Semi-automated web resource discovery and analysis:

An approach based on interactive machine learning principles

BY ASLE PEDERSEN

Submitted to the Faculty of Information and Communication Technology

in partial fulfilment of the requirements for the degree of

SIVILINGENIØR IN INFORMATION AND COMMUNICATION TECHNOLOGY

At Agder University College (Norway)

Grimstad, May 2001

Abstract

This report presents a semi-automated approach to web Resource Discovery based on an architecture in which both human learning and machine learning is integrated in the same model. The fundamental idea of the semi-automated approach is to let the machine automatically identify semantic ambiguities and ask a human analyst to resolve them. An application prototype is developed that demonstrates an experimental Focused Crawler algorithm and the use of the ISO 13250 Topic Map standard for the purpose of knowledge representation. The main area of deployment is for use in Competitive Intelligence activities on the Internet.

Content

Acknowledgements

Thanks to InterMedium for not putting any constraints on this project whatsoever and for giving me the opportunity to be creative, explore new technology and pursue those directions I deemed appropriate. This same thanks attributes to my supervisor Vladimir Oleshchuk.

Thanks also to my family and my girlfriend for standing out with me for the duration of this project. A final thanks goes to Toro for their deliciously “Rett i koppen” products and the free coffee machine in the Ericsson building that kept me wake on all those late nights.

Problem description

Outlined below is the original problem description provided for this project (in Norwegian)

Oppgavens innhold og gjennomføring:

- Det skal utvikles en prototype på en kontrollerbar menneskeassistert crawler til bruk i oppgaver der det er behov for kartlegging og analyse av informasjonskilder på internett. Crawleren skal ta hensyn til "rapid-fire" problematikken og overholde Robot Exclusion standarden. Crawleren skal (ideelt sett) med utgangspunkt i en eller fortrinnsvis flere pekere til nettsider (URL), antakeligvis supplert med nøkkelord, eksempeldokumenter eller annen forhåndskunnskap kunne konstatere at relevant informasjon finnes, hvor den finnes og i hvilken sammenheng den opptrer.

- Med utgangspunkt i de behov, problemstilling og muligheter som er skissert skal det utvikles egne modeller og metoder samt se på muligheter som ligger i forbedring av eksisterende metoder og teknikker.

- Anvendelsesområdet er kartlegging av et gitt marked og identifikasjon av aktører som operer her, for å synliggjøre markedsmuligheter og for å støtte opp om markedsorientering og strategiutvikling. Basert på prototypen skal metodegrunnlaget verifiseres gjennom casestudier innenfor anvendelsesområdet. For å oppnå et best mulig sammenlikningsgrunnlag skal det fortrinnsvis brukes minst en casestudie som på forhånd er godt dokumentert gjennom bruk av manuelle søk- og analyse metoder.

Chapter 1 Introduction

The main objective of this project has been to develop an application for semi-automatic Resource Discovery on the World Wide Web.

Some of the fundamental ideas are introduced using an extract of an interview with William Mularie of Advanced Research Projects Agency (DARPA) on “Renegotiating the human-machine interface” [21].

> “Right now, a typical *[*defence*]* analyst who wants to gain an understanding of the enemy *(competitor)* will spend most of his time scouring databases, rather than doing what humans do best, which is using deep cognitive abilities. He’s looking for not just needles in a haystack but pieces of needles. And as the world moves much faster, humans really can’t keep up. So we have to start assigning to machines more of the job of searching data, looking for associations, and then presenting to the analyst something he can understand. It’s like a prosthesis, except it doesn’t just assist the analyst, it lets him do a 40-foot pole vault. It amplifies what the human is good at.”

This statement testifies an insightful perception of the human computer interface. By omitting the word defence in the beginning of the extract and replacing the word enemy with competitor the statement could be further generalised and adaptable to any type of analyst work.

Ambitiously the proposed approach presented in this report aims to be the prosthesis Mularie is envisioning. The main idea of the semi-automatic approach to Resource Discovery is to let the machine automatically identify semantic ambiguities and ask a human operator to resolve them. An architecture was developed for this purpose in which both human learning and machine learning is integrated in the same model. The architecture is discussed in chapter 4.

1.1 Area of deployment

The main area of deployment is for use in Competitive Intelligence activities on the Internet.

