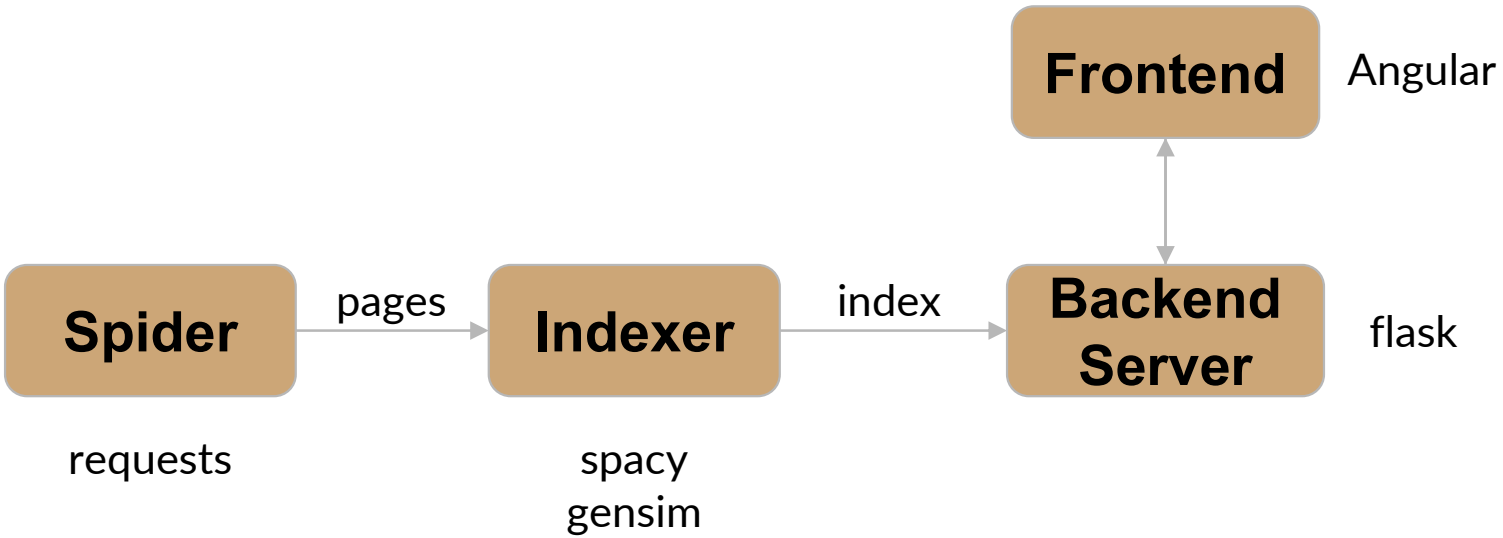
Simple Search Engine

By Group 11
—Zhang Hexiao, Ma Xiaolei, Zhao Siqi

Design Structure Diagram



File Structures for Index

- Considering the relatively small number of pages to be indexed, the backend did not use database.
- Index objects are directly saved as a pickle file
    - Pickle is a standard library in python that can save objects as binary files.

Algorithm Used

- Text preprocessing:
  - Tokenization: Spacy's tokenizer
  - Stopword removal: Custom list of stopwords
  - Lemmatization: Spacy's lemmatizer
  - Index of phrases: Spacy's named entity extraction

Algorithm Used

- Search engine:
  - Inverted index creation:
    - using dictionaries
    - keys: terms in the corpus
    - values: lists of document IDs where those terms appear.
  - Query processing and ranking:
    - TF-IDF
      - weight the importance of each word in a document
      - which is based on its frequency in the document and its rarity in the corpus
    - Cosine Similarity
      - compute document similarities based on TF-IDF vectors

Installation Procedure

- Environments:
  - Python 3.9 or newer
  - Nodejs
- Building:
  - For Backend:
    - cd backend
    - pip install -r requirements.txt
    - python -m spacy download en_core_web_sm
    - flask run
  - For Frontend:
    - cd frontend
    - npm install
    - npm run start

