Open Source Search: A Data Mining Platform*

Wray Buntine
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki & Helsinki University of Technology
P.O. Box 9800, FIN-02015 HUT, Finland
wray.buntine@hiit.fi

Abstract

Commercial search engines provide a quality service at no cost to consumers thanks to embedded targeted marketing. Despite this, I argue there are still reasons why an open source effort should be encouraged in the community: as part of broader open publishing initiatives, to allow quality subject specific search engines to develop, and "because we can," because search is one of the great grand challenge problems and we, the research community, cannot join in without an accessible, non-proprietary system. This talk outlines arguments and discusses some of the new technology that could go into such a system, as well as the infrastructure that would be required to make it work.

1 Introduction

Open Source Search is a topic made popular with the Nutch[1] software, launched in 2003 and partly funded by Yahoo. In this talk, I outline the motivation for developing open source search as developed in our ALVIS project[2]. Arguing for an open source project requires both a business and a technology case, so both these issues are considered here first. Then infrastructure that would seem necessary for such a project are considered. Some related aspects of this can be found at the theme website `http://opensourcesearch.org`.

2 Motivational Background

I first discuss aspects of the operating systems competition between Microsoft and Linux, and then the current business environment for search. This sets the background for the basic arguments.

*A revised version of an invited talk given at the The Fourth IEEE International Conference on Data Mining, Nov. 2004, Brighton, UK.

[1] An extension of Lucene by Doug Cutting, see `http://www.nutch.org`.

[2] Project website `http://www.alvis.info`.

2.1 Case Study: Microsoft and Linux

Linux was started in the early 90's by University of Helsinki student Linus Torvalas, and grew with the extensive collection of open source software developed in the GNU project. Linux is now starting to get use in cost-sensitive desktop applications in places like Spain, Germany, Mexico, Thailand and Peru, and is considered the biggest threat to Microsoft's dominance, according to their own 2003 10K filing. Microsoft is now actively engaged in political, hardware and software standards efforts in an attempt to thwart open source and Linux. Microsoft's large financial resources, aggressive business strategies, entrenched market share, and the proprietary nature of its defacto industry standards (such as Word's document format) all combine to make direct commercial competition with it difficult.

All of this, of course, would not matter were operating systems not such a critical resource in areas such as intelligence, finance, defence, science and low cost computing in education. So within these specific areas Linux is only able to act as a competitor ("with potential", since its market share is small) because of the viral nature of the GNU Public License and the general philosophy and culture of open source. Development is open and shared, and a productive culture of survival of the fittest emerges as the free-market of developers competes to make the system work. Linux has had a slow but study uptake as it gradually extends its domain of competence. But Linux is viewed by any realist as an alternative to Microsoft's operating system, suited in some areas and not others.

2.2 The Business of Search

In 2004, search is driven by advertising and has become a one billion dollar business, with the largest share going to Yahoo and its subsidiary Overture (using pay for placement in results) and Google (using targeted keyword advertising). Search is essential for navigation in the Internet, just like directories such as Yahoo were in the early stages. Dominant players have emerged as consolidation and market share changes have rapidly diminished competition in the last few years. Overture has patented the pay for placement concept and Google has a broad patent on link-based authority measures that includes PageRank^{TM} (arguably, the best practical *authority model* for web pages), making their respective positions very strong. Google was recognised in 2004 as the number three business-to-business company, along side such giants as the Wall Street Journal. Microsoft has recently entered the search business, both because of the business potential, and to challenge Google's forays into its own area.

The business of keyword search is a natural monopoly just like operating systems: "size matters" when it comes to coverage and when it comes to providing fast response time. It would not matter were it not such a critical information resource. Some analysts believe that no new monolithic search engine for the Internet as a whole can emerge as a direct competitor to existing search engines. The scale of investment and development required is too great. The extent of Google's investment can be seen in their large scale clustering resources [3]. Of course, some venture capitalists currently disagree, but even Microsoft's efforts here are not reviewed favourably.

