Metasearch engine for Austrian research information

Marek Andricík
Vienna University of Technology

Summary

Majority of Austrian research relevant information available on the Web these days can be indexed by web full-text search engines. But there are still several sources of valuable information, which cannot be indexed directly. One of effective ways of getting this information to end-users is using metasearch technique. For better understanding it is important to say that metasearch engine does not use its own index. It collects search results provided by other search engines, and builds a common hit list for end users. Our prototype provides access to five sources of research relevant information available on the Austrian web.
Keywords: search engine, metasearch, ranking, transformation rules, Perl, CGI, parsing, forward.

1 Search engines

Basically, web full-text search engines can be divided into two main categories:

1. General, big well-known search engines, which try to index the „whole Internet“ (e.g., Google, Altavista, Yahoo, Excite, Lycos). They utilize their own special software, which, in most cases, is not open to public and details about implementation and interface are available very seldom. It is not rare case that even among these engines exists significant incompatibilities in query languages.
2. Specialized, usually smaller, topic-based or area-restricted engines. Small engines do not have to have the „military grade“ level like their general counterparts - they do not handle very big load and even middle-sized computer can host them. Also, software requirements and mainly data storage optimization are not so high. Several commercial and free solutions are available on the Internet these days.

Note: On the homepages of search engines one can usually find also the directory service. „Cheap“ implementation, often found on engines aimed at advertisement, is done as a gateway to the search service itself (e.g., Epilot). Professional engines maintain directories by hand. Yahoo provides these days most credited directory.

According to Sergey Brin and Lawrence Page [1], first search engines started appearing in 1994 with hundreds of thousands of indexed documents. Three years later they reached tens of millions of indexed documents. Nowadays, indexes of big well-known full-text search engines reach milliards ($=10^9$) of indexed documents. Together with the growth of indexes also the average number of queries raised from thousands per day at the beginning to hundreds of millions per day now. Such a big amount of data and traffic has to have corresponding processing power and storage capacity behind (e.g., Google consists of 10 000 computers in cluster, has several tens of terabytes of storage capacity and has over 2 milliard documents indexed [2]).

Major, well-established metasearch service providers claim that no single search engine can have all available documents indexed [5]. Having on mind the fact, that Internet is decentralized heterogeneous network, practically none single search engine can have up-to-date index of all available documents. There are several factors, which can turn search results unsatisfactory:

1. The engine does not know about the existence of some particular.
2. Documents on the web may change anytime. Search engine will not get any notification about changes. The only practical way of being up-to-date is polling every document in the index and check for changes. It is clear that it cannot be done immediately for all indexed documents - they have to be checked step by step. Depending on the nature of the document, its attributes or configuration of the search engine, checking period vary. Simple consequence of this fact is that search engines sometimes return links which do not conform to the query (document has changed) or even points „nowhere“ (document was deleted). In both cases, search results do not reflect reality.
3. There still exist sources of research relevant data, which cannot be directly crawled and indexed, but they can be searched through proprietary search interface. Mostly, they are dynamically on-the-fly generated documents from databases, which could have static URL but not every single document is listed anywhere and the only other way to get it would be „guessing“ its URL. One is thus forced to use proprietary search interface.

The existence of several search engines working concurrently, means that some of them already could index documents which others did not. It also effectively raises the probability, that the right and up-to-date document is found when several search engines are queried together. This is the time when the metasearch comes onto the scene to help end user overcome some problems and simplify querying process.

2 Metasearch engines

First metasearch engines appeared early, just one year later after regular search engines did. There are several ideas behind the metasearch, one of the most important is bringing „the best of all worlds“, which in terms of searching, means collection of relevant documents. When user decides to search for documents using several search engines, metasearch service comes handy. Not only it provides much more comfortable way how to get list of documents but it also saves the time. It has to be said, that metasearch engine does not have its own index nor it combines indexes of other search engines nor even it has direct access to them.

In the previous section some bottlenecks of search engines were identified. Let us examine how metasearch deals with problems:

1. Finding document. Metasearch directly will not help. Query is submitted to several engines at the same time, theoretically, bigger area is covered, which raises the chance of finding documents.
2. Tracking document changes. Applies the same as above.
3. Accessing documents behind proprietary search interface. Previous two cases could be considered supplemental but this one is the case where metasearch brings significant improvement over regular search. The way how general search engine crawl the web and collect documents fails in the situation when the is no complete list of all available documents - simply, search engine only follows links but does not submit forms, which are usually the only way how to get to those documents. On the other hand, metasearch engines have natural support for submitting forms. It is their main task to „simulate“ user.

Every coin has two sides, so does the metasearch. Major problem is that query languages used by different engines differ. It is usual problem of finding „common ground“. Some support full range of boolean operators (and, or, not), suffixes and grouping of terms with parenthesis. It is rare, but one can sometimes find search engines supporting even prefixes and infixes.