Testing of Functions--Spider

```
58      unvisited = [ROOT]
59      updated = False
60      while unvisited:
61          url = unvisited.pop()
62          r = session.get(url, headers=HEADERS)
63          last_mod_time = r.headers.get("Last-Modified", r.headers["Date"])
64          if url in visited and is_not_newer(last_mod_time, visited[url].last_mod_time):
65              continue
66          print(f"page {url} added")
67          updated = True  # page updated, should commit to files
68
69          title = r.html.find("head > title", first=True).text
70          body = r.html.find("body", first=True).text
71          keywords = extract_keywords(title, body)
72          size = r.headers["Content-Length"]
73          children = set(urljoin(url, l) for l in r.html.links)
74          page = Page(
75              id=next(counter),
76              url=url,
77              title=title,
78              body=body,
79              size=size,
80              last_mod_time=last_mod_time,
81              keywords=keywords,
82              children_url=list(children),
83          )
84          visited[url] = page
85          unvisited.extend(children)
```

breadth-first

Testing of Functions--Spider

```
(venv) PS D:\Documents\PG\5930-Search Engine\project\simple-search-engine\backend> python .\spider.py
page https://www.cse.ust.hk/~kwtleung/COMP4321/testpage.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/bbc.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/bbc1.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/bbc3.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/bbc2.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/cnn.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/cnn1.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/news/cnn2.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/UG.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/PG.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/Movie.htm added
page https://www.cse.ust.hk/~kwtleung/COMP4321/Movie/131.html added
page https://www.cse.ust.hk/~kwtleung/COMP4321/Movie/269.html added
page https://www.cse.ust.hk/~kwtleung/COMP4321/Movie/120.html added
```

Testing of Functions--Spider

- Structure used to store webpage info in output.json

```python
class Page:
    id: int
    url: str
    title: str
    body: str
    last_mod_time: str
    size: str
    keywords: list[tuple[str, int]] = field(default_factory=list)
    children_url: list[str] = field(default_factory=list)
    children_id: list[int] = field(default_factory=list)
    parents_url: list[str] = field(default_factory=list)
    parents_id: list[int] = field(default_factory=list)
```

output.json


```
[
  {
    "id": 0,
    "url": "https://www.cse.ust.hk/~kwtleung/COMP4321/testpage.htm",
    "title": "Test page",
    "body": "This is the Test page for a crawler\nBefore getting the
Admission of CSE department of HKUST,\nYou should read through these
international news and these books.\nHere is my Movie List (\nNew\n)",
    "last_mod_time": "Thu, 16 Jun 2022 08:47:33 GMT",
    "size": "603",
    "keywords": [
      [
        "crawler",
        1
      ],
      [
        "admission",
        1
      ],
      [
        "cse",
        1
      ],
      [
        "department",
        1
      ],
      [
        "hkust",
        1
      ]
    ],
    "children_url": [
      "https://www.cse.ust.hk/~kwtleung/COMP4321/Movie.htm",
      "https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm",
      "https://www.cse.ust.hk/~kwtleung/COMP4321/books.htm",
      "https://www.cse.ust.hk/~kwtleung/COMP4321/news.htm"
    ],
    "children_id": [
      16,
      9,
      13,
      1
    ],
    "parents_url": [
      "https://www.cse.ust.hk/~kwtleung/COMP4321/Movie.htm",
      "https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm",
      "https://www.cse.ust.hk/~kwtleung/COMP4321/books.htm",
      "https://www.cse.ust.hk/~kwtleung/COMP4321/news.htm"
    ],
    "parents_id": [
      16,
      13,
      9,
      1
    ]
  },
  {
    "id": 1,
```

Testing of Functions--Indexer

- Library and model included

```
2
3   STOPWORDS = load_stopwords(STOPWORDS_PATH)
4   NLP = spacy.load("en_core_web_sm")
5   PUNC_ESCAPER = re.compile(r"[{}]+".format(punctuation))
6
```

- Tokenization Process

```
def get_tokens(text: str, pharse: bool = True) -> list[str]:
    text = PUNC_ESCAPER.sub(" ", text)
    doc = NLP(text)
    stems = [token.lemma_.lower() for token in doc]
    tokens = [token for token in stems if not escape(token)]
    if pharse:
        tokens.extend(
            chunk.text
            for chunk in doc.noun_chunks
            if chunk.text.count(" ") > 0 and not escape(chunk.text)
        )
    return tokens
```

