

Turning Search Upside-Down

Searching queries with documents

Alan Woodward



What are we trying to achieve?

- We want to run many queries over a stream of documents
- Originally for media monitoring, but also applies to classification, tagging, alerting, etc
- Need various information from the queries - simple yes/no matches, scores, match positions.



- Simple answer: use Lucene's MemoryIndex
 - Index each document into a single-document in-memory index
 - Run all your registered queries over the document
 - Record which queries matched and report back



Sloooooow



These are big queries

```
((;!MOBILE PHONE*"; OR ";PHONE MAST*"; OR ";HANDSET*"; OR ";CELL* PHONE*"; OR ";3G"; OR ";GPRS"; OR
";G.P.R.S"; OR ";!GENERAL !RADIO PACKET SERVICE*"; OR ";GSM"; OR ";G.S.M"; OR ";!GLOBAL SYSTEM FOR !MOBILE
COMM*"; OR ";HSDPA"; OR ";H.S.D.P.A"; OR ";HIGH SPEED DOWNLINK !PACKET ACCESS"; OR ";HSUPA"; OR
";H.S.U.P.A"; OR ";HIGH SPEED !UPLINK !PACKET ACCESS"; OR ";UMTS"; OR ";U.M.T.S"; OR ";MVNO"; OR ";M.V.N.O";
OR ";SMS"; OR ";SHORT MESSAGE !SERVICE*"; OR ";MMS"; OR ";!MULTIMEDIA MESSAGE !SERVICE*"; OR ";!MOBILES"; OR
";!CELLPHONE*"; OR ";!TELECOM*"; OR ";!LANDLINE*"; OR ";!TELEPHONE*"; OR ";PHONE*"; OR ";!TELEKOM*"; OR
";TELCO*"; OR ";VODAFONE"; OR ";T-MOBILE"; OR ";TMOBILE"; OR ";!TELEFONICA"; OR ";BT"; OR ";!MOBILE USER*";
OR ";TEXT MESSAG*"; OR ";SMARTPHONE*"; OR ";!VIRGIN !MEDIA*"; OR ";CABLE & !WIRELESS"; OR ";CABLE AND !
WIRELESS";) W/48 ((";PROFIT*"; OR ";LOSS*"; OR ";BAN"; OR ";BANNED"; OR ";PREMIUM RATE*"; OR ";FINANC*"; OR
";!REFINANC*"; OR ";OFFICE OF FAIR TRADING"; OR ";MERGER*"; OR ";!ACQUISIT*"; OR ";ACQUIR*"; OR
";TAKEOVER*"; OR ";BUYOUT*"; OR ";BUY-OUT*"; OR ";NEW PRODUCT*"; OR ";INVEST*"; OR ";SHARES"; OR ";MARKET*";
OR ";ACCOUNT*"; OR ";MONEY"; OR ";CASH*"; OR ";SECURIT*"; OR ";!ENTERPRIS*"; OR ";!BUSINESS*"; OR ";PRICE*";
OR ";JOINT*"; OR ";NEW VENTURE*"; OR ";PRICING"; OR ";COST*"; OR ";CHAIRM?N"; OR ";APPOINT*"; OR ";!
EXECUTIVE"; OR ";SALE*"; OR ";SELL*"; OR ";FULL YEAR"; OR ";REGULAT*"; OR ";!DIRECTIVE*"; OR ";LAW"; OR
";LAWS"; OR ";!LEGISLAT*"; OR ";GREEN PAPER"; OR ";WHITE PAPER*"; OR ";!MEDIAWATCH"; OR ";MORAL*"; OR
";ETHIC*"; OR ";ADVERT*"; OR ";AD"; OR ";ADS"; OR ";MARKETING"; OR ";!COMPLAIN*"; OR ";MIS-SOLD"; OR ";MIS-
SELL*"; OR ";SPONSOR"; OR ";COSTCUT*"; OR ";COST CUT*"; OR ";CUT* COST*"; OR ";FIBRE OPTIC*"; OR ";TAX"; OR
";TAXES"; OR ";TAXED"; OR ";EXPAND*"; OR ";!EXPANSION"; OR ";EMPLOY*"; OR ";STAFF"; OR ";WORKER*"; OR
";SPOKESM?N"; OR ";DEBUT"; OR ";BRAND*"; OR ";DIRECTOR*";) OR ((";FAIR"; OR ";UNFAIR"; OR ";%UNSCRUPULOUS";
OR ";NOT FAIR"; OR ";UNJUST*"; OR ";!PENALISE*";) W/12 (";CHARG*"; OR ";TARIFF*"; OR ";PRICE PLAN*"; OR
";GLOBAL";)) AND NOT (";EXPRESS OFFER"; OR ";TIMES OFFER"; OR ";READER OFFER"; OR ((";CALLS COST";) W/6
(;"FROM A LANDLINE"; OR ";FROM LANDLINE*"; OR ";BT LANDLINE*";))
```



