Applying Lucene to the Web

Doug Cutting
<cutting@apache.org>

Lucene Background

- Java library for text indexing and search
- an Apache project
- not an application
- http://jakarta.apache.org/lucene/

Lucene Architecture

- Packages:
  - org.apache.lucene.**store**
  - org.apache.lucene.**document**
  - org.apache.lucene.**analysis**
  - org.apache.lucene.**index**
  - org.apache.lucene.**search**
  - org.apache.lucene.**queryParser**

Lucene Documents

- A document is a sequence of Fields.
- A field is a <*name*, *value*> pair.
  - Name is the name of the field, e.g., title, body, subject, date, etc.
  - Value is text.
- Field values may be *stored*, *indexed* or *analyzed (and, now, vectored)*.

Lucene Analysis

- An Analyzer is a TokenStream factory.
- A TokenStream is an iterator over tokens.
  - Input is a character stream.
- A Token has:
  - Text (e.g., “analyzer”).
  - Type (e.g., “word”, “sent”, “para”).
  - Start & Length offset, in characters (e.g, <5,8>)
  - Position Increment (normally 1)

Lucene Index

- Term is <field, text>
- Inverted Index
  - Term → <df, <doc, <position>* >*>
  - e.g., “body:analysis” → <2, <2, <14>>, <4, <2>>>
- *New: Term Vectors!*

Lucene Search

- Primitive queries:
  - TermQuery: match docs containing a Term
  - PhraseQuery: match docs w/ sequence of Terms
  - BooleanQuery: match docs matching other queries. e.g., +title:Lucene +body:“Doug Cutting” -key:bogus
  - *New: Spans!*
- Derived queries:
  - Prefix, Wildcard, etc.

Lucene Scoring

- Vector Space Model (more-or-less)
- score(q,d) = $\sum_t$ tf * idf * norm
- "tf" is F(frequency of t in d)
  - default is sqrt()
- "idf" is G(frequency of t in collection)
  - default is log (#terms / freq(t))
- "norm" is H(document)
  - default is 1/sqrt(#terms in d) * boost(d)

Nutch Goals

- Increase transparency of web search.
  - search is essential to internet navigation
  - yet algorithms are secret
- A free, open-source implementation should help.

Nutch Software

- Web Search Application
  - Maintain DB of pages and links
    - Pages have scores, assigned by analysis
  - Fetches high-scoring, out-of-date pages
  - Maintain index of current page set
  - Distributed search front end
  - Based on Lucene
- http://www.nutch.org/

Nutch Documents

field	*stored*	*indexed*	*analyzed*
`url`	Yes	Yes	Yes
`title`	Yes	No	
`anchor`	No	Yes	Yes
`content`	No	Yes	Yes

Nutch Analysis

- Defined w/ JavaCC
- Words are (<letter> | [0-9_&])+
  - or acronyms: <letter> [.] (<letter> [.])+
  - or CJK character
- First word in each anchor gets big position increment, to inhibit cross-anchor matches.
- No stop list or stemmer.
- URLs, email, etc. tokenized same as other text.

Nutch Queries

- By default:
  - search url, anchors and content
  - require all query terms
  - reward for proximity
- E.g., search for “search engine” is expanded to:
  - url:(+search +engine +“search engine”~*p*^*a*)^*x*
  - anchor:(+search +engine +“search engine”~*q*^*b*)^*y*
  - content:(+search +engine +“search engine”~*p*^c)^*z*

Nutch Querys

- Certain characters cause implicit phrases:
  - dash, plus, colon, slash, dot, apostrophe and atsign
  - URL & email are thus phrase searches
  - e.g., “http://www.nutch.org/”, “n-gram”, etc.
- Stop words removed, unless in phrase or required.
  - can use N-grams if in phrase

Nutch N-Grams

- Very common terms are indexed with neighbors.
  - E.g., w/ “the”, “http”, “www”, “http-www” & “org”:
  - “Buffy the Vampire” is indexed as
    buffy, buffy-the+0, the, the-vampire+0, vampire,
  - “http://www.nutch.org/” is indexed as:
    http, http-www+0, http-www-nutch+0,
    www, www-nutch+0, nutch, nutch-org+0, org.
  - terms are field specific
  - improves performance of phrase searches

Nutch N-Gram Query

- For explicit phrase query: “Buffy the Vampire”:
  - for content field, translated to:
    content:“buffy-the the-vampire”
  - much faster b/c we don't have to search for “the”
- For query: http://www.nutch.org/:
  - implicit phrase: “http www nutch org”
  - for URL field, translated to:
    url:“http-www-nutch www-nutch nutch-org”

Demos

- http://labs.yahoo.com/demo/nutch/
- http://www.mozdex.com/search.html
- http://www.objectssearch.com/en/search.html
- http://devjr.cws.oregonstate.edu:8080/
- http://www.nutch.org/