Testing of Functions--Indexer

- What we saved in our Index object:
  1. Pages
  2. Page to Words dict
  3. Words dictionary (string to word's ID)
  4. Word to Pages dict (Inverted Index)
  5. Precomputed TFxIDF vectors

Testing of Functions--Retrieval Function

- Retrieval process:

  query (str) => tokens (list[str]) => word id list (list[int]) => TFxIDF vector

```
def search(self, query: str, topk: int = 50):  # -> list[Page]:
    tokens = get_tokens(query)
    query_bow = self.dictionary.doc2bow(tokens)
    query_tfidf = self.tfidf[query_bow]
    bsim = self.bindex[query_tfidf]
    tsim = self.tindex[query_tfidf]
    sim: np.ndarray = cal_scores(tsim, bsim)
    indices = reversed(np.argsort(sim)[-topk:])
    return [
        {"score": sim[id], **asdict(self.pages[id])}
        for id in indices
        if sim[id] > 0
    ]
```

- Mechanism that favors match in titles:

  total_score = λ* title_score + body_score, λ = 5 in our implementation.

Testing of Functions--Retrieval Function

127.0.0.1:5000/query?query=hkust

[{"body":"CSE department of HKUST\nPG Admission\nUG Admission\nBack to main","children_id":[312,0,311],"children_url":
["https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/UG.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/PG.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/testpage.htm"],"id":310,"keywords":[["cse",2],["
["admission",2],["main",1]],"last_mod_time":"Thu, 16 Jun 2022 08:47:33 GMT","parents_id":[312,311,0],"parents_url":
["https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/UG.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/PG.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/testpage.htm"],"score":2.709205150604248,"size":
department of HKUST","url":"https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm"},{"body":"This is the Test page for a crawler\nBefore getting the Admission of CSE department of HKUST,\nYou should read throug
and these books.\nHere is my Movie List (\nNew\n)","children_id":[313,1,310,9],"children_url":
["https://www.cse.ust.hk/~kwtleung/COMP4321/books.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/Movie.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/ne
[["crawler",1],["admission",1],["cse",1],["department",1],["hkust",1]],"last_mod_time":"Thu, 16 Jun 2022 08:47:33 GMT","parents_id":[313,310,9,1],"parents_url":
["https://www.cse.ust.hk/~kwtleung/COMP4321/books.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/Movie.htm","https://www.cse.ust.hk/~kwtleung/COMP4321/ne
934764862,"size":"603","title":"Test page","url":"https://www.cse.ust.hk/~kwtleung/COMP4321/testpage.htm"},{"body":"Program Information\nThe Department of Computer Science at HKUST offers two degree programs fo
interested in learning about computing and information technology:\n1. COMP: BEng in Computer Science\nThis is our general undergraduate program that provides broad education in all areas of Computer Science.\n
Science (Information Engineering)\nThis is a specialized undergraduate degree program focusing on the areas of multimedia computing and computer networking.\nWhile other Information Engineering programs in Hong
Electrical and Electronic Engineering, our CSIE program is unique in its emphasis on software. In fact, the CSIE program also satisfies all the general requirements of a typical Computer Science program. It has
with the specific needs of Hong Kong in mind.\nBesides, the Department, jointly with the Department of Electrical and Electronic Engineering, also offers the BEng in Computer Engineering.\nA Spectrum of IT-Rela
offered by our Department and the Department of Electrical and Electronic Engineering can be best viewed as the following spectrum:\nSoftware<---------->Hardware\nCOMP\u00a0\u00a0\u00a0 CSIE\u00a0\
CPEG\u00a0\u00a0\u00a0 EEIC\u00a0\u00a0\u00a0 ELEC\nGenerally speaking, the emphasis shifts from \"soft\" to \"hard\" as we move across the spectrum. The different programs suit different student interests and
Opportunities\nBesides the regular study programs, Computer Science students at HKUST also enjoy a number of other opportunities, including the following:\nOverseas exchange program:\nOne-year full-time oversea
HKUST's tuition fee\nWork-study program:\nOne-year full-time work outside HKUST after two years of study\nIndustrial training:\nTraining modules during winter and summer breaks on specific practical skills\nEng
training in written and conversational English, including presentation skills\nFinal year projects:\nOne year project in an area of specialization under the guidance of faculty advisors\nStudent tutors:\nSecond
helpers providing help to students in first-year courses\nOur Graduates\nAbout 20% of our graduates choose to pursue postgraduate study immediately after graduation. While many of them stayed in HKUST for MPhil
