Bronco ID: 014644097

Last Name: Pham

First Name: Nathan

* 1. Querying in a database table is easier compared to querying text documents for two reasons: database records allow for indexing and text documents are unstructured.

Databases use indexing mechanisms to optimize query performance. These indexes enable rapid data retrieval using well-defined semantics on organized databases with specific columns and criteria, reducing the need for full scans of the data.

Text documents often lack a predefined structure making queries to specific terms take longer. Because text documents are unformatted, it is necessary to use natural language processing (NLP) techniques for tasks like entity recognition, sentiment analysis, or text classification. In contrast, databases store data in a structured format, reducing the need for complex data extraction.

* 1. Information Retrieval researcher can use text to compare multimedia documents by labeling or adding tags to these types of documents, so when a text query is looking for multimedia document, it can index and compare results of the multimedia document the text query was looking for. This scenario is changing with the popularization of machine learning techniques. Multimedia documents can now be categorized with techniques like classification and collaborative filtering.
  2. Web search engine: Searches through the entire world wide web but not the deepweb or the darkweb.
  3. Vertical search engine: Searches for a specific topic or segment of the internet like products on Amazon’s website.
  4. Enterprise search engine: Searches from sources such as databases and intranets and usually used within companies.
  5. Desktop search engine: Searches for data within the user’s computer and built into the operating system.
  6. Peer-to-peer search: Searches from computers connected via the same network. P2P searches are different compared to the other types of search engines because it is not centralized, meaning that data is stored on individual computers.
  7. Ad hoc search: searches through the data what is likely relevant to the user’s text query.
  8. Classification: searches through labeled data for documents related to the user’s query
  9. Question answering: searches through data and response with an answer to the user’s question query
  10. Filtering: searches through data and what is likely relevant to the user based on information extracted from the user (not through the user’s query).
  11. Topical relevance is important to ensure the search engine returns information that is objectively related to the user. User relevance is important because it returns information that is specialized to individual users and can be tailored to be more subjective.
  12. Query: Best Picture Movie 2022, Output: Everything Everywhere All at Once. The result objectively won Best Picture by the Oscars in 2022 and does not consider any information from the user to determine the result.
  13. Query: Restaurant, Output: Nearby Vegan Market. The search engine considered the user location data and their preferred diet to determine the most relevant result and did not return a restaurant, highlighting that there is no topical relevance.
  14. Query: Video game, Output: Pokémon Scarlet/Violet. The search engine considered the user’s past searches to help determine the result. The output is a single example of the topic of the query, showing topical relevance.

1. S

|  |  |  |
| --- | --- | --- |
|  | relevant | irrelevant |
| retrieve | 2 | 1 |
| not retrieved | 1 | 1 |

precision: 2/3 = 67%, recall: 2/3 = 67%

|  |  |  |
| --- | --- | --- |
|  | relevant | irrelevant |
| retrieve | 3 | 2 |
| not retrieved | 0 | 0 |

precision: 3/5 = 60%, recall: 3/3 = 100%

|  |  |  |
| --- | --- | --- |
|  | relevant | irrelevant |
| retrieve | 2 | 0 |
| not retrieved | 1 | 2 |

precision: 2/2 = 100%, recall: 2/3 = 67%

|  |  |  |
| --- | --- | --- |
|  | relevant | irrelevant |
| retrieve | 0 | 2 |
| not retrieved | 3 | 0 |

precision: 0/2 = 0%, recall: 0/3 = 0%

1. The web search engine first begins with webpages being crawled and parsed. Web crawlers extract links from the HTML content from the webpages and maintain a list of URLs to be visited. Once the web crawler finishes fetching the entire web page, the parser begins parsing the HTML markup of the page, extracting selected elements like tags, attributes, or images. More importantly, the parser handles text acquisition on the web pages, and these texts will be used for index creation and stored on a document data storage for later usage. Index creation is the process of converting texts into terms, measuring these terms’ frequency, creating weights for each term to determine their relative importance, and matching each term to their respective webpages. Indexes are useful because they allow for quick retrieval of webpages though the use of terms.

The user interacts with the web search engine by entering a query. The query is edited to remove all stopping words and stems words into their base form. The edited query then references the indexes and returned is a list of webpages ranked from most relevant to least relevant using a function called Term Frequency - Inverse Document Frequency (TF-IDF). The user sees the ranked webpages as well as snippets of the webpages from the document data storage.

Finally, the web search engine needs to evaluate itself. When the user clicks on one or more of the ranked webpages, the web search engine records this user interaction and logs the data to measure its performance. Two performance metrics call precision and recall can be used to measure how successful the web search engine is able to return relevant webpages. Precision indicates the proportion of retrieved webpages that are relevant by calculating the number of relevant and retrieved webpages divided by the sum of relevant and retrieved webpages plus the irrelevant and retrieved webpages. Recall is the proportion of relevant webpages that are retrieved by calculating the number of relevant and retrieved webpages divided by the sum of relevant and retrieved webpages plus the relevant and not retrieved webpages.

d1 = “I love cats and cats”.

d2= “She loves her dog”.

d3= “They love their dogs and cat”.

Stopwords: pronouns, conjunctions; Stemming: remove *s*

d1 = “love cat cat”

d2 = “love dog”

d3 = “love dog cat”

|  |  |  |  |
| --- | --- | --- | --- |
|  | love | cat | dog |
| d1 | 0 | .117 | 0 |
| d2 | 0 | 0 | .088 |
| d3 | 0 | .59 | .59 |

1. <https://github.com/nathanvpham/CS4250Assignment1>