Open Source Search

Andreas Pesenhofer

max.recall information systems GmbH
Künstlergasse 11/1 • A-1150 Wien • Austria

ICIC, October 2013

max.recall information systems

- max.recall is a software and consulting company enabling enterprises to capitalize on the hidden value in the rapidly growing amount of textual data
- Customized Solutions for
  - Intelligent data analytics
  - Vertical search
- Products and Services
  - quantalyze: quantity analytics technology
  - smart.coder: open-ended question coding tool for market researchers
- Founded 2010 and located in Vienna, Austria
- Operates worldwide with int'l customers from sectors such as IP, market research, news and media, IT services

Recall and precision

- Recall
  - Percent of relevant documents (items) returned
  - 50 good answers in system, 25 returned = 50% recall
- Precision
  - Percent of documents returned that are relevant
  - 100 returned, 25 are relevant = 25% precision

- Ideal is 100% recall and 100% precision: return all relevant documents and only those
- 100% recall is easy – return all documents, but precision is low, relevant documents can't be found
- Need adequate recall & enough precision for the task - that will vary by application (data & users)

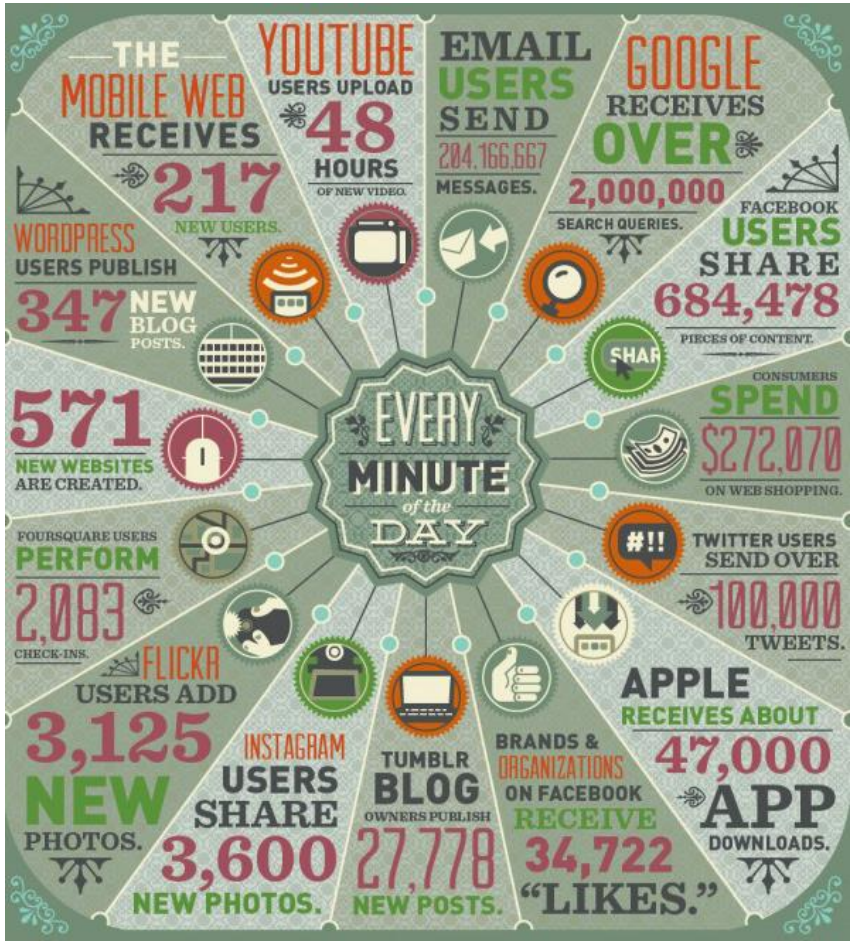
How to get good recall

- Collect, index and search all the data
  - Check for missing or corrupt data
- Index everything
  - Search everything … limit results by category AFTER the search (clustering/faceting)
- Normalize the data
  - Convert to lower case, strip/handle special characters, stemming, ...
- Use spell-checking, synonyms to match users' vocabulary with content
  - Adaptive spell-checking, application-specific synonyms
- Light (or real) natural language processing for abstract concepts