Each query has its syntax and semantics. When metasearch engine accepts primary query from end user it has to submit semantically the same query to every engine participating at the search. Depending on supported syntax of each particular engine submitted secondary queries would

very likely syntactically differ. When the richness of query languages differ, it is possible that some complex queries cannot be expressed in all languages. Developers of metasearch engines has two possibilities (not counting the case when they ignore the problem):

1. Define reduced common grammar of metasearch engine, so that any primary query can be transformed to all secondary queries.
2. Define simplifying rules, which will be applied when primary query cannot be transformed to secondary queries.

In the first case it is guarantied that results obtained by metasearch will be the same as results obtained by manual searches (assuming the same conditions on the search engines). But, it must be noted, that more complex queries (those, which cannot be, expressed in common grammar) are not allowed. In second case, it can happen that search results differ. Very likely it occurs when transformation rules are applied. Awkward consequence is that in case of more complex queries, metasearch engine must really submit modified or even simplified query, possibly with different meaning. Results are very likely to be different when compared to results obtained from search engines queried directly. [4] Since the process is quite complex, it is very hard to tell in advance what will happen. Somewhere between these two extremes lies our prototype.

3 Our metasearch prototype

All what was said in previous sections naturally applies also to Austrian research relevant information. Some of them can be indexed directly - and they are indexed on the regular basis by our instance of mnoGoSearch, web full-text search engine. For the rest (data, which cannot be indexed directly) there is metasearch engine. Current prototype is implemented as stateless CGI gateway written in the Perl language. As it is usual with the CGI technology, the program processes only request at the given time. In case of several parallel requests, each of them will have its own independent instance of running program. Program operates in several steps:

1. Accepts primary query and using transformation rules transforms it into set of secondary queries. The table of features of each search engine supports user's decision about the level of complexity of the query.
2. Requests are in parallel dispatched and metasearch engine then waits until all search engines deliver results or until timeout occur.
3. Responses are serially parsed; title and URLs are extracted and put into the final list of links.
4. If requested, the list is sorted according to ranking.
5. List of documents is displayed together with number of hits and time spent for each engine.

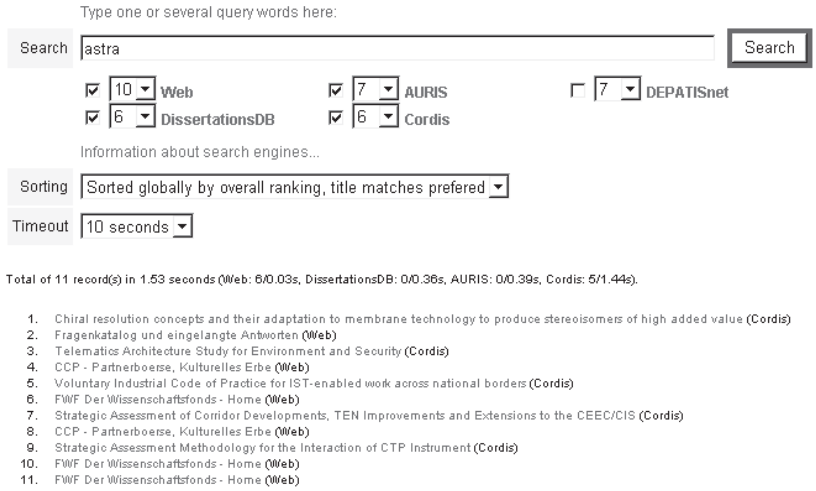

Figure 1: Metasearch results

3.1 Transformation rules

Current prototype has support for five search engines:
- Our mnoGoSearch search engine - indexes web documents
- Österreichische Dissertationsdatenbank - database of dissertations
- AURIS - Österreichische Forschungsdatenbank - old version of research portal
- Cordis - Community Research & Development Information Service
- DEPATISnet - German Patent and Trade Mark Office

Each of them has varying support for complex queries. All of them support the AND and OR operators but, one does not support the NOT operator and only minority has full boolean, phrase and support for the * and ( ) operators.

According to the content of the document there are four categories defined:
- Persons - researchers, project leaders, ...
- Institutes - research units, ...
- Projects
- Results - dissertations, thesis, patents, ...

Table on the page lists either level of support of operators and content coverage for each particular engine (see) . Using transformation rules, each primary query is converted to set of secondary queries. Why information retrieval standards as (e.g. Z39.50) are not used? Such standards can be utilized only when both sides participating at search support them. Vast majority of search engines provides only web interface accessible only through HTTP.

Name	AND	OR	NOT	( )	Full boolean	Phrase	*	Persons	Institues	Projects	Results	Comment
Web	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	Mnogosearch base full–text search engine
AURIS	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	AURIS – Osterreichische Forschungsdatenbank
DEPATISnet	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	German Patent and Trade Mark Office
DissertationsDB	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗	✓	Austrian Research Centers – Dissertationsdatenbank
Cordis	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	Community Research & Development Information Service

Figure 2: Feature table

3.2 Ranking

engine has very small amount of information, which can be used during the process of ranking (metainformation). Moreover, very few engines list numerical rating, which can be used by metasearch. Algorithm used by the prototype assumes, that partial results from each engine come already ordered by relevance and preserves that order. Furthermore, links whose title contains some of the searched words are pushed towards the top of the list. Search engines have also so-called overall ranking number assigned. Not only it is the way to measure quality of the sources but, it also allows each user to select his own preferred engines and rank it higher.

