Search Engines--Information Retrieval in Practice

tanshuqiu

Email: tsq@cqut.edu.cn

1

Architecture of a Search Engine

First : What Is an Architecture

Second : Basic Building Blocks

Third : Breaking It Down

What Is an Architecture

Our search engine architecture is used to present high-level descriptions of the important components of the system and the relationships between them.

The architecture of a search engine is determined by these two requirements: effectiveness and efficiency.

What Is an Architecture

Effectiveness (quality): We want to be able to retrieve the most relevant set of documents possible for a query.

Efficiency (speed): We want to process queries from users as quickly as possible.

Basic Building Blocks

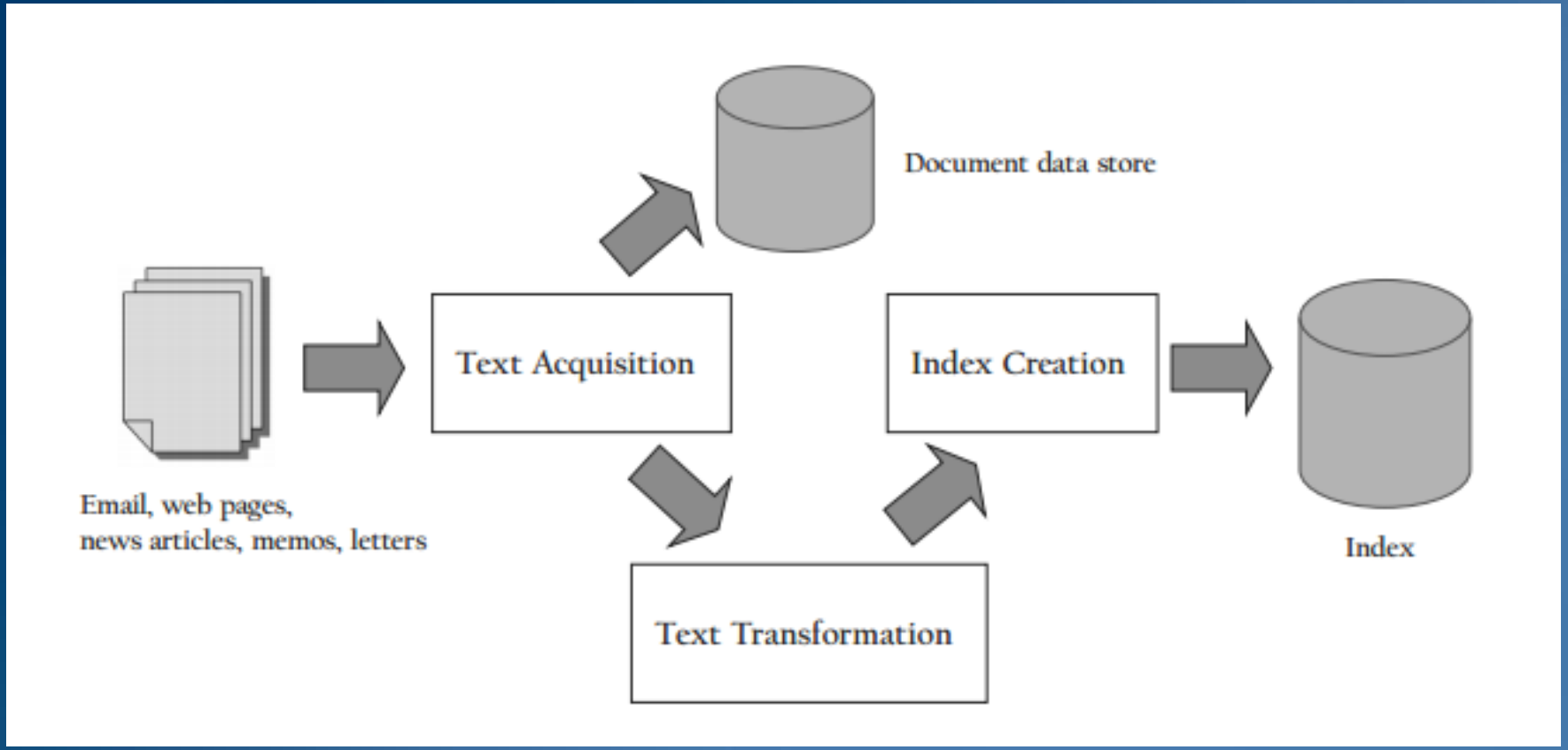
Search engine components support two major functions, which we call the **indexing process** and the **query process**.

The indexing process builds the structures that enable searching, and the query process uses those structures and a person’ s query to produce a ranked list of documents.

Basic Building Blocks

These major components of indexing process are text acquisition, text transformation, and index creation.

Basic Building Blocks



The indexing process

Basic Building Blocks

The task of the text acquisition component is to identify and make available the documents that will be searched.

In addition to passing documents to the next component in the indexing process, the text acquisition component creates a document data store, which contains the text and metadata for all the documents. Metadata is information about a document that is not part of the text content, such the document type, document structure, and other features, such as document length.