2.3 Small-Scale Alternatives

¿From a general user perspective, there are few hard reasons to want an alternative to major search engines. Issues such as lack of local content and lack of personalisation apply. For instance, Exalead, a French search engine, claims better local content is one of their key leverage points in France. But nevertheless, major search engines do a reasonable job, they are free, and they

are constantly innovating. Fortunately, as in the Microsoft-Linux business case, we have strong arguments for wanting alternatives in some specific areas.

Opportunities for improved search in subject specific areas exist where a combination of new technology and tender loving care can combine to create a better service. One parallel business area does this already. The so-called Enterprise Content Management community touts phrases such as "better return from your digital assets." Making a better search service is viewed as part of their task. These services, however, are largely targeted at high-end customers with prices to match.

The open source world usually develops in a low cost environment. Some communities that would like specialised search services, not in a global scale, are as follows:

Alternative Languages: smaller European language communities that challenge current keyword search are the minority languages with a rich morphology, Estonian, Slovenian. Larger languages that challenge standard search engines are Turkish (rich in morphology) and Chinese (which lacks word segmentation), but these have adequate commercial incentive so an open source option is not needed. For instance, the Chinese provider `http://www.baidu.com` is considered by some to be the best Chinese language search engine.

Digital Libraries: libraries require richer user interfaces and better document and access control than the standard search engine, but nevertheless, they see the need for better entry into the Internet landscape.

Publishing Initiatives: a number of open publishing/access initiatives are growing on the web to foster publication or distribution of content on the web, e.g., self publishing via `http://cafepress.com`, and archival reports on `http://www.openarchives.org`.

Academic Special Interest Groups: academics have their own document genres and sometimes rich ontologies, named entities and keywords that they would hope to incorporate into the search process, e.g., `http://linguistlist.org`.

Several other categories of communities along these lines exist. Analysis of these communities reveals the potential for incorporating additional capability into a search engine: subject categories, genre categories, named entities, keywords, etc. In digital library applications, for instance, this kind of feedback and capability is valued [5].

Thus, opportunities exist to build software for subject specific search, and these are the intended target for discussion to follow. Moreover, we, the Computer Science Community, have other reasons for making this software happen.

2.4 Search as a Grand Challenge

Some computer scientists argue that *intelligent searching of the Internet*, the next generations beyond keyword search, is a task that holds pride of place with high profile challenges such as computer programs against the World Chess Champion, soccer playing robots, or sending robots to Mars. This is a problem involving natural language processing, information extraction and general artificial intelligence, although it is in its traditional area of speciality, information retrieval, where a wide body of relevant knowledge already exists. It is a problem of international scope and clear need that has its origins and its solutions firmly in computer and information science.

To let general researchers access this grand challenge we can build an open source search engine. This would serve the Computer Science, Information Retrieval and Intelligent Systems communities in the same general way that open source operating systems, database systems,

compilers, interpreters and web servers have. They not only serve as excellent research and educational tools, they can also support a wide variety of applications from Internet service providers, Wall Street, to low-cost educational computing in the developing world. Open source initiatives may be just a small blip on the commercial screen, in some cases barely affecting the major commercial players. However, they would provide a fabulous resource to the R&D community, as well as an important commodity to cost-conscious organisations that may provide services.

2.5 Summary

So the case for open source search is not unlike the case for operating systems, i.e., Linux vs. Microsoft, with one clear distinction: global search engines provide a good service that is *free*, and thus there is no cost incentive for consumers to move away.

Open source search will only be able to grow if it develops options and capabilities not easily found at the global search engines, and that means harnessing the capabilities of experts within their own domains. For instance, the "Environmental Search Engine" must provide services unique to its own domain, for instance recognised lists of corporations, pollutants and species, relevant subject categories, better selection of material, cataloguing of material under genres, etc.

Open source search can target small-scale alternatives where individual commercial incentives are inadequate. Open source search then leverages its products across many such alternatives. The grand challenge of intelligent search then becomes accessible to researchers across many institutions, and with the right designs their developments can be integrated into the system.

