Lucene Analysis Process

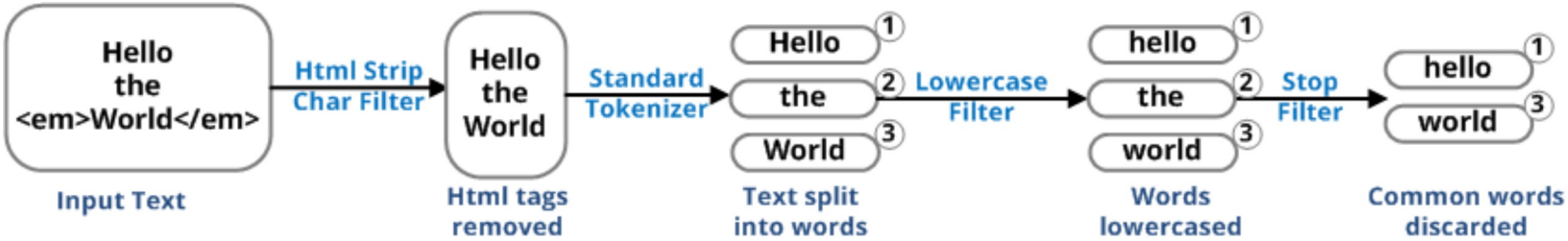
Sep 16, 2016

An analyzer executes several actions to break text and extract searchable units known as terms. These terms are the basic building blocks of an index and are used to identify the documents that match queries during search. While the previous article (../../../../blog/lucene-core-indexing-and-search-classes) focused on search, this article discusses what goes inside an analyzer.

Analyzer - Under the Hood

An analyzer usually consists of a series of tokenizer and filter classes, which may be chained into a pipeline so that output from one becomes input for the next. Tokenizers break down data into smaller chunks known as tokens while filters examine the token stream and decide what to keep, transform and discard.

An example of an analysis pipeline



Examples of operations performed by an analyzer include:

- Splitting text into individual words
- Removing unnecessary characters, such as punctuation
- Making text more uniform by removing accents and lowercasing
- Removing “noise” or very common words such as “a”, “the”, “an”
- More advanced operations to converts words into more fundamental forms – known as stemming and lemmatization

The output from an analyzer consists of a stream of tokens. A token carries some important data such as:

- A word from the original text
- Information about the position where that word occurred in the original text e.g. in “hello world”, “world” is the second word
- Character offset about each word e.g. in “hello world”, “world” starts at character number 7 and ends at character number 11
- The type of token, which depends on the analyzer used. E.g. “word”, “email”, “alphanum”
- Optional flags and payloads for more extensibility

These important elements from each token are saved into the index.

Tokens from Built-in Analyzers

Lucene contains several built-in analyzers which act differently on any given text and generate distinctive output. The table below summarizes the differences between Lucene’s built-in analyzers.

Lucene's built-in analyzer comparison

Analyzer	Split point	Highlights
KeywordAnalyzer	Does not split	Treats the whole text as a single token
WhitespaceAnalyzer	Whitespace	Leaves tokens as is after splitting
SimpleAnalyzer	Non-letter characters	Lowercases tokens Discards numeric characters
StopAnalyzer	Non-letter characters	Lowercases tokens Discards numeric characters Discards punctuation Discards common words (known as stop words) such as “a”, “an”, “the”
StandardAnalyzer	Sophisticated but mostly whitespace	The most advanced of the built-in analyzers Has logic to identify email addresses, urls, names etc. Lowercases tokens Discards punctuation Discards stop words

The output produced from Lucene’s built-in analyzers (application source code in github (https://github.com/r15h1/lucenedemo/tree/master/src/Lucene02Analysis)) when the text “My name is Joe. I’m 25 years old. My email address is joe.black007@gmail.com.” is provided as input is shown next. Note the differences in terms of:

- The number of tokens produced
- The position at which the text is broken
- The value inside each token and whether it was modified (e.g. lowercasing)
- The discarded characters/text
- The type of token produced

Square brackets [] are not part of the tokens and are meant for clarity only.

Tokens from KeywordAnalyzer, count = 1

Position	Offset	Value	Type
1	0-77	[My name is Joe. I'm 25 years old. My email address is joe.black007@gmail.com.]	word

Tokens from WhitespaceAnalyzer, count = 13

Position	Offset	Value	Type
1	0-2	[My]	word
2	3-7	[name]	word
3	8-10	[is]	word
4	11-15	[Joe.]	word
5	16-19	[I'm]	word
6	20-22	[25]	word
7	23-28	[years]	word
8	29-33	[old.]	word
9	34-36	[My]	word
10	37-42	[email]	word
11	43-50	[address]	word
12	51-53	[is]	word
13	54-77	[joe.black007@gmail.com.]	word

Tokens from SimpleAnalyzer, count = 16

Position	Offset	Value	Type
1	0-2	[my]	word
2	3-7	[name]	word
3	8-10	[is]	word
4	11-14	[joe]	word
5	16-17	[i]	word
6	18-19	[m]	word
7	23-28	[years]	word
8	29-32	[old]	word
9	34-36	[my]	word
10	37-42	[email]	word
11	43-50	[address]	word
12	51-53	[is]	word
13	54-57	[joe]	word
14	58-63	[black]	word
15	67-72	[gmail]	word
16	73-76	[com]	word

Tokens from StopAnalyzer, count = 14

Position	Offset	Value	Type
1	0-2	[my]	word
2	3-7	[name]	word
4	11-14	[joe]	word
5	16-17	[i]	word

Position	Offset	Value	Type
6	18-19	[m]	word
7	23-28	[years]	word
8	29-32	[old]	word
9	34-36	[my]	word
10	37-42	[email]	word
11	43-50	[address]	word
13	54-57	[joe]	word
14	58-63	[black]	word
15	67-72	[gmail]	word
16	73-76	[com]	word

Tokens from StandardAnalyzer, count = 11

Position	Offset	Value	Type
1	0-2	[my]	<ALPHANUM>
2	3-7	[name]	<ALPHANUM>
4	11-14	[joe]	<ALPHANUM>
5	16-19	[i'm]	<APOSTROPHE>
6	20-22	[25]	<ALPHANUM>
7	23-28	[years]	<ALPHANUM>
8	29-32	[old]	<ALPHANUM>
9	34-36	[my]	<ALPHANUM>
10	37-42	[email]	<ALPHANUM>
11	43-50	[address]	<ALPHANUM>
13	54-76	[joe.black007@gmail.com]	<EMAIL>