Basic Building Blocks

The text transformation component transforms documents into **index terms** or **features**.

Index terms, as the name implies, are the parts of a document that are stored in the index and used in searching.

A "feature" is more often used in the field of machine learning to refer to a part of a text document that is used to represent its content, which also describes an index term.

Basic Building Blocks

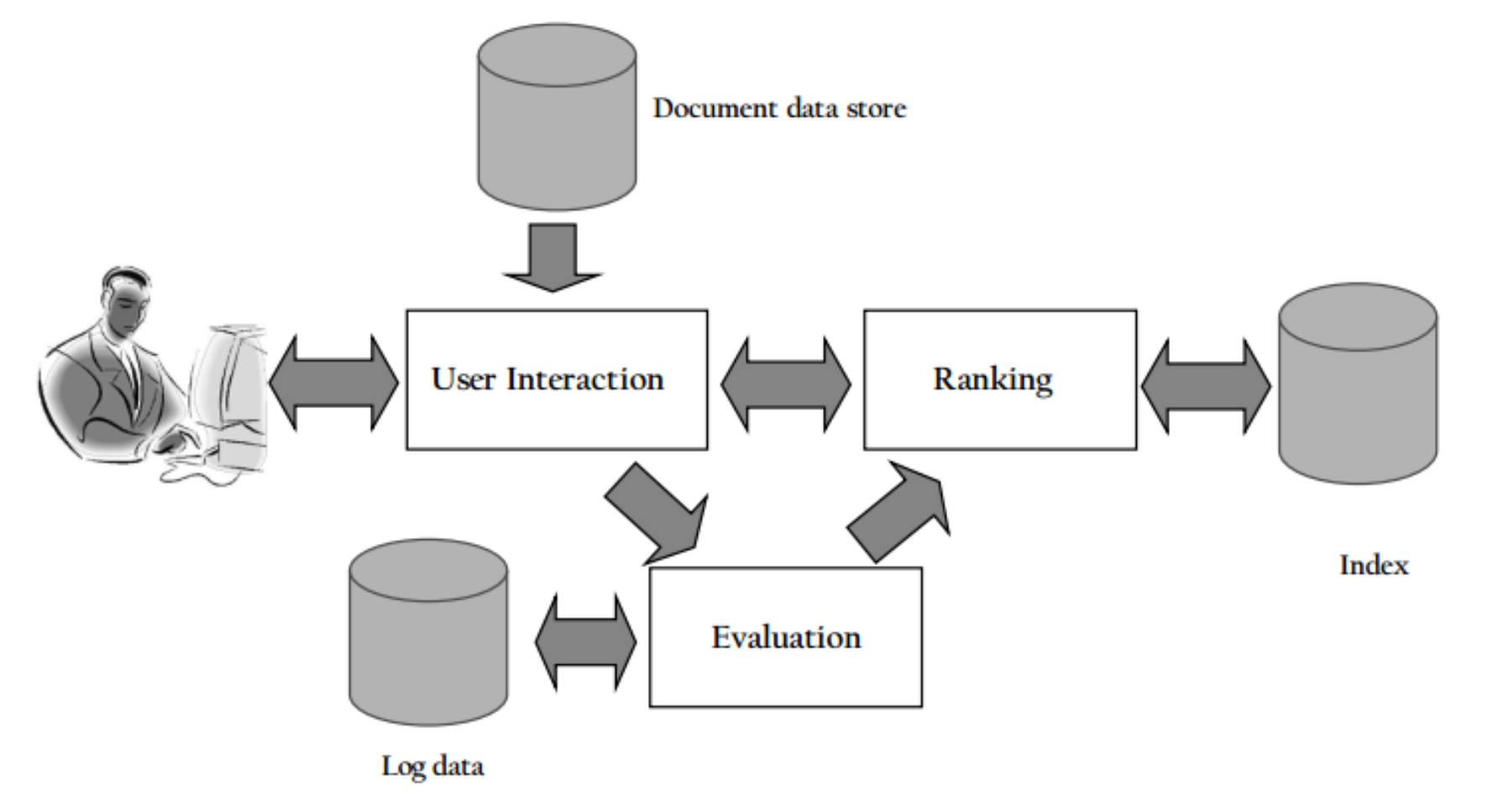
The index creation component takes the output of the text transformation component and creates the indexes or data structures that enable fast searching.

Indexes must also be able to be efficiently updated when new documents are acquired. **Inverted indexes**, or sometimes inverted files, are by far the most common form of index used by search engines.

Basic Building Blocks

These major components of query process are user interaction, ranking, and evaluation.

Basic Building Blocks



The query process

Basic Building Blocks

The user interaction component provides the interface between the person doing the searching and the search engine.

One task for this component is accepting the user’s query and transforming it into index terms. Another task is to take the ranked list of documents from the search engine and organize it into the results shown to the user.

Basic Building Blocks

The ranking component is the core of the search engine. It takes the transformed query from the user interaction component and generates a ranked list of documents using scores based on a retrieval model.

Ranking must be both efficient and effective.

