True, unique=True),
        title=TEXT(stored=True),
        contents=TEXT(stored=True, analyzer=StandardAnalyzer())
    )
```

Tokenization and Term Creation:

For example:

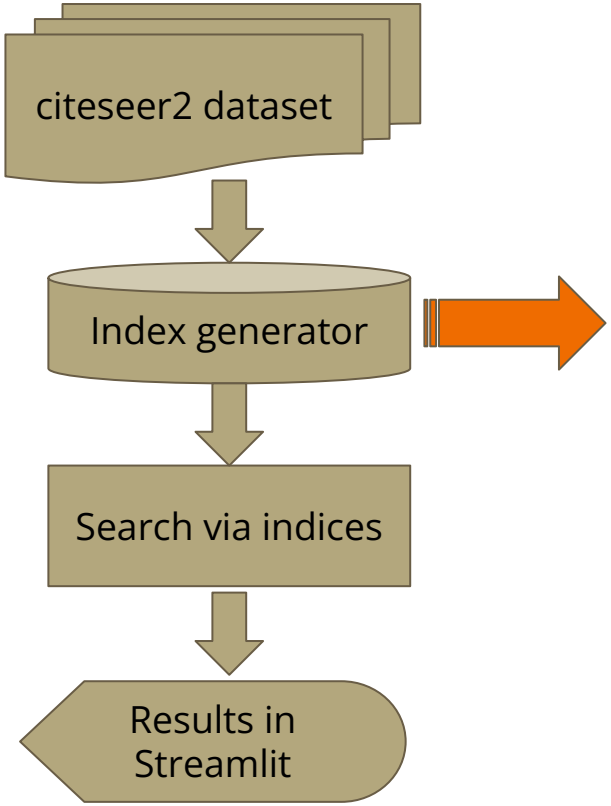
Document contains :"The quick brown fox jumps over the lazy dog."

Create index:

Term: "quick" -> Appears in Documents: [doc1, doc5, doc12]

Term: "dog" -> Appears in Documents: [doc2, doc4, doc12]

Indexed search




```python
#Index all documents in a given directory
def index_docs(writer, docs_path):
    counter = 0
    for root, _, files in os.walk(docs_path):
        for file in files:
            file_path = os.path.join(root, file)
            with open(file_path, 'r', encoding="utf-8") as f:
                title = f.readline().strip()  #First line as title
                contents = f.read()  #The rest of the file is content

                #Create a document and add fields
                writer.add_document(
                    path=file_path,
                    title=title,
                    contents=contents
                )
                counter += 1
                if counter % 1000 == 0:
                    print(f"Indexed {counter}-th file: {file}")
    print(f"Total files indexed: {counter}")
```

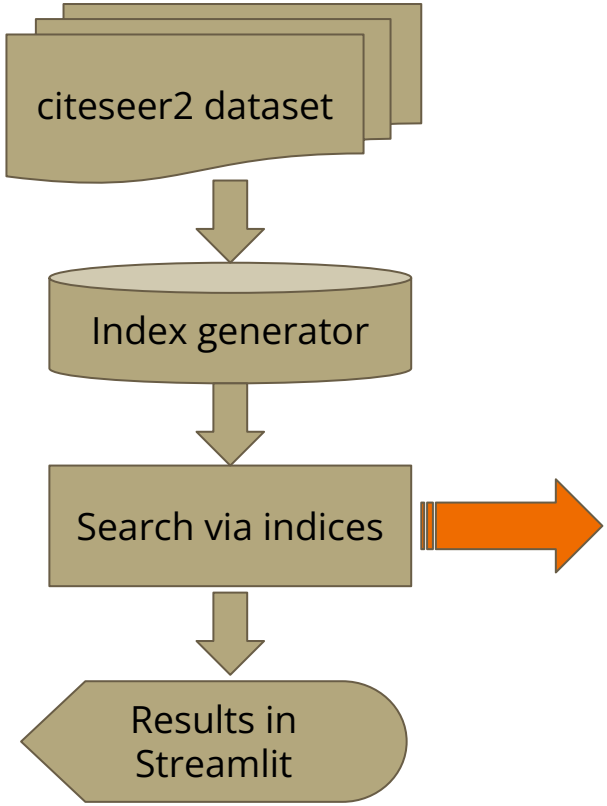
In main() function:

```python
#Create schema and index directory
schema = create_schema()
idx = create_in(index_path, schema)
writer = AsyncWriter(idx)