study, some went abroad to study at top universities, including Stanford University, University of Southern California, University of Toronto, University of Texas at Austin, and University of Illinois at Urbana
choose to work found excellent jobs. While graduates from related disciplines also look for computing related jobs, Computer Science graduates clearly have great advantages after completing a comprehensive prog
training in fundamentals and many emerging areas of IT and computing.\nInquiries\nStudents who need advice or assistance on application procedures, choice of programs, entrance requirements or other related mat
us:","children_id":[310],"children_url":["https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm"],"id":312,"keywords":[["program",16],["science",7],["student",7],["study",7],["engineering",6]],"last_mod_time":"
GMT","parents_id":[310],"parents_url":["https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm"],"score":0.15491583943367004,"size":"4070","title":"UG","url":"https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/UG
Program\nApplicants for admission to the postgraduate programs are required to have\u00a0 completed, by the time they enter HKUST, a bachelor's degree in computer science or a related science or engineering fie
postgraduate program must demonstrate competence in the following core computer science areas:\n\nComputer Organization\nPrinciples of Programming Languages\nPrinciples of Systems Software\u00a0 (Operating Syst
of Algorithms\nIn addition, students admitted into the PhD program must demonstrate competence in:\nTheory of Computation\n\nCompetence in these areas is appraised during the admission process. A student whose
background is deemed inadequate in any of the above areas may be admitted on a provisional basis and will be required to take remedial courses prescribed by the Postgraduate Studies Committee and obtain a grade
courses. Such additional requirements will be stipulated in the individual offer of admission. Deficiencies in any core computer science area can be made up concurrently with postgraduate work.\nApplicants are
submit:\n\nofficially certified academic transcripts of undergraduate studies (and postgraduate studies, if any);\nresults of the computer science subject Graduate Record (GRE) (required for PhD applicants only
submitted in the year after admission);\nresults of an English proficiency test (see below);\ntwo letters of recommendation: a one-page statement of plan for the postgraduate study; and\na completed application
admission.\n\nFor admission into the MSc and MPhil programs, results of standard examinations, such as the computer science subject Graduate Record Examination (GRE), may be helpful in determining suitability f
submitted as supplementary information, if available.\nAll students admitted into a postgraduate program must demonstrate sufficient competence in English. Students whose previous degrees are from an institutio
language of instruction is other than English should submit the results of an English proficiency test (such as TOEFL) together with their application. The English proficiency test must have been taken within t
application. Notwithstanding the above, the Department may require newly admitted students to undergo language proficiency assessment and, if deemed necessary, to take English language enhancement courses.\nAdm
postgraduate programs will be based on the applicant's academic record as well as the research needs of the Department.","children_id":[310],"children_url":["https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.ht
[["postgraduate",9],["admission",8],["program",7],["science",7],["student",6]],"last_mod_time":"Thu, 16 Jun 2022 08:47:41 GMT","parents_id":[310],"parents_url":
["https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm"],"score":0.03579392656683922,"size":"3267","title":"PG","url":"https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/PG.htm"}]

Testing of Functions--Retrieval Function & Indexer

```python
if __name__ == "__main__":
    index = load_index()
    ret = index.search("HKUST")
    for r in ret:
        print(r["score"], r["url"], r["title"])
```

```
(venv) PS D:\Documents\PG\5930-Search Engine\project\simple-search-engine\backend> python .\index.py
Loaded data at data/output.json
index loaded from data/index.pkl
2.709205150604248 https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse.htm CSE department of HKUST
0.22966603934764862 https://www.cse.ust.hk/~kwtleung/COMP4321/testpage.htm Test page
0.15491583943367004 https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/UG.htm UG
0.03579392656683922 https://www.cse.ust.hk/~kwtleung/COMP4321/ust_cse/PG.htm PG
```