3 Infrastructure for Search

I have argued that open source search should provide a platform in specialised domains, and a platform for research that may lead to new opportunities. Two of the most promising technology areas for search the ALVIS group see are peer-to-peer systems [4, 9] and information extraction (IE), which the GATE group[3] describe as processing "unrestricted text in order to extract information about pre-specified types of events, entities or relationships". Recently, IE has also been shown to be able to develop simple ISA hierarchies [8], used for instance by Semantic Web systems. Thus information extraction can semi-automate the task of tagging web content, the task that is generally considered a major hurdle for Semantic Web progress.

The marriage of these three technologies, peer to peer systems, smaller subject specific search engines, and information extraction, is perfect for search in the open source context: information extraction tools provide the technology to help domain experts customise smaller search engines without large investments of time or sophisticated programming efforts; peer to peer systems become more efficient at information retrieval when the nodes are topically oriented [6]; and, peer to peer systems provide the technology to make some sense out of a network of smaller search engines, to let users have a single access point to a network of such services.

In the ALVIS project we have begun the design of an infrastructure to support this kind of effort. Our overriding design principle is that we need to have an open system with separated functioning parts interacting through clearly defined interfaces. We need to be able to harness the efforts of the diverse experts that can contribute to the overall system and existing software available either as open source or free for non-commercial use such as information retrieval systems, Lucene, Zettair, Lemur, information extraction systems MALLET [7], GATE, and

[3] `http://gate.ac.uk/ie/`

enumerable tools for shallow parsing and crawling. As an example, Attardi's group has invested considerable effort in tuning the IXE indexer [2] and a separate IXE parallel crawler. We would like such a group to contribute their focussed research and development without them becoming experts in all the other functioning parts of the system.

There are many ways of viewing search engines, but a general decomposition we have arrived at which takes into account the additional needs of information extraction and peer-to-peer querying is as follows:

Crawl: this can be made quiet independent, and several good systems are available for this stage. This is also an area where peer-to-peer systems excel.

Document Processing: general document processing and integration of content into the runtime system.

Collection Processing: collection level processing to maintain dictionaries, categories, genres and other collection-wide resources used by the system. Maintenance of the Google spell check would fall here, as does the process of ontology discovery used in a system with some semantic web support.

Runtime: the unified indexing system and the system for generating document displays with snippets, keywords, and so forth, that would be viewed as a single node in the peer-to-peer network.

Query processing: query processing and ranking system that handles both linguistic preprocessing of queries, and the peer-to-peer services supporting querying and distributed indexing.

User Interface: the actual interface the user sees, along with particular formatting and content relevant to their domain.

Once experts from various research communities started co-operating on the design of this platform, a number of significant issues arose that needed to be addressed. At the core, we needed common formats that could allow independent components to be added and integrated as needed without tight procedural or functional integration. Peer-to-peer services add complications to several of these components, and a general peer-to-peer framework is given in [1].

The basic *document processing component* is given in Figure 1. Here we expect third party software to contribute to the basic steps. The data is carried in an XML format and we provide companion tools for compression and decompression plus reconstruction of key data (e.g., tokenisation, sentence segmentation), which may not be carried in the compressed version. With this, foreign tools such as commercial named entity recognisers in molecular biology, telephone and address recognisers, or bibliographic recognisers can be readily integrated into the process.

Canonical Document Formats: general structural processing of content cannot be written to operate on all kinds of material, PDF, email, Word, etc., thus we developed a simplified structural format containing just sections, lists and links. Documents are first converted into this format and the general processing applies to this. The original document is kept for reference purposes.

Linguistic Annotation: natural language processing (NLP) of any kind results in annotations made to a document, for instance to tag a noun phrase as a *person*, or to give the basic dictionary forms or lemma of a word and part of speech (e.g., for "ran" or "runs" it would be "run/VERB"). Annotation is a long studied problem in this area. An ISO proposition (TC37SC4/TEI) is being developed by the NLP community, and we have adopted a simple variation of this. Note linguistic annotation can be very bulky, thus XML compression needs explicit guidance here.