Really big

```
((";!MOBILE PHONE*"; OR ";PHONE MAST*"; OR ";HANDSET*"; OR ";CELL* PHONE*"; OR ";3G"; OR ";GPRS"; OR ";G.P.R.S"; OR ";!  
GENERAL !RADIO PACKET SERVICE*"; OR ";GSM"; OR ";G.S.M"; OR ";!GLOBAL SYSTEM FOR !MOBILE COMM*"; OR ";HSDPA"; OR ";H.S.D.P.A";  
OR ";HIGH SPEED DOWNLINK !PACKET ACCESS"; OR ";HSUPA"; OR ";H.S.U.P.A"; OR ";HIGH SPEED !UPLINK !PACKET ACCESS"; OR ";UMTS"; OR  
";U.M.T.S"; OR ";MVNO"; OR ";M.V.N.O"; OR ";SMS"; OR ";SHORT MESSAGE !SERVICE*"; OR ";MMS"; OR ";!MULTIMEDIA MESSAGE !  
SERVICE*"; OR ";!MOBILES"; OR ";!CELLPHONE*"; OR ";!TELECOM*"; OR ";!LANDLINE*"; OR ";!TELEPHONE*"; OR ";PHONE*"; OR ";!  
TELEKOM*"; OR ";TELCO*"; OR ";VODAFONE"; OR ";T-MOBILE"; OR ";TMOBILE"; OR ";!TELEFONICA"; OR ";BT"; OR ";!MOBILE USER*"; OR  
";TEXT MESSAG*"; OR ";SMARTPHONE*"; OR ";!VIRGIN !MEDIA*"; OR ";CABLE & !WIRELESS"; OR ";CABLE AND !WIRELESS";) W/48  
((";PROFIT*"; OR ";LOSS*"; OR ";BAN"; OR ";BANNED"; OR ";PREMIUM RATE*"; OR ";FINANC*"; OR  
";!REFINANC*"; OR ";OFFICE OF FAIR TRADING"; OR ";MERGER*"; OR ";!ACQUISIT*"; OR ";ACQUIR*"; OR ";TAKEOVER*"; OR ";BUYOUT*"; OR  
";BUY-OUT*"; OR ";NEW PRODUCT*"; OR ";INVEST*"; OR ";SHARES"; OR ";MARKET*"; OR ";ACCOUNT*"; OR ";MONEY"; OR ";CASH*"; OR  
";SECURIT*"; OR ";!ENTERPRIS*"; OR ";!BUSINESS*"; OR ";PRICE*"; OR ";JOINT*"; OR ";NEW VENTURE*"; OR ";PRICING"; OR ";COST*";  
OR ";CHAIRM?N"; OR ";APPOINT*"; OR ";!EXECUTIVE"; OR ";SALE*"; OR ";SELL*"; OR ";FULL YEAR"; OR ";REGULAT*"; OR ";!DIRECTIVE*";  
OR ";LAW"; OR ";LAWS"; OR ";!LEGISLAT*"; OR ";GREEN PAPER"; OR ";WHITE PAPER*"; OR ";!MEDIAWATCH"; OR ";MORAL*"; OR ";ETHIC*";  
OR ";ADVERT*"; OR ";AD"; OR ";ADS"; OR ";MARKETING"; OR ";!COMPLAIN*"; OR ";MIS-SOLD"; OR ";MIS-SELL*"; OR ";SPONSOR"; OR  
";COSTCUT*"; OR ";COST CUT*"; OR ";CUT* COST*"; OR ";FIBRE OPTIC*"; OR ";TAX"; OR ";TAXES"; OR ";TAXED"; OR ";EXPAND*"; OR ";!  
EXPANSION"; OR ";EMPLOY*"; OR ";STAFF"; OR ";WORKER*"; OR ";SPOKESM?N"; OR ";DEBUT"; OR ";BRAND*"; OR ";DIRECTOR*");) OR  
((";FAIR"; OR ";UNFAIR"; OR ";%UNSCRUPULOUS"; OR ";NOT FAIR"; OR ";UNJUST*"; OR ";!PENALISE*");) W/12 (";CHARG*"; OR ";TARIFF*";  
OR ";PRICE PLAN*"; OR ";GLOBAL";)) AND NOT (";EXPRESS OFFER"; OR ";TIMES OFFER"; OR ";READER OFFER"; OR (";CALLS COST";) W/6  