Competitive Intelligence (CI), the art of gathering information about competitors, activities, and general market trends is now as important as tapping Business Intelligence (BI) techniques to understand internal operations. Organisations scan their external environment all the time in order to identify threats and opportunities, gain competitive advantage, improve long- and short term planning and developing and implementing or revising strategies [1].

Range of uses

- Mapping global competitors and players in the business landscape
- Discovery of resources deemed important to strategy issues and market orientation in an organisation.
- Market surveys and market analysis
- Needs not pursued in the belief that they were unattainable or other unidentified needs.

Out of these assumptions, our challenge is first of all to confirm that relevant information is to be found, where to find it and in what connection it is to occur. This activity will be mentioned as Resource Discovery throughout in this report.

1.2 Background

This section aims to give a brief introduction to the World Wide Web and provide a brief discussing of the shortcomings of the existing Web search-engines.

1.2.2. The World Wide Web

According to a study released in October 2000, the directly accessible "visible web" consists of about 2.5 billion pages, while the "invisible web" (dynamically generated web pages) consists of about 550 billion pages, 95% of which are publicly accessible [2].

By comparison, the Google [66] search engine index released in June 2000 contained 560 million full-text-indexed pages. In other words, Google — which, according to a recent measurement, has the greatest coverage of all search engines — covers only about 0.1% of the publicly accessible web, and the other major search engines do even worse [3].

1.2.3 The search-engines

It is both a public opinion and a throughout documented fact that today's existing search-systems, developed for the World Wide Web, are not serving acceptable replies on requested information. What we see as non-acceptable results can be divided in two categories; those with too many hits and on the other hand the ones with not enough. The first category is referred to as the Abundance problem i.e. the number of hits that could reasonable be returned by search-system is far too large for a human user to digest. The latter category is by Kleinberg [4] characterised as a Scarcity Problem.

The main cause to this problem is that it is difficult to represent and communicate interests and information needs to the search-engines in a completely satisfying way. In addition search-engines fail to exploit the knowledge and experience possessed by the user in order to refine search results. The result of this is that much work need to be redone each time we want to search for something even if we did a similar search just a short while ago.

The World Wide Web is the world's largest database but it is not organised as a traditionally database, which makes it very resource demanding to search in. The search-engines solve this problem by indexing everything. Imagine creating a traditional back-of-book index by taking every single word in the book, removing a couple of hundreds of the most obviously useless ones and then including every single usage of those that remain. Even with some intelligence to allow for inflected forms and synonyms the result would be of no practical used whatsoever.

The problem with full text indexes such as those search-engines provides is their lack of discrimination. What a manager at a company needs to know is different from what people that work for him care about. Also because the search-engines are to answer all thinkable questions, they do not have particularly good assumptions to present the results in an intuitively and easily understood way.

1.2.4 The call for alternative approaches to Resource Discovery.

“There are many paths that lead to Rome”, the difference lies in how long it takes to get there and also the amount of recourses we have to use to get there.

Assuming we have enough time, experience, knowledge and systematic abilities, it is possible to find most of the information we need using the search-engines. However lack of resources, especially when it comes to time and combined with lack of experience and knowledge urge for a need to automate this work and the belonging analysis. This will bring more time on other activities deemed more interesting.

Because the search-engines do not satisfy the needs declared in the range of uses, this calls for the development of new tools based on new ideas. The goal is thus not to replace or renew the search-engines as we know them today. We are to satisfy a so far uncovered need which the search-engines or other commercial available tools have in only small amount, or not at all, paid attention to.

Chapter 2 Related work

This chapter will introduce some of the previous and ongoing research in the area of Web information retrieval, Focused Crawlers and Competitive Intelligence software.

2.1 Competitive Intelligence Software

The “Intelligence Software Report 2000” [5] compiled by Fuld & Company [6] discusses in detail the requirements for Competitive Intelligence software and introduces some of the currently available products. Only a few Resource Discovery products for Competitive Intelligence on the Web are available and none of those are considered to be complete implementations according to the requirements standards set by Fuld & Company. Cispider is one of the products mentioned in the report and this software seem to take an approach to Resource Discovery in a similar way as the solution proposed in this report. CISpider is developed at the AI lab of University of Arizona and implements a combination of a Focused Crawler, Noun Phrase extraction and Self Organizing Maps [7]. Besides this the effort is only vaguely documented on the homepages of University of Arizona.