#Index the documents
index_docs(writer, docs_path)
```

Built-in functions in whoosh

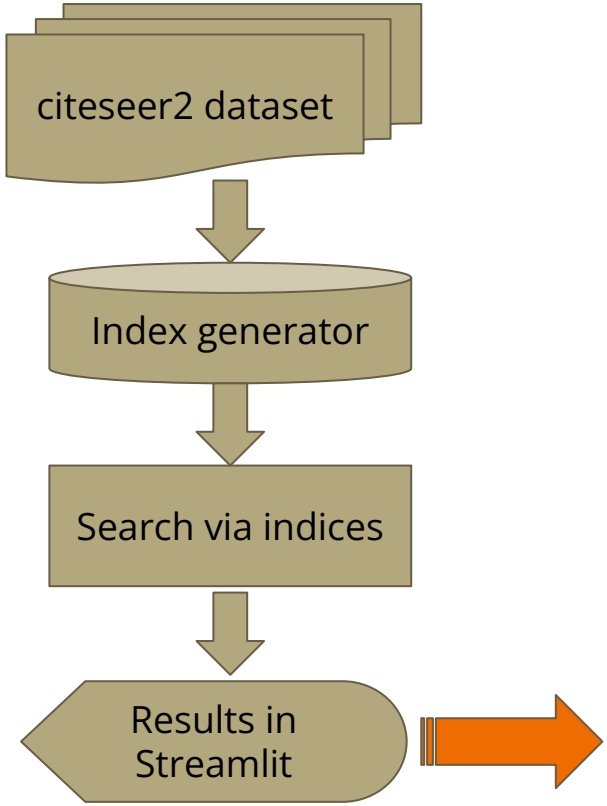
Indexed search




```
from whoosh.index import open_dir
from whoosh.qparser import QueryParser
```

```
def search_indexed_docs(keywords, num_of_results):
    # Define the path to the index directory (you need to adjust this path)
    index_dir = "/home/zhang3s2/workspace/8380/index"
    # Open the index
    ix = open_dir(index_dir)
    # Create a searcher object
    with ix.searcher() as searcher:
        # Define the analyzer and parser for the search query
        parser = QueryParser("contents", ix.schema)
        query = parser.parse(keywords)
        # Perform the search, limit the results
        results = searcher.search(query, limit=num_of_results)
        # Prepare the results
        result_list = []
        for i, result in enumerate(results):
            result_dict = {
                "path": result['path'],
                "title": result.get('title')
            }
            result_list.append(result_dict)
        return result_list
```

Indexed search

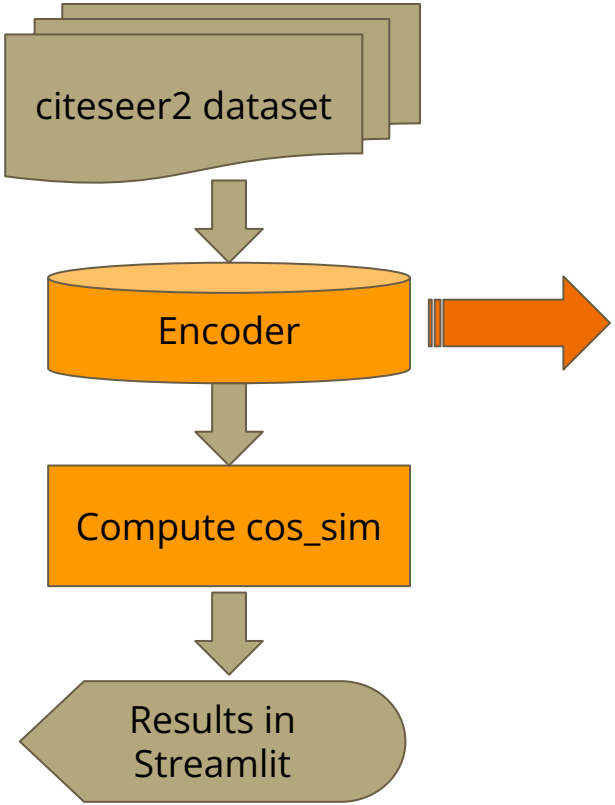



```python
# Streamlit app
st.title('Search Engine for Indexed Documents')

# Input fields for keywords and number of results
keywords = st.text_input('Enter keywords:')
num_of_results = st.number_input('Enter the number of results:', min_value=1,
                                 max_value=100, value=10)
```