How to get good precision

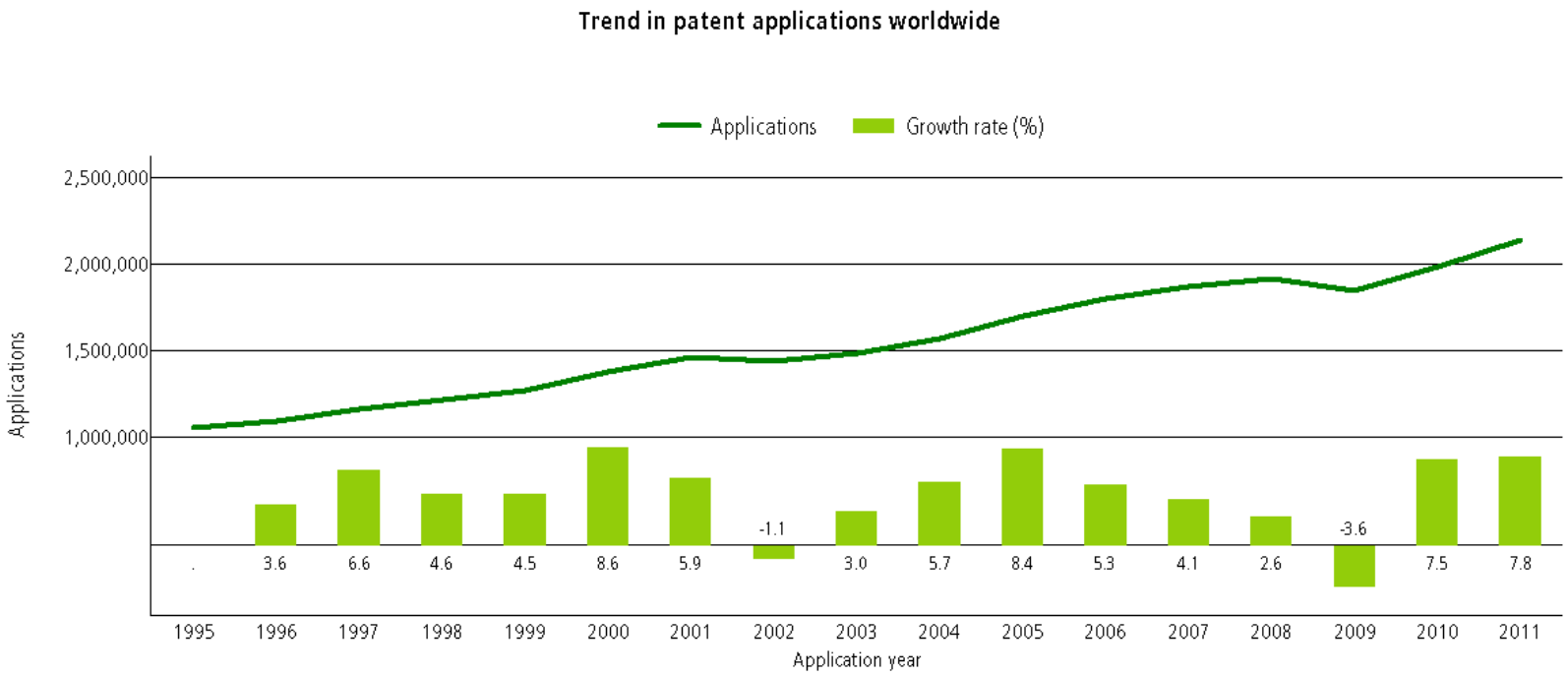
- Term frequency (TF) – more occurrences of query terms is better
- Inverse document frequency (IDF) – rarer query terms are more important
- Phrase boost – query terms near each other is better
- Field boost – where the query term is in doc matters (e.g., in 'title' better)
- Length normalization – avoid penalizing short docs
- Recency – all things being equal, recent is better
- Authority – items linked to, clicked on or bought by others may be better
- Implicit and explicit relevance feedback, more-like-this – expand query
- Clustering/faceting – intent is not specific
- Lots of data

