EU-IST Project IST-2003-506826 SEKT
SEKT: Semantically Enabled Knowledge Technologies


# D1.1.1 Collecting Data for Ontology Generation

Blaž Novak, Blaž Fortuna ,
Dunja Mladenic, Marko Grobelnik
J. Stefan Institute, Slovenia

## SEKT Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2003-506826.

**British Telecommunications plc.**
Orion 5/12, Adastral Park
Ipswich IP5 3RE
UK
Tel: +44 1473 609583, Fax: +44 1473 609832
Contactperson: John Davies
E-mail: john.nj.davies@bt.com

**Empolis GmbH**
Europaallee 10
67657 Kaiserslautern
Germany
Tel: +49 631 303 5540
Fax: +49 631 303 5507
Contactperson: Ralph Traphöner
E-mail: ralph.traphoener@empolis.com

**Jozef Stefan Institute**
Jamova 39
1000 Ljubljana
Slovenia
Tel: +386 1 4773 778, Fax: +386 1 4251 038
Contactperson: Marko Grobelnik
E-mail: marko.grobelnik@ijs.si

**University of Karlsruhe, Institute AIFB**
Englerstr. 28
D-76128 Karlsruhe
Germany
Tel: +49 721 608 6592
Fax: +49 721 608 6580
Contactperson: York Sure
E-mail: sure@aifb.uni-karlsruhe.de

**University of Sheffield**
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1891
Fax: +44 114 222 1810
Contactperson: Hamish Cunningham
E-mail: hamish@dcs.shef.ac.uk

**University of Innsbruck**
Institute of Computer Science
Techikerstraße 13
6020 Innsbruck
Austria
Tel: +43 512 507 6475
Fax: +43 512 507 9872
Contactperson: Jos de Bruijn
E-mail: jos.de-bruijn@deri.ie

**Intelligent Software Components S.A.**
Francisca Delgado, 11 - 2
28108 Alcobendas
Madrid
Spain
Tel: +34 913 349 797
Fax: +49 34 913 349 799
Contactperson: Richard Benjamins
E-mail: rbenjamins@isoco.com

**Kea-pro GmbH**
Tal
6464 Springen
Switzerland
Tel: +41 41 879 00
Fax: 41 41 879 00 13
Contactperson: Tom Bösser
E-mail: tb@keapro.net

**Ontoprise GmbH**
Amalienbadstr. 36
76227 Karlsruhe
Germany
Tel: +49 721 50980912
Fax: +49 721 50980911
Contactperson: Hans-Peter Schnurr
E-mail: schnurr@ontoprise.de

**Sirma AI EOOD (Ltd.)**
135 Tsarigradsko Shose
Sofia 1784
Bulgaria
Tel: +359 2 9768, Fax: +359 2 9768 311
Contactperson: Atanas Kiryakov
E-mail: naso@sirma.bg

**Vrije Universiteit Amsterdam (VUA)**
Department of Computer Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
Tel: +31 20 444 7731, Fax: +31 84 221 4294
Contactperson: Frank van Harmelen
E-mail: frank.van.harmelen@cs.vu.nl

**Universitat Autonoma de Barcelona**
Edifici B, Campus de la UAB
08193 Bellaterra (Cerdanyola del Vall` es)
Barcelona
Spain
Tel: +34 93 581 22 35, Fax: +34 93 581 29 88
Contactperson: Pompeu Casanovas Romeu
E-mail: pompeu.casanovasquab.es

## Executive Summary

In this report we give an overview of different approaches for collecting data from the Web where the goal is to collect documents on a target predefined topic. The described approaches are identified in the literature under the name "Focused Crawling". The main idea of most of the described approaches is to use the initial "seed" information given by the user to find similar documents by exploiting (1) background knowledge (ontologies, document taxonomies), (2) web topology (following hyper-links from the relevant pages), and document repositories (through search engines). The general assumption for most of the "Focused Crawling" methods is that pages with a similar content are better connected between each other then to the rest of the web. In the cases where this assumption is not true (or we cannot count on it) we can still use the methods for selecting the documents through search engine querying. In general we could say that "Focused Crawling" serves as a generic technique for collecting the data for the next stages of data processing such as building and populating ontologies for the Semantic Web

## Contents

# 1 Introduction

Web search engines (e.g. Google, Altavista, Teoma) collect data from Web by "crawling" it – performing a simulated browsing of the web by extracting links from pages, downloading all of them and repeating the process ad infinitum. This process requires enormous amounts of hardware and network resources, ending up with a large fraction of the visible web* on the crawler's storage array. Since the data collected in WP1.1 will be mainly used for ontology creation which does not require all of the information present on the Web, a specialization of the aforementioned process called "focused crawling" will be used, enabling us to collect only relevant information in much shorter time.

