# CS 121 A3 Search Engine Report

Group: 106: Claire Li, Emily Sun, Leyang Cheng, Richard Youn

**Summary of Improvement(all extra credit are concerned)**

| Problem Type | Original Behavior | Optimized Behavior |
|---|---|---|
| Keyword Position | Keywords matched anywhere in the body, with no importance distinction | -Introduced a flag system: Title > URL > Body, prioritizing key pages<br>-Combined with **proximity score** to favor documents where keywords appear closer |
| Spelling Robustness | Misspelled or deformed terms (e.g., “enginerring”) returned no results | Integrated N-gram fuzzy matching with a 200ms timeout |
| Duplicate Content | Both exact and near-duplicate pages retained | Removed exact + near duplicates using SimHash and content comparison |
| Empty/No-title Pages | Documents without meaningful titles polluted top results | Filtered out documents with [No Title] or low-quality metadata |
| Dead URLs | Broken or unreachable links included | Integrated crawler checks to eliminate dead or invalid URLs |
| Ranking Strategy | Only use tf-idf score | Combined PageRank + HITS + normalized flag score for holistic ranking |
| Boolean AND Search | Performed naive Boolean AND without robustness | Now supports stemmed AND search + N-gram fallback for robustness |
| User Interface | Simple Interface without design | Using streamlit for more beautiful UI/ highlight keyword/allow pagination/ able to see viewing history |
| Web Crawling | Broken html and page that are unable to open are included | We use crawler to filter out those broken html. Now, our inverted index is build on openable page. |

# Our Progress

## MileStone1

### Stage 1: Inverted Index Construction And Basic Boolean Retrieval

- Built a custom inverted index to map each term to a list of postings, storing document ID, term frequency (TF), and word positions.
- Created a doc_meta_finished.json file to maintain a consistent mapping between URLs and document IDs.
- Implemented basic **Boolean search** (AND queries) using exact term matching.

## MileStone2

### Stage 2: TF-IDF Ranking Integration

- Calculated and stored **TF-IDF weights** for each term during indexing.
- Summed TF-IDF values across matched query terms to form a basic relevance score.
- Ensured position-aware indexing for potential proximity analysis.

cristina lopes

machine learning

ACM

master of software engineering

### REVIEW M2 at the end

Result in final search engine:

**cristina lopes: 58 results in 2.6 ms**

**machine learning: 2622 results in 37.0 ms**

**ACM: 2480 results in 16.3 ms**

**master of software engineering: 993 results in 146.4 ms**

20 Testing queries

**Should Be Good**:
machine learning, UCI computer science, artificial intelligence, Software Engineering Course, Database management systems, web development, graduate program, faculty research, data structures algorithms, network security

**Should Be Bad**:
computer *(too broad)*, Information *(too broad)*, machien leraning *(grammar)*, what is programming

*(question)*, CS121 project milestone *(too specific)*, best worst good bad *(meaningless)*, zxcvbnm qwerty *(meaningless)*, the and or but *(stopwords)*, easy difficult programming, nonexistent terminology xyz *(nonexistent)*

## MileStone3

### Stage 3: Link-Based Scoring – PageRank and HITS

- Constructed a **directed graph of document links** from extracted HTML.
- Calculated **PageRank** to model the global influence of a document.
- Applied the **HITS algorithm** to compute authority scores, capturing topic relevance based on link structure.

### Stage 3.5: Design Bottleneck + Flow Redesign

- Faced significant complexity in combining multiple signals.
- Struggled with ranking logic and signal weight balance.
- Sketched out a full pipeline and scoring architecture on paper. (See figures below)
- Consulted TA for advice on signal fusion and pipeline bottlenecks.