Every minute …



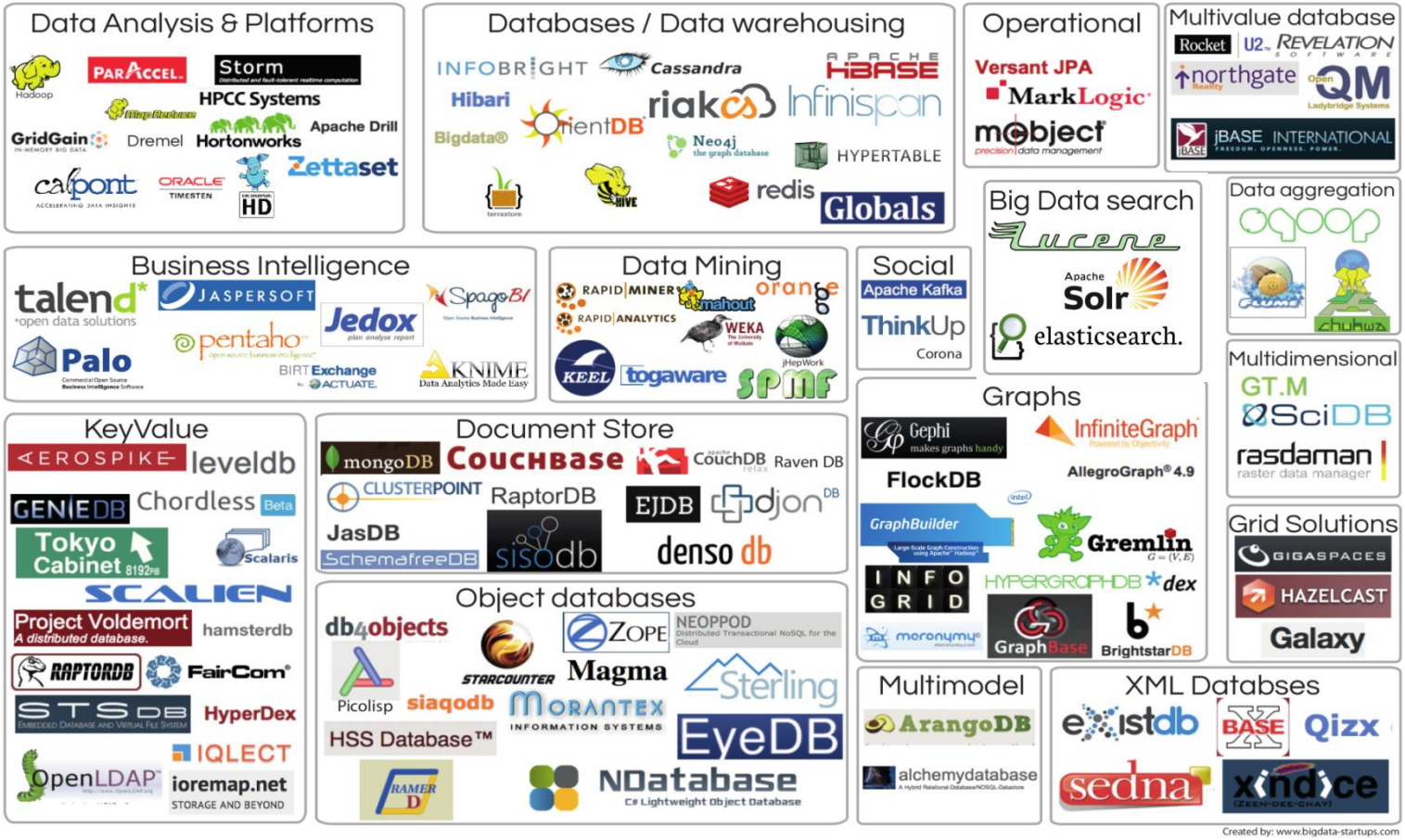
http://www.domo.com/

Groth of patent applications



Note: World totals are WIPO estimates covering around 125 patent offices (see Data Description)- These estimates include direct applications and PCT national phase entry data-
Source: WIPO Statistics Database, October 2012

Big Data Open Source Tools



Apache Lucene

Apache Lucene™ is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

- Scalable, High-Performance Indexing
  - over 150GB/hour on modern hardware
  - small RAM requirements - only 1MB heap
  - incremental indexing as fast as batch indexing
  - index size roughly 20-30% the size of text indexed

Apache Lucene (2)