# 2 Focused crawling

The Web in many ways simulates a social network: links do not point to pages at random but reflect the page authors' idea of what other relevant or interesting pages exists. This information can be exploited to collect more on-topic data by intelligently choosing what links to follow and what pages to discard. This process is called "focused crawling".

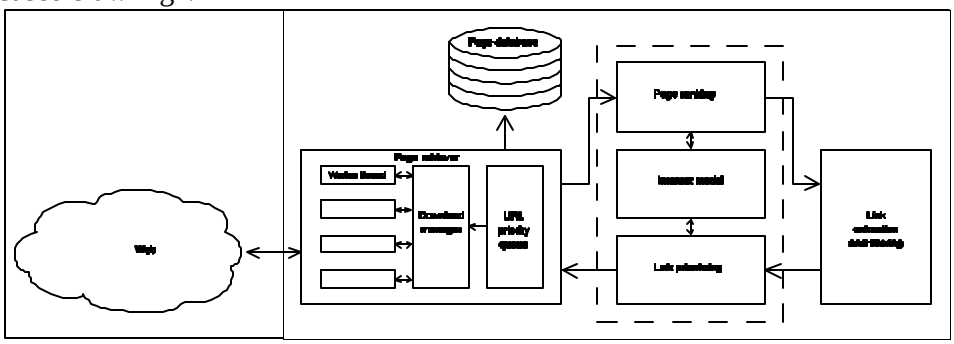

Figure 1: Architecture of focused crawlers.

Figure 1 shows typical architecture of a focused crawler. Initial (seed) pages are put in a priority queue and subsequently downloaded. Download manager enforces several constraints including download speed and rate of pages located on a single host and domain while still trying to comply with URL priorities. That way slow remote servers and links are not overloaded by requests. Retrieved pages are then evaluated for topic relevance. This may range from simple keyword matching to complex machine learning classification schemes. Hyperlinks found on pages are extracted and ran through a filter. One possible reason for link to be omitted from crawl is a presence of 'do not follow' META tag on the source page. It is also possible for the webmaster to specify parts of the site not to be indexed. Compliance with this so called 'Robots Exclusion Protocol' is not mandatory and can be administratively ove rridden. The crawler administrator can also specify a list of pages and sites to be excluded from the crawl – for example to avoid infinitely large automatically generated crawler traps. The next step is to predict the usefulness of following each link based on information seen so far and enqueueing it. Gathered

* "Visible web" is the part of the Web that can be accessed by only following the links. The vast majority of the structured information is however only accessible through constructing and submitting apropriate queries through web forms.

pages can then be postprocessed and possibly the prediction model updated with new information. A non-focused crawler lacks the components marked with a dashed rectangle.

Focused crawlers are usually evaluated by "harvest rate" which is the ratio between number of relevant and all of the pages retrieved. "Loss rate" is equal to 1 minus harvest rate.

In the rest of the report the page on which a link was found will be called 'parent page' and the one to which the link points 'child page'.

# 3 Existing research

## 3.1 Crawling without external help

Some early work on the subject of focused collection of data from the Web was done by [DeBra94] in the context of client-based search engine. Web crawling was simulated by a "group of fish" migrating on the web. In "fish search" each URL corresponds to a fish whose survivability is dependant on visited page relevance and remote server speed. Page relevance is estimated using a binary classification (the page can only be relevant or irrelevant) by a means of a simple keyword or regular expression match or an external program. Only when fish traverse a specified amount of irrelevant pages they die off - that way information that is not directly available in one 'hop' can still be found. On every document the fish produce offspring – its number being dependant on page relevance and extracted link count. The school of fish migrates in the general direction of relevant pages that are then presented as results. Starting point is specified by the user by providing 'seed' pages that are used to gather initial URLs. By adding URLs to the beginning of the crawl list this is a sort of a depth-first search.

[Hersovici98] extends this into "shark-search" algorithm. URLs of pages to be downloaded are prioritized by taking into account a combination of source page relevance, anchor text and neighbourhood (of a predefined size) of the link on the source page and inherited relevance score. Inherited score is parent pages relevance multiplied by a specified decay factor. Unlike in [DeBra94] page relevance is calculated as a similarity between document and query in vector space model and can be any real number between 0 and 1. Anchor text and anchor context scores are also calculated as similarity to the query.