The efficiency of ranking depends on the indexes, and the effectiveness depends on the retrieval model.

Basic Building Blocks

The task of the evaluation component is to measure and monitor effectiveness and efficiency. An important part of that is to record and analyze user behavior using log data. The results of evaluation are used to tune and improve the ranking component.

Breaking It Down

Contents

- ✓Text Acquisition
- ✓Text Transformation
- ✓Index Creation
- ✓User Interaction
- ✓Ranking
- ✓Evaluation

Text Acquisition

Crawler: There are a number of different types of crawlers, but the most common is the general web crawler.

A web crawler is designed to follow the links on web pages to discover and download new pages.

For enterprise search, the crawler is adapted to discover and update all documents and web pages related to a company’ s operation.

Text Acquisition

Feeds: Document feeds are a mechanism for accessing a real-time stream of documents. For example, a news feed is a constant stream of news stories and updates. In contrast to a crawler, which must discover new documents, a search engine acquires new documents from a feed simply by monitoring it.

Text Acquisition

Conversion: The documents found by a crawler or provided by a feed are rarely in plain text. Instead, they come in a variety of formats, such as HTML, XML, Adobe PDF, Microsoft Word, Microsoft PowerPoint, and so on. Most search engines require that these documents be converted into a consistent text plus metadata format.

Text Acquisition

Document data store: The document data store is a database used to manage large numbers of documents and the structured data that is associated with them. The document contents are typically stored in compressed form for efficiency. The structured data consists of document metadata and other information extracted from the documents.

Text Transformation

Parser: The parsing component is responsible for processing the sequence of text tokens in the document to recognize structural elements such as titles, figures, links, and headings.

Tokenizing the text is an important frst step in this process.

Text Transformation

Stopping: The stopping component has the simple task of removing common words from the stream of tokens that become index terms. The most common words are typically function words that help form sentence structure but contribute little on their own to the description of the topics covered by the text. Examples are “the”, “of”, “to”, and “for”.

Text Transformation

Stemming: The task of the stemming component (or stemmer) is to group words that are derived from a common stem. Grouping “fish”, “fishes”, and “fishing” is one example. By replacing each member of a group with one designated word (for example, the shortest, which in this case is “fish”), we increase the likelihood that words used in queries and documents will match.

Text Transformation

Link extraction and analysis: Links and the corresponding anchor text in web pages can readily be identified and extracted during document parsing. Extraction means that this information is recorded in the document data store, and can be indexed separately from the general text content.

Text Transformation

Information extraction: Information extraction is used to identify index terms that are more complex than single words. Tis may be as simple as words in bold or words in headings, but in general may require significant additional computation.

Index Creation

Document statistics: The task of the document statistics component is simply to gather and record statistical information about words, features, and documents.

Index Creation

Weighting: Index term weights reflect the relative importance of words in documents, and are used in computing scores for ranking.

One of the most common types used in older retrieval models is known as tf.idf weighting. There are many variations of these weights, but they are all based on a combination of the frequency or count of index term occurrences in a document and the frequency of index term occurrence over the entire collection of documents.

Index Creation

Inversion: The inversion component is the core of the indexing process. Its task is to change the stream of document-term information coming from the text transformation component into term-document information for the creation of inverted indexes.

Index Creation

Index distribution: The index distribution component distributes indexes across multiple computers and potentially across multiple sites on a network. Distribution is essential for efficient performance with web search engines.

User Interaction

Query input: The query input component provides an interface and a parser for a query language. The simplest query languages, such as those used in most web search interfaces, have only a small number of operators.

User Interaction

Query transformation: The query transformation component includes a range of techniques that are designed to improve the initial query, both before and after producing a document ranking.

Spell checking and query suggestion are query transformation techniques that produce similar output. In both cases, the user is presented with alternatives to the initial query that are likely to either correct spelling errors or be more specific descriptions of their information needs.

User Interaction

Results output: The results output component is responsible for constructing the display of ranked documents coming from the ranking component.

Ranking

Scoring: The scoring component, also called query processing, calculates scores for documents using the ranking algorithm, which is based on a retrieval model.

Ranking

Performance optimization: Performance optimization involves the design of ranking algorithms and the associated indexes to decrease response time and increase query throughput.

Ranking

Distribution: Given some form of index distribution, ranking can also be distributed. A query broker decides how to allocate queries to processors in a network and is responsible for assembling the final ranked list for the query.

Evaluation

Logging: Logs of the users' queries and their interactions with the search engine are one of the most valuable sources of information for tuning and improving search effectiveness and efficiency. Query logs can be used for spell checking, query suggestions, query caching, and other tasks.

Evaluation

Ranking analysis: Given either log data or explicit relevance judgments for a large number of (query, document) pairs, the effectiveness of a ranking algorithm can be measured and compared to alternatives. This is a critical part of improving a search engine and selecting values for parameters that are appropriate for the application.

Evaluation

Performance analysis: The performance analysis component involves monitoring and improving overall system performance, in the same way that the ranking analysis component monitors effectiveness. A variety of performance measures are used, such as response time and throughput, but the measures used also depend on the application.