- Powerful, Accurate and Efficient Search Algorithms
  - ranked searching - best results returned first
  - many powerful query types: phrase queries, wildcard queries, proximity queries, range queries and more
  - fielded searching (e.g. title, author, contents)
  - sorting by any field
  - multiple-index searching with merged results
  - allows simultaneous update and searching
  - flexible faceting, highlighting, joins and result grouping
  - fast, memory-efficient and typo-tolerant suggesters
  - pluggable ranking models, including the Vector Space Model and Okapi BM25
  - configurable storage engine (codecs)
- Cross-Platform Solution

Apache Lucene (3)

- Cross-Platform Solution
  - Available as Open Source software under the Apache License - Lucene in both commercial and Open Source programs
  - 100%-pure Java
  - Implementations in other programming languages available, the index is compatible

- Apache Lucene 4.5.0 was released on October 5th, 2013.

Apache SOLR

- Apache SOLR is an open source enterprise search platform from the Apache Lucene project.
- major features:
  - full-text search
  - hit highlighting
  - faceted search
  - dynamic clustering
  - database integration
  - handling of rich documents (e.g., Word, PDF)
  - providing distributed search and index replication, Solr is highly scalable.
- Apache SOLR 4.5.0 was released on October 5th, 2013.

elasticsearch

- elasticsearch is a distributed, RESTful, open source search server based on Apache Lucene. It is developed by Shay Banon and is released under the terms of the Apache License.
- major features:
  - fully supports the near real-time search of Apache Lucene
  - cluster setup needs no additional software
  - features of Lucene are made available through the JSON and Java API
  - JSON in / JSON out (and YAML)

- elasticsearch 0.90.5 was released on September 17th, 2013, based on Lucene 4.4.

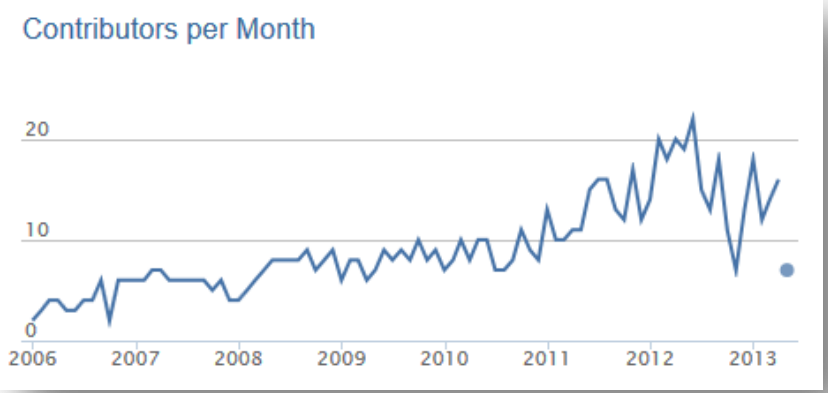
All Time Top Committers

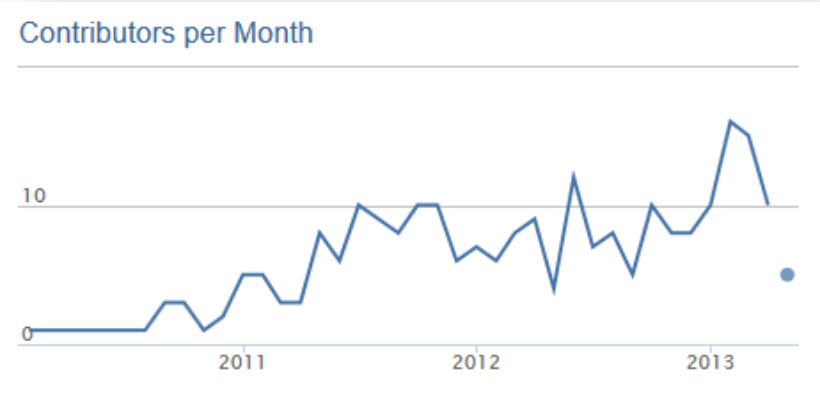
Name	12 Month Commits	All Time Commits	5 Year Trend
yonik	135	1447	5 year commit count
markrmiller (Manager)	540	890	5 year commit count
rmuir	157	599	5 year commit count