[Cho98] propose the use of PageRank score of the parent page calculated on the graph induced by pages seen so far for URL prioritization. They show some improvement over the standard breadth-first algorithm. The improvement however is not large. This may be due to the fact that this PageRank is calculated on a very small non-random subset of the web and is also too general for use in topic-driven tasks [Menczer01, Menczer02].

## 3.2 Crawling with the help of local background knowledge

[Chakrabarti99] use an existing document taxonomy (e.g. pages in Yahoo tree) and seed documents to build a model for classification of retrieved pages into categories (corresponding to nodes in the taxonomy). The use of a taxonomy also

helps at better modelling of the negative class: irrelevant pages are usually not drawn from a homogenous class but could be classified in a large number of categories with each having different properties and features. In this paper the same applies for the positive class because the user is allowed to have interest in several non-related topics at the same time. The system is built from 3 separate components: crawler, classifier and distiller. The classifier is used to determine page relevance (according to the taxonomy) which also determines future link expansion. Two different rules for link expansion are presented. Hard focus rule allows expansion of links only if the class to which the page belongs with the highest probability is in the 'interesting' subset. Soft focus rule uses the sum of probabilities that the page belongs to one of the relevant classes to decide visit priority for children and no page is eliminated a priori. Periodically the distiller subsystem identifies hub pages (using a modified hubs&authorities algorithm [Kleinberg98]). Top hubs are then marked for revisiting.

Experiments show almost constant average relevance of 0.3 – 0.5 (averaged over 1000 URLs). Quality of results retrieved using unfocused crawler almost immediately drops to practically 0.

In [Chakrabarti02] page relevance and URL priorities are decided by separate models. Baseline model for evaluating page relevance can be anything that outputs a binary classification. The model for URL ranking (also called "apprentice") is on-line trained by samples consisting of source page features and the relevance of the target page (which is of course available only after both pages have been downloaded). For each retrieved page, the apprentice is trained on information from baseline (in this case the aforementioned taxonomy model) classifier (i.e. with what probability does the parent page belong to some class) and features around the link extracted from the parent page – to predict the relevance of the page pointed to by the link. Those predictions are then used to order URLs in the crawl priority queue.

Number of false positives is shown to decrease significantly – between 30% and 90%.

[Ehrig03] consider an ontology-based algorithm for page relevance computation. After preprocessing, entities (words occurring in the ontology) are extracted from the page and counted. Relevance of the page with regard to user selected entities of interest is then computed by using several measures on ontology graph (e.g. direct match, taxonomic and more complex relationships). The harvest rate is improved compared to the baseline focused crawler (that decides on page relevance by a simple binary keyword match) but is not compared to other types focused crawlers.

A variation of this method could be used for an iterative process of ontology bootstraping – by using the ontology being constructed for crawler guidance.

[Bergmark02] describe modified 'tunnelling' enhancement to best-first focused crawler approach. Since relevant information can sometimes be located only by visiting some irrelevant pages first and since the goal is not always to minimize the number of downloaded pages but to collect a high-quality collection in a reasonable amount of time they propose to continue crawling even if irrelevant pages are found. With statistical analysis they find out that the path history does have an impact on relevance of pages to be retrieved in future (compared to just using current parent pages relevance score) and construct a document distance measure that takes into account parent pages' distance (which is in turn based on its parents pages' distance etc).

### 3.3 Other approaches

[Angkawattanawit02] deal with improving recrawling performance by utilizing several databases (seed URLs, topic keywords and URL relevance predictors) that are built from previous crawl logs and used to improve harvest rate (percent of relevant pages retrieved). Future seed URLs are computed using BHITS ([Bharat98]) algorithm on previously found pages by selecting pages with both high hub and authority score. Topic keywords are extracted from title and anchor tags of previously found relevant pages. Link crawl priority is then computed as a weighted combination of link anchor text similarity to keywords, source page score and predicted link score. Link score prediction is based on previously seen relevance for that specific URL.

[Aggarwal01] introduce a concept of "intelligent crawling" where the user can specify an arbitrary predicate (e.g. keywords, document similarity, … - anything that can be implemented as a function that determines documents relevance to the crawl based on URL and page content) and the system adapts itself in order to maximize harvest rate. It is suggested that for some types of predicates the topical locality assumption of focused crawling (i.e. relevant pages are located close together) might not hold. In those cases the URL string, actual contents of pages pointing to the relevant one (not to be confused with the relevance of those pages!) or something else might do a better job at predicting relevance. A probabilistic model for URL priority prediction is trained using information about content of inlinking pages, URL tokens, short-range locality information (e.g. "parent does not satisfy predicate, children does") and sibling information (i.e. number of sibling pages matching the predicate so far). Results are not compared to other focused crawlers. While this might be an interesting concept, type of predicates for a specific application within SEKT will probably be known beforehand and can therefore be hand-optimized.