2.2. Focused Crawlers

The notion of Focused Crawlers descends from research undertaken at the IBM Almaden Research Center [8] in co-operation with the Department of Computer Science at University of California, Berkeley [9]. One of the prototypes developed for the Clever projects [10] starts with 200 pages that are the results of an ordinary search-engine key-word search. It then adds all pages that link to, or are linked to by those 200 pages. This step typically swells the set of pages to 1.000 pages or more. The pages are ranked according to the two classes of pages: hubs and authorities. (The idea of hubs and authorities will be discusses later on in this report.) Results are also ranked according to a scoring function based on where a term appears in a document. Several papers have been presented on these topics in the past few years, most recently in a paper by Soumen Chakrabarti at the Tenth World Wide Web Conference [11] on integrating the Document Object Model (DOM) to enhance web information retrieval [12]. Many of the methods presented in this project exploits ideas similar to those of Clever and its successors.

The overall impression this far is that compared to the currently available search-systems the research on Focused Resource Discovery undertaken in academia are by all means specialised and focused, but compared to the mentioned range of uses, they may be considered too generic. This demands for further specialisation in order to be employed in Competitive Intelligence applications. Fine grained methods for segmented hubs, named micro-hubs proposed later on in this report is one attempt to attain such specialisation, which is also (coincidently) backed up by the recent work presented by Chakrabarti at WWW10 in May 2001.

2.3 Knowledge representation and Topic Maps

SemanText [13][14] is a prototype application developed to demonstrate how the Topic Map standard (ISO/IEC 13250:2000) can be used to represent semantic networks. SemanText builds a knowledge base, in the form of a semantic network, from the Topic Map. SemanText is written in Python and utilises a Topic Map processor called tmproc. Although not having defined the same targets, at least from the point of view of tools utilised in SemanText and the use of Topic Maps for knowledge representation required it to be mentioned in this chapter.

Chapter 3 Topic Maps

This chapter is an introduction to the Topic Maps, which is the empowering technology used to implement the proposed solution to be introduced in the next chapter.

Dubbed “the GPS (Global Position System) of the information universe”, Topic Maps are envisioned to provide powerful new ways of navigating large and interconnected corpora. Topic Maps federate and exploit different worldviews simultaneously even if those worldviews are cognitively incompatible with each other [15].

The Topic Map initiative urge that information users should not be forced to use a single ontology, taxonomy, glossary, namespace or other implicit worldview. They also express that finding information should be application- and vendor neutral so that users can freely exploit it in many ways and contexts.

This is why new methodologies are called for and Topic Maps provide an approach that marries the best of several worlds, including those of traditional indexing, library science and knowledge representation, with advanced techniques of linking and addressing.

3.1 ISO 13250 - The Topic Map standard

This section is introduced with an excerpt from the Topic Map standard followed by a brief introduction to the Topic Map technology. Most of the introduction is based on excerpts from the introductory paper “The TAO of Topic Maps – Finding the Way in the Age of Infoglut” by Steve Pepper [16].

“Topic Maps enable multiple, concurrent views of sets of information objects. The structural nature of these views is unconstrained; they may reflect an object oriented approach, or they may be relational, hierarchical, ordered, unordered, or any combination of foregoing. Moreover, an unlimited number Topic Maps may be overlaid on a given set of information resources” [17]

The Topic Map standard was finalised in 1999 and published as ISO/IEC 13250:2000 in January 2000. ISO 13250 is based on SGML, however in the advent of XML, an initiative delivered the core of an XML interchange syntax for Topic Maps, the XTM 1.0 specification in December 2000. A revision to XTM 1.0 was released in February 2001. The basic concepts of the Topic Map model is Topics, Associations and Occurrences (TAO)

3.1.1 Topics

A topic, in its most generic sense, can be any “thing” whatsoever – a person, an entity or a concept – regardless of whether it exists or has any other specific characteristics. We might think of a “subject” as corresponding to what Plato called an idea. A Topic on the other hand, is like the shadow that the idea casts on the wall of Plato’s cave.