Apache Solr

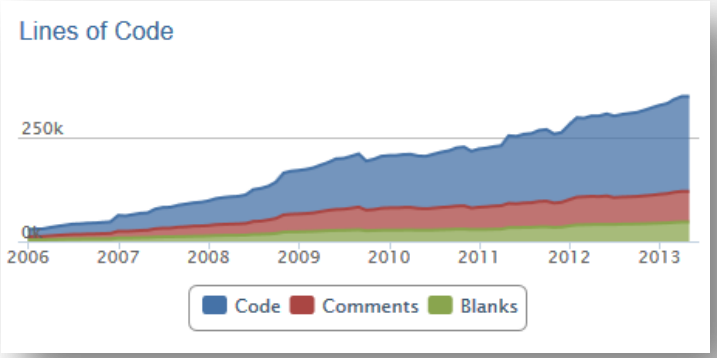
Name	12 Month Commits	All Time Commits	5 Year Trend
kimchy (founder, developer)	0	2478	5 year commit count
Shay Banon	642	1504	5 year commit count
Martijn van Groningen	165	165	5 year commit count
Igor Motov	119	152	5 year commit count

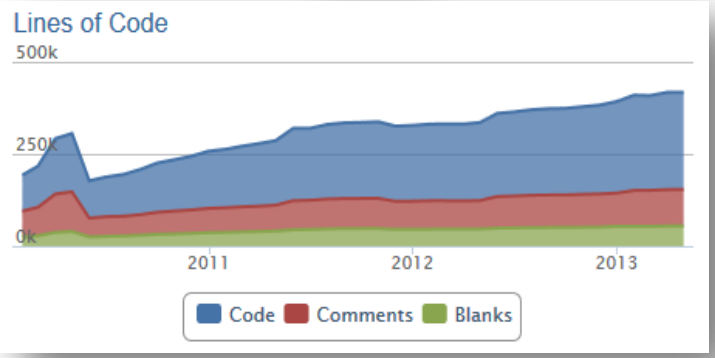
Active Contributors

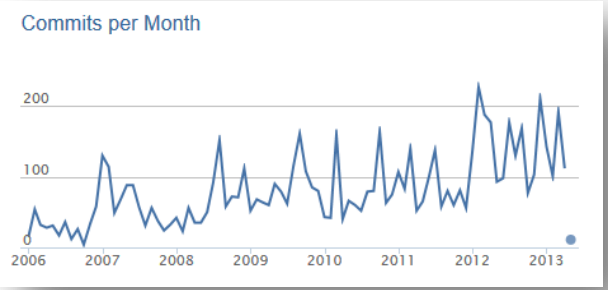


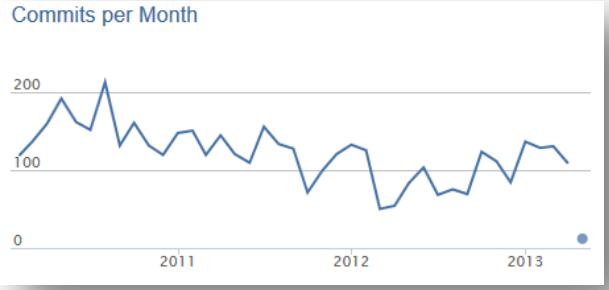


Lines of Code

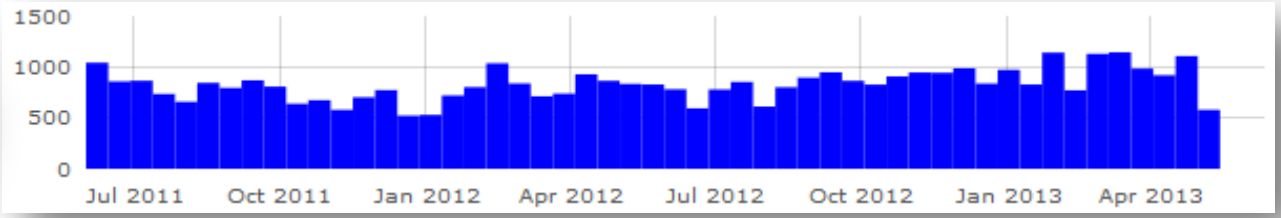




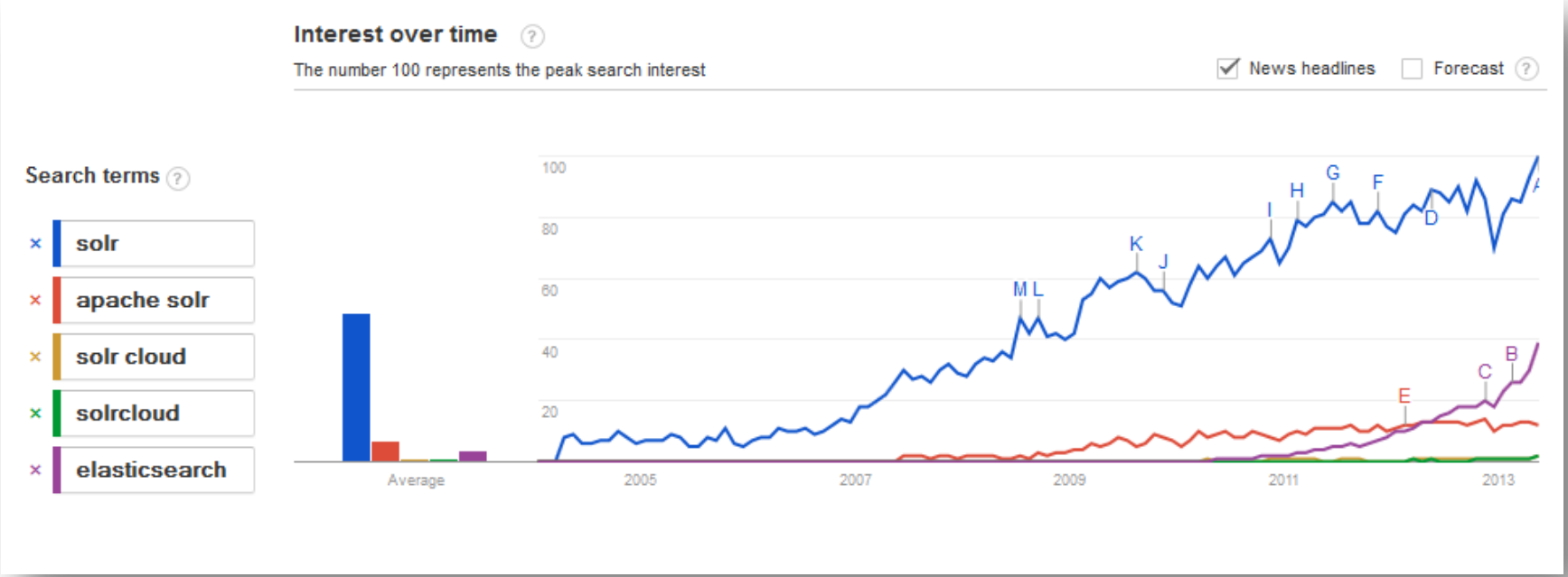




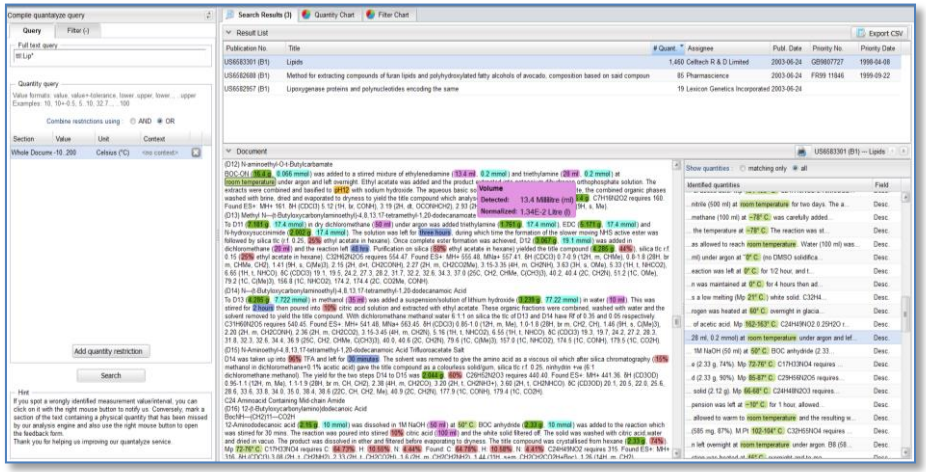
The Mailing Lists



Interest over time

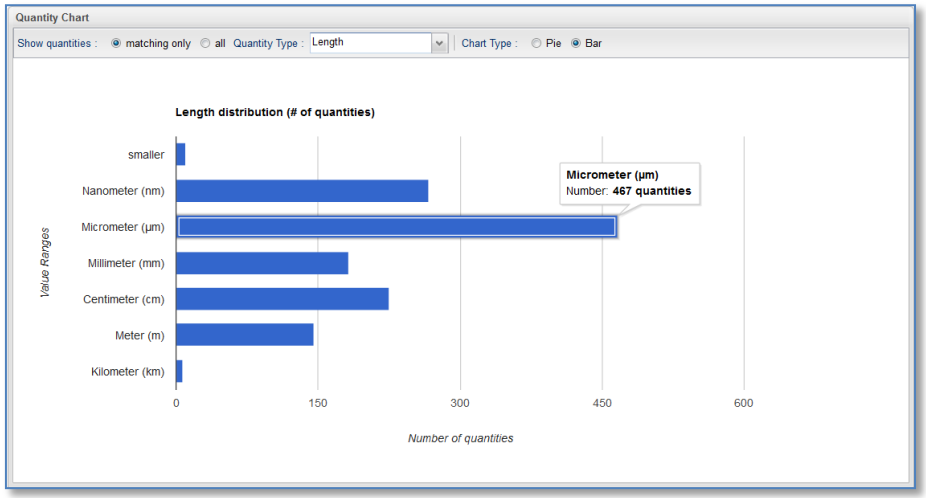


Case study - quantalyze



patents.quantalyze.com

- Runs in your browser
- Filter and keyword search
- Physical quantity search
- Interval search
- Print view



Visualization:

- Physical quantity distributions
- Cross-tabulations (e.g. concepts vs. quantity type)
- Different chart types

Case study - StumbleUpon

Create a world-class customer experience

- A “Stumble” provides real-time recommendations to 30 million customers per day
- Intelligent search is key to providing fast and more informed recommendations
- Update your searches immediately with newly posted content

Develop and scale easily

- Build in intelligent search to scale with millions of users and interactions