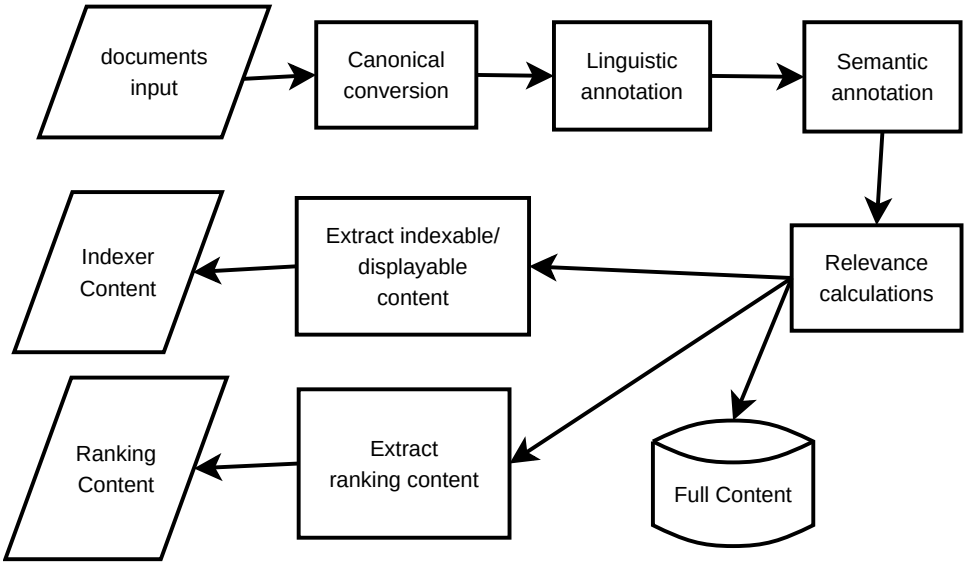

Figure 1: Document Processing

Semantic Annotation: General semantic annotation of the kind envisaged in the semantic web is not currently feasible in an automatic mode. Yet identification of basic classes of named entities (person, place, etc.), and classification of sections (for instance, using machine learning to classify text content into one of 10 predefined categories, or creating those categories from scratch) is feasible.

In order to make a *user interface component* as independent as possible, we have developed an XML format for representing search results. This has the ability to incorporate the grouping of documents, auxiliary annotations, and auxiliary categorisations found in many current avant garde search engines. For instance, keyword information, topic categorisation, geographic information, relevant names of people or organisations, or other auxiliary information could be included in this format.

The design of a *crawler component* interface is simpler and could follow the open source crawler, Grub, for instance, that allows for ranked URLs to be input, and document content to be returned with some auxiliary collection data.

4 CONCLUSION

General Internet search is a profitable advertising business and not suited to open source efforts. But if targeted at the right communities, search and the many spin-offs available from it via information extraction offer a fabulous opportunity for open source development. With newer techniques available such as peer-to-peer systems and information extraction, networks of subject specific search engines could evolve. However, to make this work, we need some common standards and open architectures to allow independent development and ease of integration of different components.

Acknowledgements

This material borrows many of the ideas from the ALVIS Consortium's submissions to the EU's 6th Framework Programme, and benefits from interactions with the ALVIS partners. Many thanks also to Katharina Morik for supporting the talk, and Giuseppe Attardi for reminding me of key software issues.

References

[1] K. Aberer, F. Klemm, M. Rajman, and J. Wu. An architecture for peer-to-peer information retrieval. In *SIGIR workshop on Peer-to-Peer Information Retrieval*, 2004.

[2] G. Attardi and A. Cisternino. Reflection support by means of template metaprogramming. In *GCSE*, pages 118–127, 2001.

[3] L.A. Barroso, J. Dean, and U. Hölzle. Web search for a planet: The Google cluster architecture. *IEEE Micro*, 23:22–28, 2003.

[4] J. Callan and N. Fuhr. The SIGIR peer-to-peer information retrieval workshop. *SIGIR Forum*, 38(2), 2004.

[5] O. Drori. How to display search results in digital libraries-user study. In *NDDL 2003*, pages 13–28, 2003.

[6] J. Lu and J.P. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *CIKM*, pages 199–206, 2003.

[7] A.K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[8] C. Nedellec. Ontologies and information extraction. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*. Springer Verlag, 2004.

[9] W. Nejdl. How to build Google2Google - an (incomplete) recipe. In *International Semantic Web Conference*, pages 1–5, 2004.