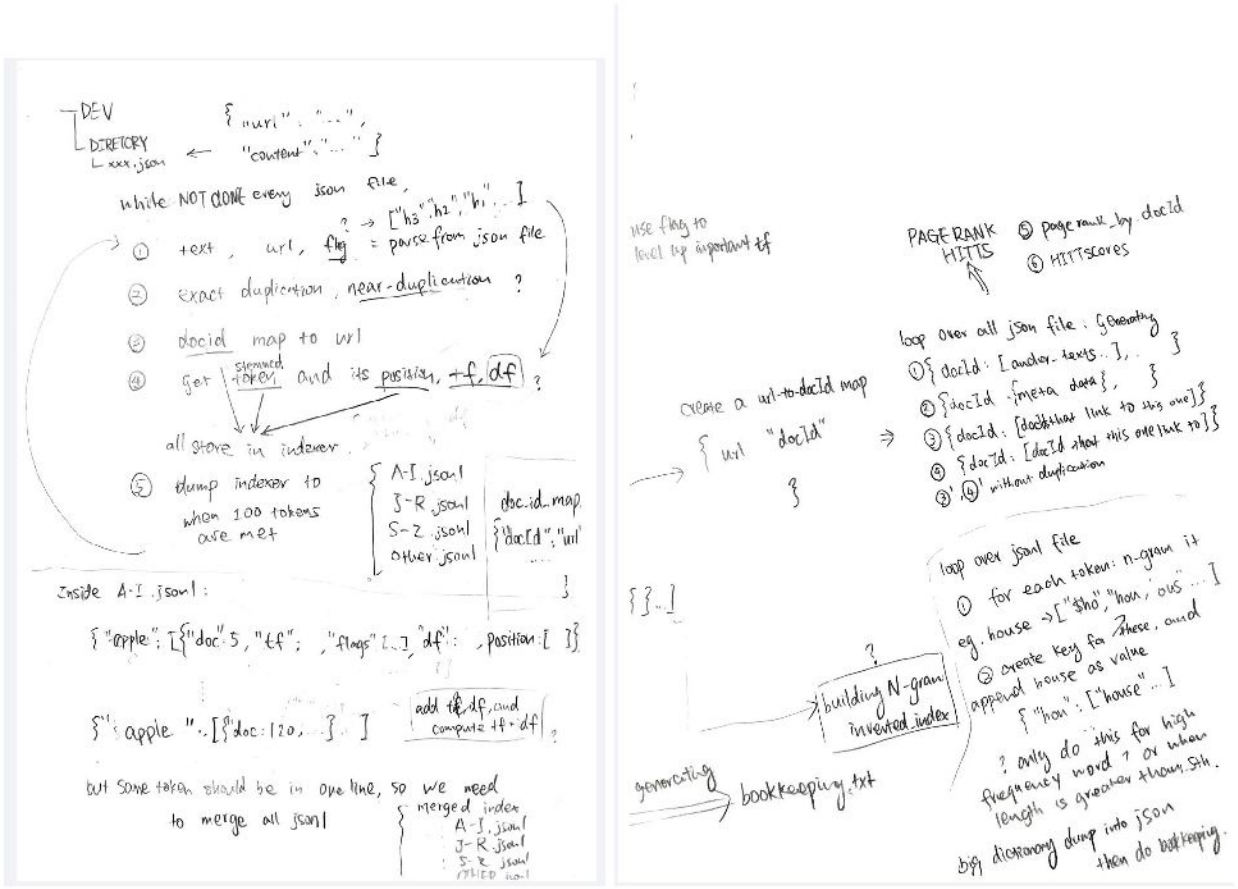



**Figure1. Logic Flow of building index**

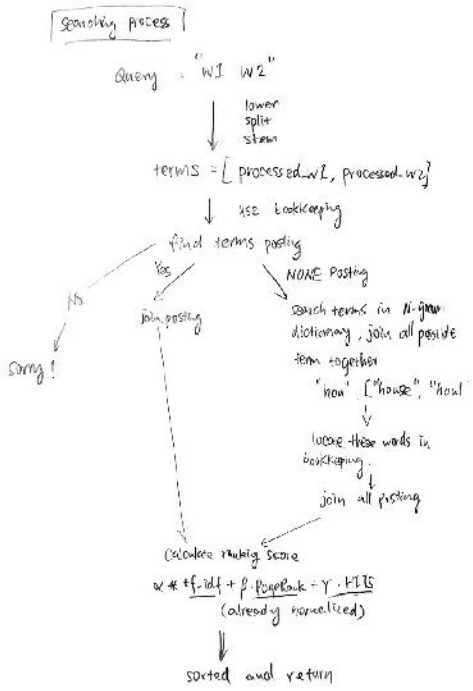



**Figure2. Logic Flow of Searching Queries**

## Stage 4: Score Normalization & Ranking Fusion

- Introduced a weighted scoring formula:
  score = α × normalized_tf-idf + β × normalized_pagerank + γ × normalized_hits
- Applied **Min-Max normalization** across all metrics to ensure fairness and avoid skew from varying scales.
- Allowed flexible tuning of alpha, beta, and gamma parameters to adjust ranking emphasis.

## Stage 5: Semantic Refinement – N-Gram Matching & Proximity Score

- Implemented **N-Gram fuzzy matching** to expand queries when exact match fails, enabling spelling tolerance and improved recall.
- Added a **proximity scoring function** based on sliding windows, giving higher ranks to documents where query terms appear closer together.
- Optimized performance by:
  - Limiting proximity scoring to top-K candidates after initial ranking
  - Skipping proximity for short queries (e.g., ≤ 2 terms)
  - Capping fuzzy match time with max_fuzzy_time_ms

Testing Phase2: All queries that tested in phase1 perform well.

## Stage 6: Title Boosting

- Added token-level **boosting flags** to prioritize terms appearing in the <title> or <meta> tags of a document.
- Adjusted TF calculation during indexing to assign extra weight (boosted_tf = freq × weight) for flagged tokens, enhancing the ranking relevance of title content.