(";FROM A LANDLINE"; OR ";FROM LANDLINE*"; OR ";BT LANDLINE*");)) OR (";!MOBILE PHONE*"; OR ";PHONE MAST*"; OR ";HANDSET*"; OR  
";CELL* PHONE*"; OR ";3G"; OR ";GPRS"; OR ";G.P.R.S"; OR ";!GENERAL !RADIO PACKET SERVICE*"; OR ";GSM"; OR ";G.S.M"; OR ";!  
GLOBAL SYSTEM FOR !MOBILE COMM*"; OR ";HSDPA"; OR ";H.S.D.P.A"; OR ";HIGH SPEED DOWNLINK !PACKET ACCESS"; OR ";HSUPA"; OR  
";H.S.U.P.A"; OR ";HIGH SPEED !UPLINK !PACKET ACCESS"; OR ";UMTS"; OR ";U.M.T.S"; OR ";MVNO"; OR ";M.V.N.O"; OR ";SMS"; OR  
";SHORT MESSAGE !SERVICE*"; OR ";MMS"; OR ";!MULTIMEDIA MESSAGE !SERVICE*"; OR ";!MOBILES"; OR ";!CELLPHONE*"; OR ";!TELECOM*";  
OR ";!LANDLINE*"; OR ";!TELEPHONE*"; OR ";PHONE*"; OR ";!TELEKOM*"; OR ";TELCO*"; OR ";VODAFONE"; OR ";T-MOBILE"; OR  
";TMOBILE"; OR ";!TELEFONICA"; OR ";BT"; OR ";!MOBILE USER*"; OR ";TEXT MESSAG*"; OR ";SMARTPHONE*"; OR ";!VIRGIN !MEDIA*"; OR  
";CABLE & !WIRELESS"; OR ";CABLE AND !WIRELESS";) W/48 (";PROFIT*"; OR ";LOSS*"; OR ";BAN"; OR ";BANNED"; OR ";PREMIUM  
RATE*"; OR ";FINANC*"; OR ";!REFINANC*"; OR ";OFFICE OF FAIR TRADING"; OR ";MERGER*"; OR ";!ACQUISIT*"; OR ";ACQUIR*"; OR  
";TAKEOVER*"; OR ";BUYOUT*"; OR ";BUY-OUT*"; OR ";NEW PRODUCT*"; OR ";INVEST*"; OR ";SHARES"; OR ";MARKET*"; OR ";ACCOUNT*"; OR  
";MONEY"; OR ";CASH*"; OR ";SECURIT*"; OR ";!ENTERPRIS*"; OR ";!BUSINESS*"; OR ";PRICE*"; OR ";JOINT*"; OR ";NEW VENTURE*"; OR  
";PRICING"; OR ";COST*"; OR ";CHAIRM?N"; OR ";APPOINT*"; OR ";!EXECUTIVE"; OR ";SALE*"; OR ";SELL*"; OR ";FULL YEAR"; OR  
";REGULAT*"; OR ";!DIRECTIVE*"; OR ";LAW"; OR ";LAWS"; OR ";!LEGISLAT*"; OR ";GREEN PAPER"; OR ";WHITE PAPER*"; OR ";!  
MEDIAWATCH"; OR ";MORAL*"; OR ";ETHIC*"; OR ";ADVERT*"; OR ";AD"; OR ";ADS"; OR ";MARKETING"; OR ";!COMPLAIN*"; OR ";MIS-SOLD";  
OR ";MIS-SELL*"; OR ";SPONSOR"; OR ";COSTCUT*"; OR ";COST CUT*"; OR ";CUT* COST*"; OR ";FIBRE OPTIC*"; OR ";TAX"; OR ";TAXES";  
OR ";TAXED"; OR ";EXPAND*"; OR ";!EXPANSION"; OR ";EMPLOY*"; OR ";STAFF"; OR ";WORKER*"; OR ";SPOKESM?N"; OR ";DEBUT"; OR  
";BRAND*"; OR ";DIRECTOR*");) OR (";FAIR"; OR ";UNFAIR"; OR ";%UNSCRUPULOUS"; OR ";NOT FAIR"; OR ";UNJUST*"; OR ";!PENALISE*");)  
W/12 (";CHARG*"; OR ";TARIFF*"; OR ";PRICE PLAN*"; OR ";GLOBAL";)) AND NOT (";EXPRESS OFFER"; OR ";TIMES OFFER"; OR ";READER  
OFFER"; OR (";CALLS COST";) W/6 (";FROM A LANDLINE"; OR ";FROM LANDLINE*"; OR ";BT LANDLINE*");))
```