- Take advantage of powerful and flexible APIs for easy data integration
- Use easy to use but powerful solutions for your big data search and analytics needs

Strengths of open source search

- Best practice segmented index (like Google, Fast)
- Scalability
- Best practice, flexible ranking (term/field/doc boosts, function queries, custom scoring…)
- Best overall query performance and complete query capabilities (unlimited Boolean operations, wildcards, findsimilar, synonyms, spell-check…)
- Multilingual, query filters, geo search, memory mapped indexes, near real-time search, advanced proximity operators…
- Rapid innovation
- Extensible architecture, complete control (open source)
- No license fees (open source)

Weaknesses of open source search

- Those typical of open source
  - No formal support
  - Limited access to training, consulting
  - Lack of stringent integrated QA
  - Speed of development and open source environment too complex for some (e.g., what version should I download? What patches? GUI?)
- Others
  - Lucene/Solr/Elasicsearch development has tended to focus on core capabilities, so missing certain features for enterprise search (e.g., connectors, security, alerts, advanced query operations)

Addressing open source weaknesses

- Community
  - Community has a wealth of information on web sites, wikis and mailing lists
  - Community members usually respond quickly to questions
- Consultants
  - May be especially helpful for systems integration or addressing gaps
- Commercialization
  - Companies commercializing open source provide commercial support, certified versions, training and consulting
- Internal resources

Product strengths of commercial competitors

- Well established players tend to be full-featured
- Some organizations have focused on a particular application or domain (e.g., ecommerce, publishing, legal, help desk)
- Some competitors have focused on appliance

Weaknesses of top commercial competitors

- Usually expensive, especially at scale
- Platform or portability limitations
- Limited transparency
- Limited flexibility, especially for other than intended application or domain
- Limited customization, especially for appliance-like products
- Sometimes limited scalability
- Technical debt and/or lack of rapid innovation
- Customers are dependent on the company's continued business success

Competitive landscape

- Last years commercial companies have felt increasing competition from Lucene/Solr/Elasticsearch because of the combination of its capability and price
- Some competitors have responded with diversification
- Some have been acquired

- Need for good, affordable, flexible search remains

Questions

Credits

lucid IMAGINATION

sematext