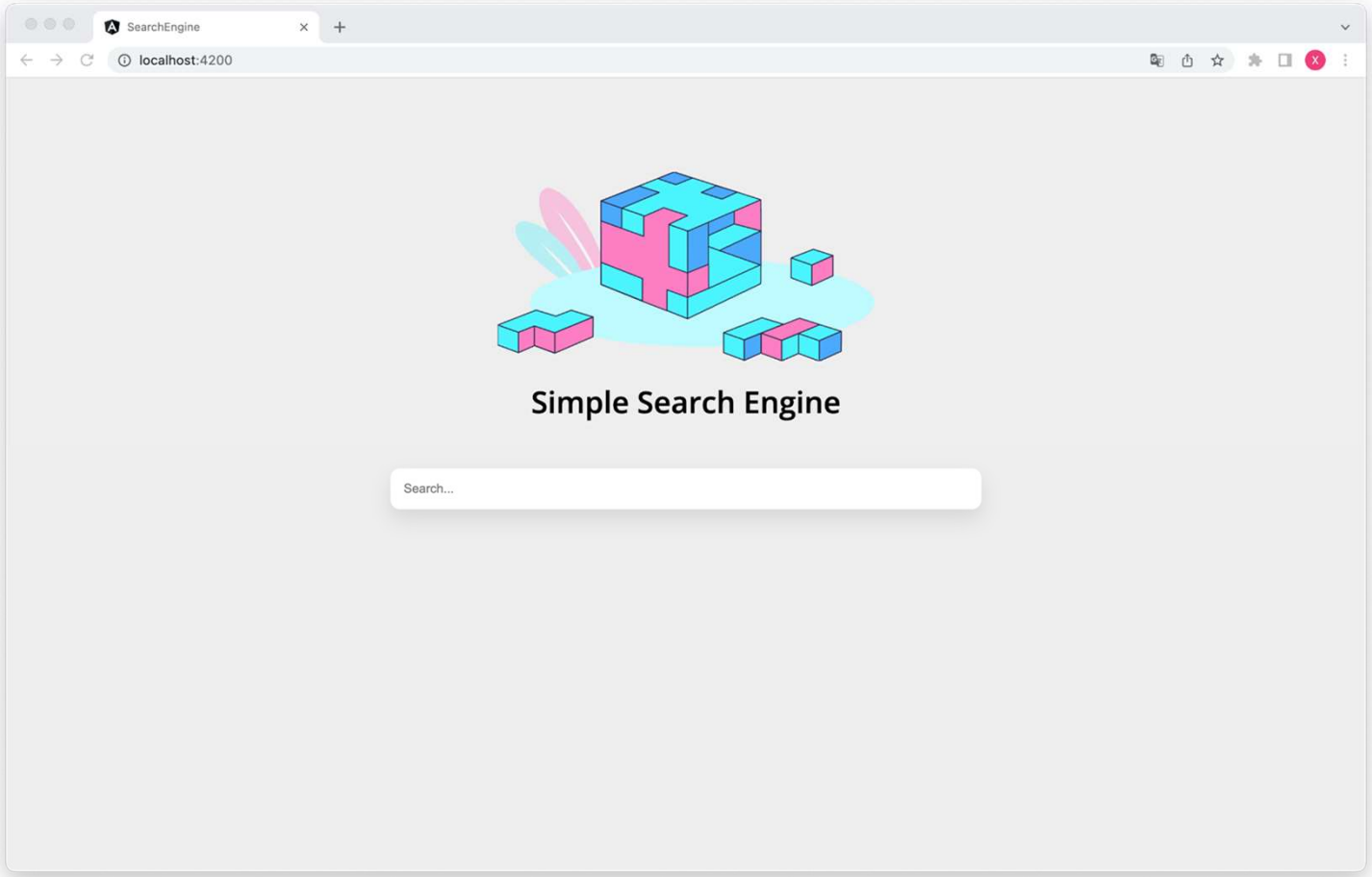
Testing of Functions--Indexer

- Tokens → BOW → TF-IDF vector

```python
from gensim.corpora import Dictionary
from gensim.models import TfidfModel
from gensim.similarities import SparseMatrixSimilarity

from spider import Page, run_spider
from utils import INDEX_PATH, get_tokens, load_pkl, save_pkl


class Index:
    def __init__(self, pages: list[Page]) -> None:
        self.pages = pages
        bodies = [get_tokens(page.body) for page in pages]
        titles = [get_tokens(page.title) for page in pages]

        self.dictionary = Dictionary(bodies + titles)
        self.bodies_bow = [self.dictionary.doc2bow(body) for body in bodies]
        self.titles_bow = [self.dictionary.doc2bow(title) for title in titles]
        corpus = self.bodies_bow + self.titles_bow
        self.tfidf = TfidfModel(corpus, self.dictionary)
        self.bindex = SparseMatrixSimilarity(
            self.tfidf[self.bodies_bow], len(self.dictionary)
        )
        self.tindex = SparseMatrixSimilarity(
            self.tfidf[self.titles_bow], len(self.dictionary)
        )
```