**Stage 7: Done!**

**Stage extra for time and resources consuming, we find out that many links is out of data, whether 404 or not accessible or just a download link, not a html page. We filter it out at very beginning of the DEV iteration so we get a clean ver of DEV to run with.**

Testing Phase1 Detailed Statistics

| Good Query | Key Takeaways |
|---|---|
| **machine learning** | 4419 results in 27.4 ms; solid matches with academic pages; first result is a non-informative file list – needs filtering. |
| **UCI computer science** | 8027 results in 90.3 ms; many strong faculty-related pages, but duplicates suggest the need for deduplication. |
| **artificial intelligence** | 1096 results in 10.4 ms; good match with faculty and research pages; second result is a raw file – should be flagged. |
| **Software Engineering Course** | 3299 results in 51.8 ms; top result is a directory – low value. Others include faculty and legacy course pages. |
| **Others (DB, web dev, grad program, etc.)** | Generally fast responses (25–60 ms) with relevant academic or research content. |

| Bad Query | Problem |
|---|---|
| **computer, Information** | Too broad – high result count (18K–20K); lacks focus. |
| **machien leraning** | 0 results – spelling error. |
| **what is programming** | 6158 results, 90 ms – question format isn't ideal for document retrieval. |
| **CS121 project milestone** | 0 results – overly specific/internal term. |
| **best worst good bad** | 51 results – semantically meaningless for ranking. |
| **zxcvbnm qwerty** | 0 results – keyboard mashing. |
| **the and or but** | 8800 results in 219.5 ms – stopwords dominate. |

| | |
|---|---|
| **nonexistent terminology xyz** | 0 results – fake term. |

Testing Phase2 Detailed Statistics

| Good Query | Key Takeaways |
|---|---|
| **machine learning** | 2622 results in 41.2 ms |
| **UCI computer science** | 5305 results in 103.7 ms |
| **artificial intelligence** | 804 results in 11.8 ms |
| **Software Engineering Course** | 1977 results in 77.9 ms |
| **Database management systems** | 1997 results in 71.5 ms |
| **web development** | 2703 results in 57.4 ms |
| **graduate program** | 2903 results in 50.0 ms |
| **faculty research** | 2952 results in 43.1 ms |
| **data structures algorithms** | 1069 results in 64.7 ms |
| **network security** | 568 results in 26.6 ms |

| Bad Query | Result |
|---|---|
| **computer** | **13694 results in 61.7 ms** |
| **Information** | **15247 results in 68.3 ms** |

| | |
|---|---|
| **machien leraning** | **0 results in 1.6 ms** |
| **what is programming** | **3631 results in 100.8 ms** |
| **CS121 project milestone** | **0 results in 62.3 ms** |
| **best worst good bad** | **36 results in 37.7 ms** |
| **zxcvbnm qwerty** | **0 results in 3.0 ms** |
| **the and or but** | **5823 results in 287.0 ms** |
| **easy difficult programming** | **202 results in 45.5 ms** |
| **nonexistent terminology xyz** | **0 results in 4.5 ms** |

Testing Phase3 with CLEAN DEV Detailed Statistics

| Good Query | Key Takeaways |
|---|---|
| **machine learning** | 1021 results in 57.9 ms |
| **UCI computer science** | 882 results in 122.9 ms |
| **artificial intelligence** | 291 results in 31.0 ms |
| **Software Engineering Course** | 108 results in 105.6 ms |
| **Database management systems** | 150 results in 82.9 ms |
| **web development** | 389 results in 64.5 ms |
| **graduate program** | 133 results in 37.4 ms |
| **faculty research** | 428 results in 82.8 ms |

| | |
|---|---|
| **data structures algorithms** | 180 results in 68.4 ms |
| **network security** | 70 results in 32.8 ms |

| Bad Query | Result |
|---|---|
| **computer** | **6256 results in 73.0 ms** |
| **Information** | **6504 results in 127.0 ms** |
| **machien leraning** | **1 results in 36.8 ms** |
| **what is programming** | **340 results in 110.0 ms** |
| **CS121 project milestone** | **0 results in 22.1 ms** |
| **best worst good bad** | **4 results in 48.2 ms** |
| **zxcvbnm qwerty** | **0 results in 3.0 ms** |
| **the and or but** | **882 results in 430.7 ms** |
| **easy difficult programming** | **18 results in 22.4 ms** |
| **nonexistent terminology xyz** | **0 results in 8.2 ms** |

## Detailed Log Are Below

1) `machine learning` — 4419 results in 27.4 ms

Top 5 page:No Title
http://mondego.ics.uci.edu/datasets/maven-contents.txt

drwxrwxrwx sourcerer/sourcerer 0 2013-06-10 14:44 maven/ drwxrwxrwx sourcerer/sourcerer 0 2012-07-06 14:10 maven/wss4j/ drwxrwxrwx sourcerer/sourcerer 0 2012-07-06 14:10 maven/wss4j/wss4j/ drwxrwxrwx sourcerer/sourcerer 0 2012-07-06 14:10 maven/wss4j/wss4j/1.5.0/ -rwxrwxrwx sourcerer/sourcerer 536118 2012-07-06 14:10 maven/wss4j/wss4j/1.5.0/jar.jar...

Relevance Score: 111.1717