And they're ugly

- Represent*
- Represent* AND represent*
- Represent* AND represent* AND *present*
- Represent* AND represent* AND *present* AND Represent AND Represent* AND *presentation AND ...



- Typical document passing through a system with ~20,000 queries matches on 1 or 2.
- So that's ~19,999 that are taking up valuable time producing no hits
- Need a way of reducing the amount of time spent on zero-hit queries. Preferably to 0 ms.



- Preselect queries that are likely to give you a hit, and filter out queries that will definitely *not* hit.
- Search a space of queries with a document
- Index your queries, and search them with your document.
- Aim is as few false positives as possible, and zero false negatives.



Luwak

- A lucene-based Java library for fast stored queries
- Queries are registered with a Monitor, which stores them in an internal lucene index.
- Documents are passed to the monitor, which converts them into queries over its internal index, and selects which stored queries to run.
- Query-indexing and Document-to-Query conversion are handled by pluggable Presearcher implementations.
- Match reporting is handled by pluggable Matchers.

Indexing Queries

- Terms are extracted from queries using specialised `Extractor<Query>` implementations.
 - `TermQuery`: just get the term
 - Numeric queries : reduce to term queries
 - Booleans : recurse through the query tree, collecting terms



Indexing Queries

- We can selectively index boolean queries
 - MUST_NOT clauses can be ignored
 - If a query contains only MUST_NOT and SHOULD clauses, then all SHOULD clauses must be extracted.
 - If a query contains **any** MUST clauses, then we can choose just one of those, and ignore all other clauses.

Indexing Queries

- What's the best way to choose which clause to index?
 - Longer terms tend to be rarer
 - Clauses with fewer sub terms will hit more rarely
 - Certain fields are better than others
- We use TermWeightor classes to calculate a weight for each candidate clause, and the best one is selected for indexing.



Indexing Queries

- Proximity and phrase queries can be treated as conjunctions - just need a single term
- Wildcard terms can be indexed by their longest invariant substring
- Some queries we can't extract a term from, and will have to be indexed under a special ANY token.
 - Normally discarded by the TermWeightor

Querying with Documents

- Need to extract all terms in a document and convert them to a large boolean disjunction.
- Since we are already indexing the document into a MemoryIndex, we can just use a TermsEnum to get the list of terms.
- If we're trying to match wildcard queries, then we pass the terms through an ngram filter and then deduplicate.
- Terms can be compared against the queryindex's terms list to remove clauses that won't match against any query.

Numbers

- Without presearcher: Monitor with 22,000 queries takes ~5 seconds to process a document
- With presearcher: Same query set will process a document in ~0.25 seconds.
- Presearcher knocks down number of queries run from all 22,000 to average of 250.



Reporting Matches

- Matches are collected by an implementation of CandidateMatcher
- Pass a MatcherFactory<? extends CandidateMatcher> to Monitor.match()
- SimpleMatcher: did this query match or not?
- ScoringMatcher: return the scores of the matching queries.



Reporting Matches

- IntervalsMatcher: Uses minimum-interval query semantics from LUCENE-2878 to report the exact positions of all matching terms in a query.
- Also useful for debugging the presearcher - can tell you exactly which terms in a query have caused it to be selected.



Extending Luwak

- Extend the existing presearchers with QueryExtractors
- Write your own presearcher implementation
- Write your own CandidateMatcher
- Pull requests welcome!
- <http://github.com/flaxsearch/luwak>



Questions?

alan@flax.co.uk
[@romseygeek](#)

