Elasticsearch Training Session 2

Rajan Manickavasagam

Agenda

- Introduction to Lucene
- Features
- Key Concepts
- Vector Space Model
- TF-IDF
- Algorithm
- Elasticsearch & Lucene

Introduction to Lucene

- Doug Cutting created Lucene in 2000. He is also founder of Nutch and Hadoop.
- It is open source and licensed under Apache 2 license.
- The core is a Java search library packaged in a single Jar file.
- It has been translated into many other languages like C#, Python etc.

Features

- Minimum RAM requirements - ~ 1 MB heap.
- Supports batch and incremental indexing.
- Index size is ~ 20-30% of the data indexed.
- Provides various types of search queries.
- Provides sorting, faceting, joins, grouping & suggesters.
- Configurable storage engine.

Key Concepts

- Lucene implements an ACID transactional model with respect to indexing
  - Atomic: All changes by a writer while indexing are committed or none (using file based lock).
  - Consistency: An index will be consistent.
  - Isolation: When writing to an index, until a commit is done, changes cannot be read.
  - Durability: If an unhandled exception occurs, index will remain consistent until latest commit.

Key Concepts

- Index is made of multiple segments.
- Operations like Add, Update, Delete are buffered in memory before being flushed to disk.
- “Deleted” data is held in index (marked as deleted) until the index is “Optimized”.

Key Concepts

- Index:
  - Stores the data. It is file or RAM based.
  - It is divided into write once, read many segments.
  - Index is written in an append only mode. Once a segment is written, it is not updated. Deleted documents are held in a separate file.
  - Process of merging segments is called optimize/segment merge.
- Document: Main data carrier for indexing and search.
- Field:
  - Part of the document, that contains a name and value.
  - Multiple fields make up a document.
- Term:
  - Unit of search. Can be made up of a word or a phrase.
- Token:
  - Occurrence of a term in a field.
  - It consists of the text, start and offset.

Key Concepts

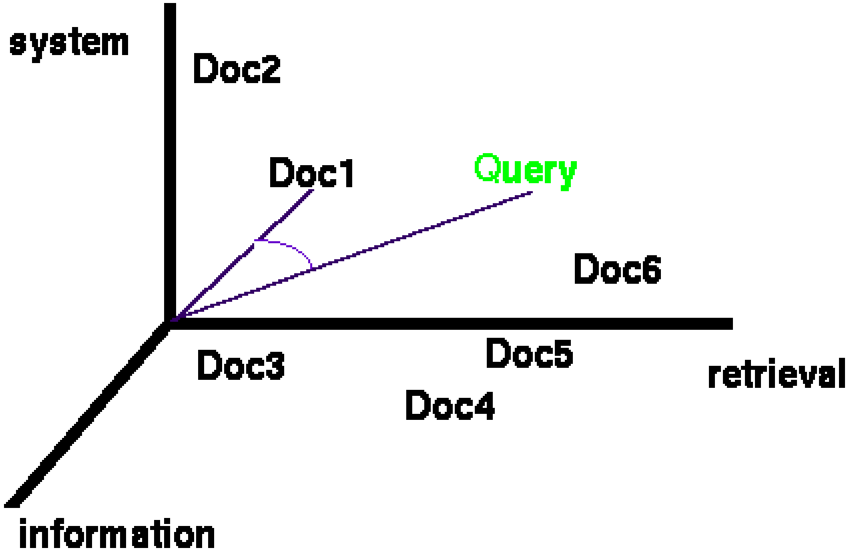
- Analyzer:
  - Consists of a tokenizer and 0 or more filters.
  - Tokenizer divides the text into tokens.
  - Filters include – lowercase, ASCII folding, Synonyms, Languages etc.
- Querying:
  - Some queries are analyzed and others not.
  - Querying should match index analyzers.

Key Concepts

- Querying:
  - Lucene syntax provides AND, OR, NOT.
  - Default is OR.
  - Query all or specific fields.
- Term modifiers:
  - ?, * for wildcards.
  - Fuzzy search with ~.
  - Boost value with ^.
  - Range searches.
  - Escape special characters with \.

Vector Space Model

- Documents and queries are modeled on a vector space model
- Closeness of query to document is calculated by cosine similarity



$$VSM = \begin{array}{c c} & \begin{array}{ccc} d_1 & d_2 & d_3 \end{array} \\ \begin{array}{c} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{array} & \begin{bmatrix} 2.4 & 0.0 & 0.0 \\ 2.1 & 2.3 & 0.0 \\ 1.2 & 0.0 & 0.0 \\ 0.0 & 2.1 & 2.5 \\ 0.0 & 0.0 & 2.4 \end{bmatrix} \end{array}$$

TF-IDF

- Term Frequency:
  - Based on the Luhn Assumption.
  - It is the frequency of a term in a document.
- Inverse Document Frequency:
  - Used for term weighting.
  - Proportion of documents that contain a term compared to all documents.

Algorithm

- Lucene Algorithm
  - Applies boolean and vector space models
  - Similarity calculated using TF/IDF weights
  - Applies sorting and filtering

Elasticsearch and Lucene

- Provided by Lucene
  - Document Analysis
  - Indexing
  - Query
  - Results
- Provided by Elasticsearch
  - Distribution
  - REST API's
  - Administration
  - Wrapper s to indexing and querying
  - Plugins

References

- Lucene - https://lucene.apache.org/