Draft - Digital forensics and Enterprise open source search engines

Joachim Hansen

May 1, 2017

1 Questions

- I might need to inspect the source code to understand what type of string search matching algorithm is used (for example Aho-Corasick String Matching).
- Look at string search algorithms as well.
- Look at Finite state transducers in apache lucene and how that relates to search performance.
- Methodology: Find Enterprise open source Search engines that is still in development (last update should be 2015/2016). Look at their documentation to find algorithms of importance. If the documentation does not provide lower level details such as which string matching algorithms are used, then use the documentation as a hauristics to identify which parts of the code to inspect to find the relevant code snippets. Compare the code snippets with known string matching algorithms and make a desicion on which it is more likely.
- string matching algorithms seems to be omitted from the documentation.
- A limitation in the existing documentation is often the string matching algorithms are not stated
- Apache Lucene - Geospatial search - Faceted search. - Full text search
- mnoGoSearch seems to be deprecated last update 2015 `http://www.mnogosearch.org/`
- Might want to add a licence type (e.g. GNU General Public License, version 2) column in the table
- Searchdaimon seems to be deprecated last update in 2015 `https://github.com/searchdaimon/enterprise-search`
- Constellio Enterprise Search engine seems to be deprecated last update in 2015 `https://sourceforge.net/projects/constellio/`

- Summa is an open source search engine developed by the State and University Library of Denmark. It is released under the Apache License, Version 2.0. DOCUMENTATION IS LACKING!!!! `https://github.com/statsbiblioteket/summa/` last updat 16 mar 2017
- DO I Exclude projects that is not well documented?
- Hsearch seems to be deprecated last update 2013 `https://sourceforge.net/projects/bizosyshsearch/?source=recommended`
- LIUS (Lucene Index Update and Search) has been deprecated `https://sourceforge.net/projects/lius/?source=recommended`
- Oxyus Search Engine might be alive? but its website is under maintanance and could not find any documentation `https://github.com/Dexn00b/Oxyus`
- cs2project is deprecated `https://code.google.com/archive/p/cs2project/source/default/commits`
- Flax Luwak uses reverse search on streams (e.g. a log file where we append in the end of the file). A normal search have many queries or complex queries that tries to match documents with the queries. If there is many queries or complex queries then this process can take alot of time of there is many new elements on the stream (log entries). But one could do a reverse search where instead of indexing documents we index queries. We index all the queries (make it more searchable) then we convert the document into a query by matching the documents terms (ORing) with the indexed queries. This process can eliminate all quires that do not match (they would not need to run) and save a lot of resouces.
- tntsearch full-text-search fuzzy-search geo-search ... `https://github.com/teamtnt/tntsearch` where is the documentation?
- `http://www.datafari.com/en/` add mainpage in table
- Elasticsearch
- Could be interesthing `http://terrier.org/`
- NO clear definition of enterprise search engine? Elasticsearch do not market itself as enterprise search, but it has alot of search capabilities. It was claimed that it is not a enterprise search engine becouse it dont have document level security? search within a enterprise.... desktop search....

list of general algorithms for information retrieval, search algorithms and dependencies:

- General algorithms of note
  - A_1: KinoSearch merge model[1]
  - A_2: Term Frequency / Inverse Document Frequency (TF/IDF) ranking algorithm
  - A_3: Stemming - Reducing words to their root.

- Search algorithms

  S_1: Full text search

- Major dependencies

  D_1: SWISH::3 is Perl module that interface with libswish3[2]. Libswish3 is a document parser that parse documents into a data structure that can be more easily processed by a search engine[3].

  D_2: OpenSearch provides different formats for the search results[4].

  D_3: Apache Lucy is a "loose C port of the Apache Lucene search engine library for Java" [5].

  D_4: Plack handles HTTP request and delegates tasks to Apache Lucy[6].

  D_5: loose definition of enterprise search (some also claim that itv includes desktop search). Maby the inclusion of search engines should be (current development + well documented + open source + has atleast search capabilities beyond some level).

Table 1: Placeholder

Name	Algorithms	Dependencies	Last update
Dezi	A_1?,A_2,A_3,S_1 Bookmarked ranking algorithm [5],[7], [8]	D_1,D_2,D_3,D_4 [6]	commit 2e82578 28 Nov 2016[9]
Apache Solr [10]			Solr v 6.4.2: 06 Mar 2017 [11]
Sphinx [12]			Sphinx 2.3.2-beta 8 Sep 2016 [13]
Indri [14]			25 Jan 2017 [**?**]
OpenSearchServer [15]			commit 5b3e89d 13 Jan 2017 [16]
Luwak [17]			commit 8336487 6 mar 2017 [17]
datafari			commit 72f2bf0 23 mar 2017 [18]

References

[1] Lucy-Wiki. KinoSearchMergeModel; 2009. Accessed on 20.03.2017. Available from: `https://wiki.apache.org/lucy/KinoSearchMergeModel`.

[2] DEZI. Swish3;. Accessed on 20.03.2017. Available from: `https://dezi.org/swish3/`.

[3] karpet. libswish3; 2009. Accessed on 20.03.2017. Available from: `https://github.com/karpet/libswish3`.

[4] opensearch. Introduction;. Accessed on 20.03.2017. Available from: `http://www.opensearch.org/Home`.

[5] Lucy A. Overview;. Accessed on 20.03.2017. Available from: `http://lucy.apache.org/`.

[6] karpet. THE STACK; 2014. Accessed on 20.03.2017. Available from: `https://github.com/karpet/Dezi/blob/master/lib/Dezi/Architecture.pod`.

[7] Lucy A. TF/IDF ranking algorithm;. Accessed on 20.03.2017. Available from: `https://lucy.apache.org/docs/perl/Lucy/Docs/IRTheory.html`.

[8] Lucy A. DESCRIPTION;. Accessed on 20.03.2017. Available from: `https://lucy.apache.org/docs/perl/Lucy/Analysis/SnowballStemmer.html`.

[9] karpet. karpet/Dezi; 2016. Accessed on 20.03.2017. Available from: `https://github.com/karpet/Dezi`.

[10] Solr. Learn more about Solr;. Accessed on 22.03.2017. Available from: `http://lucene.apache.org/solr/`.

[11] Apache. Index of /lucene/solr/6.4.2; 2017. Accessed on 22.03.2017. Available from: `http://apache.uib.no/lucene/solr/6.4.2/`.

[12] Sphinxsearch. Full-Text Diary; 2017. Accessed on 22.03.2017. Available from: http://sphinxsearch.com/.

[13] Sphinx. Sphinx 2.3.2-beta downloads; 2016. Accessed on 22.03.2017. Available from: http://sphinxsearch.com/downloads/beta/.

[14] Lemur. Indri; 2016. Accessed on 22.03.2017. Available from: https://www.lemurproject.org/indri.php.

[15] OpenSearchServer. Downloads and documentation; 2017. Accessed on 22.03.2017. Available from: http://www.opensearchserver.com/.

[16] emmanuel keller. OpenSearchServer; 2017. Accessed on 22.03.2017. Available from: https://github.com/jaeksoft/opensearchserver.

[17] romseygeek. flaxsearch/luwak; 2017. Accessed on 26.03.2017. Available from: https://github.com/flaxsearch/luwak.

[18] julienFL. francelabs/datafari; 2017. Accessed on 26.03.2017. Available from: https://github.com/francelabs/datafari.