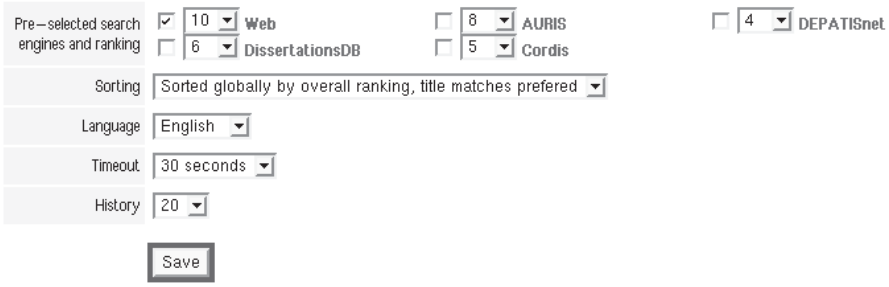

Figure 3: Customization page

3.3 User interface

User interface is designed to be highly customizable. One can work anonymously or can choose his login and password. Since then the system can keep user's preferences (pre-selected engines, their overall ranking, language and sorting criteria). User do not have to login every time, until he manually logs off his session remains valid. Moreover, user can use his login from several computers - he will have the same preferences selected. Logged users have one more advantage: reusable history of their queries. It is question of time to add more features (e.g., statistical information).

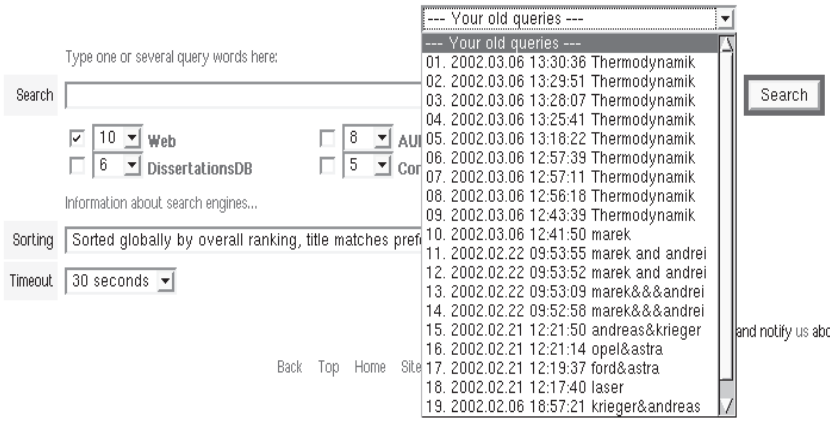

Figure 4: History of queries

4 Conclusion

CRIS users already noticed that nowadays there is a big information overload and vast majority of documents does not follow any structural or formatting standards. Full-text search becomes more and more important over structural search and the importance of search engines raises every day. Metaseach technology can help to work around limitations of search engines.

Although current version of the prototype is fully functional there still exist areas for improvement. Alternative to stateless CGI-based implementation is standalone server application, which can provide query-cashing, smooth, breaking of long listing and slightly faster responses.

The system already provides several benefits:

- Give end user comfortable way to access several research relevant data sources in one run.
- Through the simple yet powerful forward feature of the user interface overcome major drawback of metasearch engines. There are direct links to the search pages of every listed engine. After the search they are changed to links which simulate querying process on particular engine.
- Allows end user to customize behavior of the engine.
- It is general enough to be installed elsewhere and tuned for different set of search engines.

Major drawbacks of metasearch are manual search engine addition by skilled person and requirement for maintenance. This problem will be faced during the next development phrase. Another weakness which has to be noted is that user should be aware of limitations both of search and metasearch.

5 References

Brin, Sergey; Page, Lawrence: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Science Department, Stanford University.
http: //www-db.stanford.edu/pub/papers/google.pdf

Google Features page.
http: //www.google.com/help/features.html

3.Search Engine Watch - the authoritative guide to searching at Internet search engines.
http: //searchenginewatch.com
The Seven Habits of Highly Effective Web Searchers. Peachpit Press, 2001.
http: //beta.peachpit.com/vqs/73401/excerpt.html
MetaCrawler History. Metacrawler Press Center. InfoSpace, 2000.
http: //www.metacrawler.com/press/bg.html
mnoGoSearch - Full Featured Free Web site Open Source Search Engine.
http: //www.mnogosearch.org

6 Contact Information

Marek Andricík
Vienna University of Technology
Gusshausstrasse 28 / E015
A-1040 Vienna, Austria

e-mail: andricik@derpi.tuwien.ac.at