```python
# Display the results
st.write(f"Found {len(search_results)} matching documents.")
for i, result in enumerate(search_results):
    st.write(f"{i+1}. {result['path']}")
    if result['title']:
        st.write(f"   Title: {result['title']}")
```

Semantic search

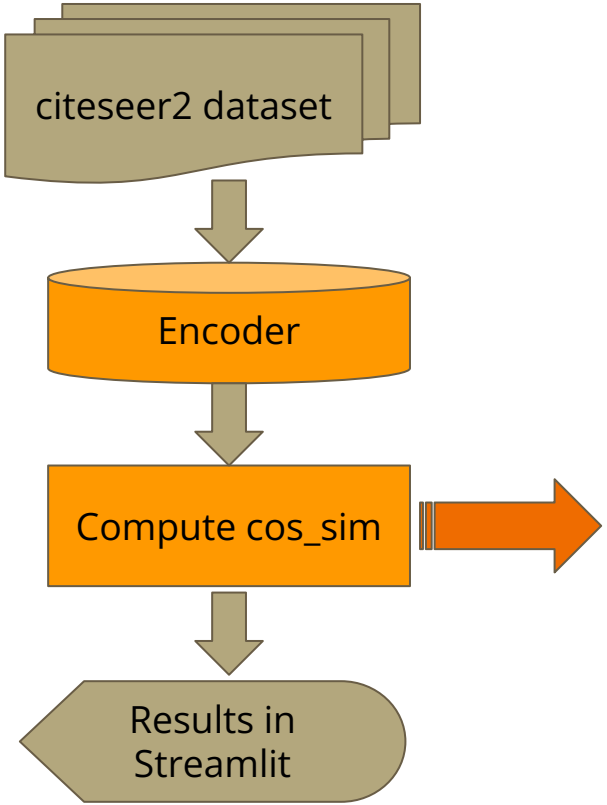



```python
import os
from sentence_transformers import SentenceTransformer
import torch
```

```python
# Encode the document contents
print(f"Encoding {len(documents)} documents...")
doc_embeddings = model.encode(documents, convert_to_tensor=True)

# Save the embeddings to a file for future use (optional)
torch.save(doc_embeddings, 'doc_embeddings.pt')
```

Semantic search



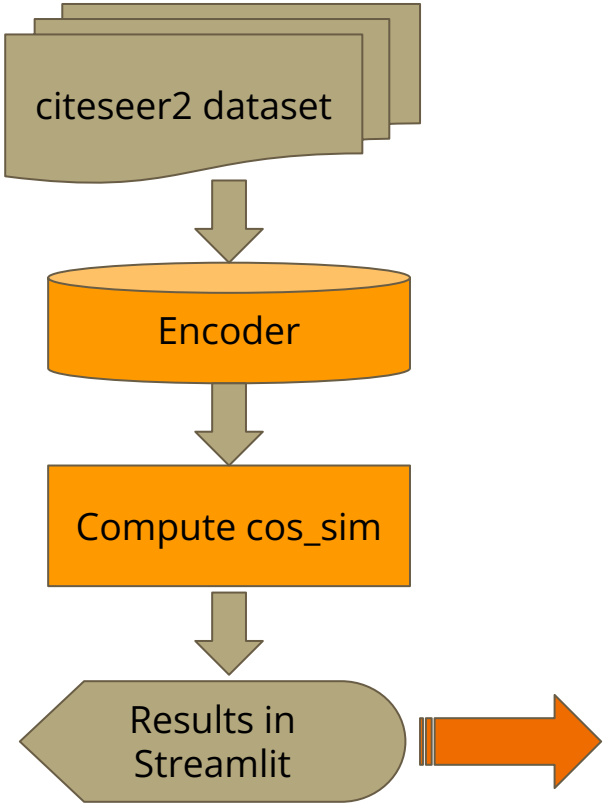

```python
# Load precomputed embeddings if they exist
doc_embeddings = torch.load('doc_embeddings.pt')
```

```python
# Encode the query
query_embedding = model.encode(query, convert_to_tensor=True)

# Compute cosine similarity between query and document embeddings
cosine_scores = util.pytorch_cos_sim(query_embedding, doc_embeddings)[0]

# Get the top results
top_results = torch.topk(cosine_scores, k=num_of_results)
```

Semantic search




```
# Display the results
st.write(f"Top {num_of_results} results for '{query}':")
for score, idx in zip(top_results.values, top_results.indices):
    st.write(f"Document {idx+1}: (Score: {score.item():.4f})")
    st.write(f"File Path: {doc_paths[idx]}")  # Show file path
    st.write(documents[idx][:500])  # Display the first 500 characters
```

Future work

- Collect more accessible datasets
- Add some more features:

  Ranking

  Classification

  Spell correction

  Search via sentences(remove stop words)...
- User interface:

  Users able to select a dataset from a list

  ......

Thanks!