Lucene spans

Introduction

In Lucene, a span is a triple (i.e. 3-tuple) of <document number, start position, end position>. Document numbers start from zero. The positions are term positions, not character positions, and start from zero (i.e. the first token of a field has position 0, the second token has position 1, etc.). Start position is the position of the first term in the span. End position is the position of the last term in the span + 1.

For example, say document 1 has the following terms:

Term This is an Apache Lucene test Position 0 1 2 3 4 5
Then “Apache Lucene” matches a span <1 (document number), 3 (first term’s position), 5 (last term’s position + 1)>.

Using spans

Span queries

You don’t have to deal with spans directly. If you create a span query, Lucene takes care of matching the spans for you. Span query cannot be parsed by the default query parser that ships with Lucene. Instead, you create an instance of SpanQuery and give it to other classes like IndexSearcher.

Direct access

You can also using the Spans class. A Span class represents a series of spans instead of a span. It is also an iterator of spans. The easiest way of creating a Span object is to use the

SpanQuery.getSpans method, which returns a series of spans matched by a span query (in the form of a Spans object).

Application

Proximity search

Queries like “phrase A” within a distance of “phrase B” can be done using span queries. The API reference has an example of how to query for something like “‘John Kerry’ near ‘George Bush’”.

Word and phrase counts

The Spans interface makes it very easy to get the document count and occurrence count of a word or a phrase (actually, anything that a SpanQuery can represent) in a document set. It can be done by iterating through a Spans object. You can even count how many times phrase A is within a certain distance of phrase B in the document set.

Performance

A typical span query seems to take about twice as long as a typical phrase query, which in turn takes about twice as long as a term query. If you can get away with using a phrase query or even a term query, you might want to do so. For example, “term A near term B” can be done using a phrase query with a non-zero slop.

Revision #1
Created 2008-08-29 14:53:21 UTC by Chan, Wing Kai
Updated 2008-08-29 15:23:33 UTC by Chan, Wing Kai