### 3.4 Use of web search engines

It is not necessary to use only locally gathered data while crawling the web. Several attempts have been made to improve the harvest rate by utilizing search engines as a source of seed URLs and back-references, most notably [Diligenti00]. They try to solve the problem of "credit assignment" by using context graphs. It is pointed out that relevant pages can be found by knowing what kinds of off-topic pages link to them.

For each seed document they construct a several layers deep graph of pages pointing to it. Because that information is not directly available from the web, they use a search engine to provide backward links. Graphs for all seed pages are then merged together and a classifier is trained to recognize a specific layer. Those predictions are then used to assign priority to the page.

Other possibilities of using remote sources include querying an index search engine for a set of seed documents, for dynamically re-seeding the crawler with random relevant pages or for retrieving all of the URLs altogether by constructing appropriate queries as done in [Ghani01]. With the advent of search index services available in 'computer readable' formats (e.g. http://www.google.com/apis/) these options become even more viable and should be used if system stability is not seriously compromised.

## 4 Classification

An important part of focused crawling is classification. It is used to determine the relevance of a web page given the query, background knowledge, etc. One of the popular methods for classification is Support Vector Machine (SVM). There are several different methods based on SVMs (binary, one-class, regression, etc.) but they all share similar background. If data is presented as a set of vectors from some vector space, then SVMs search for the best hyperplane in that space to suit the job. For example, if data is split into two classes (positive and negative), than binary classification SVM searches for a hyperplane that separates this two classes best. The class to which a new document belongs is then determined by the side of the hyperplane it lies on. This approach can be generalized to nonlinear planes. This is done with special functions called kernels.

Probably the most suitable type of SVM for focused crawling is one-class SVM, also called single-class SVM. Advantage of one-class SVM is that it only needs positive examples – the relevant web pages in our case – to train the classifier. This is important because irrelevant pages found while crawling web do not form a homogenous class and therefore do not define negative class well. In other words, while crawling the web we can find a page that is not similar to anything we have seen so far. If we were using binary SVM, then it would probably be equally likely that the page was rated as relevant or irrelevant, but with one-class SVM web page is only compared to other relevant pages, query, background knowledge, etc.

One-class SVM works by wrapping a plane around data. Classification is then performed the same way as with the binary SVM – the document is in positive class if it lies one the same side of the plane as the positive class.

## 5. Conclusion

For the focused corpus collection within the SEKT project, we have decided to implement a system that exploits the search engines instead of performing a completely independent crawl. This way, it is possible to collect a set of highly relevant pages in a matter of seconds, since the search engines index covers a large portion of the web. Traditional focused crawling algorithms (including the ones that use ontology-based approach) that do not use this kind of information can not be used for interactive work and for client-side applications since they need to retrieve many orders of magnitude more data than our tool. The system is also a great starting point if the need would occur for a centralized corpus collection application that would have the sufficient resources and time available.

## 6. Appendix – User Guide

GetRelatedWebPages.exe is a focused corpus collection utility which takes an URL as an input and collects related pages from the web. It is written as a command line utility for MS-Win32 platforms. It takes 5 command line parameters:

-i: Source URL representing the focused contents (default:'')
-ou: Ouput file name where URLs for candidate pages are collected (default: 'RelatedUrls.Xml')

-oc: Ouput file name with the results of focused crawling. Pages are sorted in the order of importance (default:'FocusedCrawl.Xml')
-ocd: flag; if true, file the whole contents of the web pages is written (default:'F' (false)) in the focused-crawl
-c: Maximal number of candidates for crawling (default:-1 (all))

Example:

(1) Execution without command line parameters specified produces help information:
> GetRelatedWebPages.exe
```
====================
usage: GETRELATEDWEBPAGES.EXE
  -i:Source-Url (default:'')
  -ou:Output-Xml-Related-Urls-File (default:'RelatedUrls.Xml')
  -oc:Output-Xml-FocusedCrawl-File (default:'FocusedCrawl.Xml')
  -ocd:Output-Focused-Crawl-Document (default:'F')
  -c:Maximal-Number-Of-Candidates (default:-1)
====================
```

(2) Next example shows focused crawling for the web pages similar to the contents of URL "http://www.bt.co.uk"