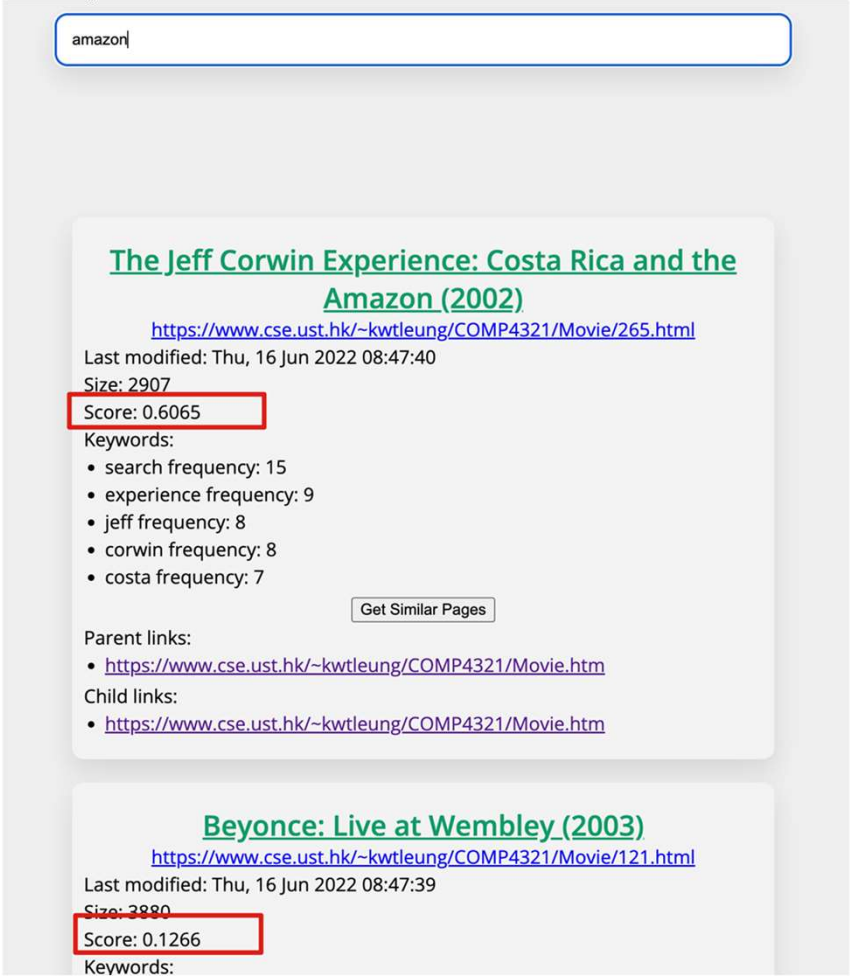
Testing of Functions--Web Interface

- Main Page



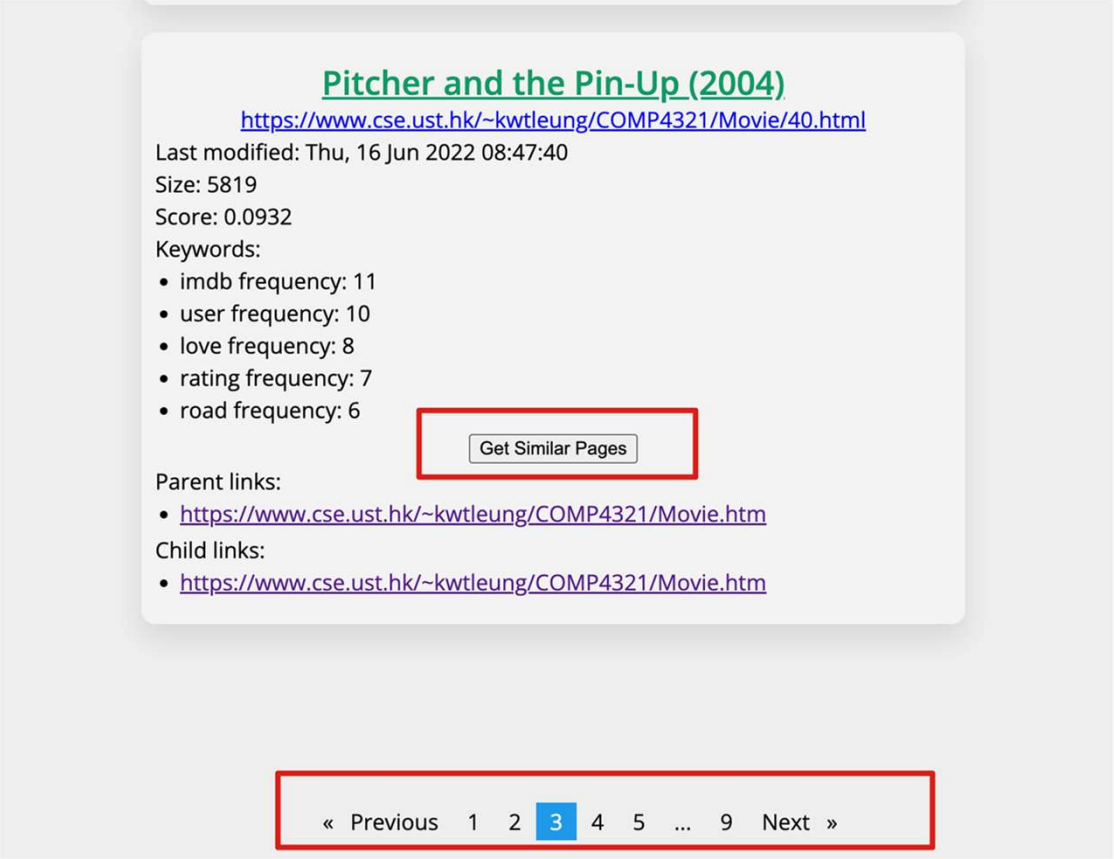
Testing of Functions--Web Interface

- Searching result
  - In descending order of document score
  - Title and URL hyperlinked to actual page on the remote server
  - Displays up to 5 most frequent stemmed keywords with occurrence frequencies



Highlight of Features

- Pagination: still displaying on the same page, but allow users to jump between pages
- Get similar pages: clicking the button to use the top 5 most frequent keywords as the query for a new search.



Strengths and Weaknesses

- Strengths:
  - Pre-computed TFxIDF, stable and quick for documents with small size
  - Developed with the latest frontend framework Angular
  - Simple and elegant UI, easy to learn and follow
- Weaknesses:
  - Did not consider costs for recomputation of TFxIDF when there is update in document collections
  - Mainly focuses on static pages

Possible Improvements

- Smarter algorithm that could update TFxIDF score when new documents come in
- Consider including other ranking algorithms based on links such as PageRank
- Including dialog box(es) to enable conversations with AI tools such as ChatGPT
- Including the function to sort resulting data by meta data(such as last modified date)