> GetRelatedWebPages.exe -i:http://www.bt.co.uk

As the result, the most related are the following URLs:
1. 1.000 http://www.bt.com/
2. 0.554 http://www.rolls-royce.com/
3. 0.381 http://www.bt.com/sitemap.jsp
4. 0.275 http://www.bt.com/broadband/
5. 0.273 http://www.btplc.com/careercentre/
6. 0.246 http://www.esatbt.com/ie/aboutus/esat_group/index_print.html
7. 0.234 http://www.bt.co.uk/index_reader.jsp
8. 0.195 http://www.btplc.com/ict/
9. 0.163 http://www.abbeynational.co.uk/
10. 0.151 http://www.bms.com/
11. 0.135 http://www.vodafone.se/
12. 0.121 http://www.btglobalservices.com/
13. 0.119 http://disney.go.com/
14. 0.095 http://www.3com.com/
15. 0.093 http://www.gm.com/
16. 0.090 http://www2.bt.com/edq_resnamesearch
17. 0.089 http://www.nationalexpress.co.uk/
18. 0.087 http://www.greentourism.org.uk/
19. 0.084 http://www.eurostar.com/
20. 0.081 http://www.xerox.com/
21. 0.080 http://www.ukphonebook.com/
22. 0.077 http://www.hutchison3g.com/index.omp
23. 0.075 http://www.sprintpcs.com/
24. 0.074 http://www.berlex.com/
25. 0.068 http://www.sony.com/

## Bibliography and references

[Aggarwal01] "Intelligent Crawling on the World Wide Web with Arbitrary Predicates", C. Aggarwal, F. Al-Garawi and P. Yu. In *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, May 2001.

[Angkawattanawit02] "Learnable Crawling: An Efficient Approach to Topic-specific Web Resource Discovery", N. Angkawattanawit, A. Rungsawang. http://citeseer.ist.psu.edu/angkawattanawit02learnable.html

[Bergmark02] "Focused Crawls, Tunneling, and Digital Libraries", D. Bergmark and C. Lagoze and A. Sbityakov. http://citeseer.ist.psu.edu/bergmark02focused.html

[Bharat98] "Improved algorithms for topic distillation in a hyperlinked environment", K. Bharat and M. R. Henzinger. In Proceedings of SIGIR-98, 21st {ACM} International Conference on Research and Development in Information Retrieval

[Chakrabarti99] "Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery", S. Chakrabarti, M. van den Berg and B. Dom. In *Proceedings of the 8th International WWW Conference*, Toronto, Canada, May 1999.

[Chakrabarti02] "Accelerated focused crawling through online relevance feedback", S. Chakrabarti, K. Punera, and M. Subramanyam. In WWW, Hawaii, May 2002. ACM.

[Cho98] "Efficient Crawling Through URL Ordering", J. Cho, H. Garcia-Molina, L. Page. In *Proceedings of the 7th International WWW Conference*, Brisbane, Australia, April 1998.

[DeBra94] "Information Retrieval in Distributed Hypertexts", P. De Bra, G. Houben, Y. Kornatzky and R. Post. In *Proceedings of the 4th RIAO Conference*, 481 - 491, New York, 1994.

[Diligenti00] "Focused Crawling Using Context Graphs", M. Diligenti, F. Coetzee, S. Lawrence, C. Giles and M. Gori. In Proceedings of the 26th International Conference on Very Large Databases (VLDB 2000), Cairo, Egypt, September 2000.

[Ehrig 03] "Ontology-focused Crawling of Web Documents", M. Ehrig, A. Maedche. In Proceedings of the 2003 ACM symposium on Applied computing, Melbourne, Florida, 2003.

[Hersovici98] "The Shark-Search Algorithm - An Application: Tailored Web Site Mapping", M. Hersovici, M. Jacovi, Y. Maarek, D. Pelleg, M. Shtalhaim and S. Ur. In *Proceedings of the Seventh International World Wide Web Conference*, Brisbane, Australia, April 1998.

[Ghani01] "Building Minority Language Corpora by Learning to Generate Web Search Queries", R. Ghani, R. Jones and D. Mladenic. Technical Report CMU-CALD-01-100, 2001.

[Kleinberg98] "Authoritative Sources in a Hyperlinked Environment", J. Kleinberg. *Proceedings of the ACM-SIAM Symposium of Discrete Algorithms* , 1998.

[Menczer01] "Evaluating Topic-Driven Web Crawlers", F. Menczer, G. Pant, P. Srinivasan and M. Ruiz. In Proceedings of the 24th Annual International ACM/SIGIR Conference, New Orleans, USA, 2001.

[Menczer02] "Topic-driven crawlers: Machine learning issues", F. Menczer and G. Pant and P. Srinivasan. ACM TOIT, 2002.