3.1.2 Topic types

Topics can be categorized according to their kind. In a Topic Map, any given topic is an instance of zero or more topic types. This corresponds to the categorization inherent in the used of multiple indexes in a book (index of names, index of works, index of places, etc.). Thus “Høgskolen i Agder” would be of type ”university”, “Grimstad” of type “city”. In other words, topic types represent a typical class-instance relationship. Topic types are themselves defines as topics by the standard. You must explicitly declare “university”, “city”, etc. as topics in your Topic Map if you want to use them as types.

3.1.3 Topic Names

The standard provides an element form for topic name, which it allows to occur zero or more times for any given topic, and to consist of one or more of the following types of name: base name (required), display name (optional), sort name (optional). The ability to be able to specify more than one topic name can be used to indicate the use of different names in different contexts, such as language, style, domain, geographical are, historical period etc. A corollary of this feature is the topic naming constraint, which states that no two subjects can have the same name in the same scope (more on scopes later). This means that “Høgskolen i Agder” could also have the name “Agder University College” in English contexts.

3.1.4 Occurrence

A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic. For example an occurrence could be an article about the topic in an encyclopaedia. An important point to note here is the separation into layers of the topics and their occurrences.

3.1.5 Occurrence roles

Occurrences may be of any number of different types such as “article”, “mention”, “commentary” etc. Such distinctions are supported in the standard by the concepts of occurrence role type. Occurrence role types are, as topic types, themselves topics.

3.1.6 Associations

A topic association is a link element that asserts a relationship between two or more topics. Examples might be “Høgskolen i Agder is *located in* Grimstad”. Just as topics can be grouped according to type and occurrences according to role, so can associations between topics be grouped according to their type. The association type for the example mentioned above is “located_in”. As with most other constructs in the Topic Map standard, association types are themselves defined in terms of topics. The ability to do typing of topic associations greatly increase the expressive power of the Topic Map, making it possible to group together the set of topics that have the same type of relationship to any given topic. This is of great importance in providing intuitive and user-friendly interfaces for navigating large pools of information.

It is important to note that we are talking about links between topics that are completely independent of whatever information resources may or may not exist or be considered as occurrences of those topics. This means that Topic Maps are information assets in their own right, irrespective of whether they are actually connected to any information resource or not. Also because of the separation between the information resources and Topic Map, the same Topic Map can be overlaid on different pools of information, just as different Topic Maps can be overlaid the same pool of information to provide different "views" to different users. Furthermore, this separation provides the potential to be able to interchange Topic Maps and to merge one or more Topic Maps.

3.1.7 Association roles

Each topic that participates in an association plays a role in that association called the association role. In the case of the relationship "Høgskolen i Agder is *located in* Grimstad" those roles might be "educational institution" and "place". The type on an association role is also a topic! Note that the names assigned to association types do not imply any kind of directionality. If A is related to B, then B must by definition be related to A. In the case of an "influenced-by" association we need to know who was influenced by whom, i.e., who played the role of "influencer" and who played the role of "influencee".

3.1.8 Identity and public subjects

Sometimes the same subject is represented by more than one topic link. This can be the case when two Topic Maps are merged. In such a situation it is necessary to have some way of establishing the identity between seemingly disparate topics. For example if reference work publishers from Norway, France and Germany were to merge their Topic Maps, there would be a need to be able to assert that the topics "Italia", "l'Italie" and "Italien" all refers to the same subject. The concept that enables this is that of public subjects, and the mechanism, used is an attribute (the identity attribute) on the topic element. This attribute addresses a resource, which identifies the subject in question as unambiguously as possible. That resource could be some official, publicly available document (for example the ISO standard that defines 2- and – letter country codes), or it could simply be a definitional description within (or outside) one of the Topic Maps. Any two topics that reference the same subject by means of their identity are considered to be semantically equivalent to a single topic that has the union of the characteristics (the names, occurrences and associations) of both topics.

3.1.9 Facets

Sometimes it is convenient to be able to assign metadata to the information resource that constitute the occurrences of a topic from within the Topic Map. To provide this capability, the standard includes the concepts of the facet.

Facets basically provide a mechanism for assigning property-value pairs to information resources. This could include properties such as "language", "security", "applicability", "user-level" etc. Once such properties have been assigned, they can be used to create query filters producing restricted subsets of resources, for example those whose language is "Italien" and user level is "secondary school student". As with occurrence role types, it generally makes sense to specify the type of the facet value, since then the power of Topic Maps can be used to convey more information about it.

3.1.9 Scope

The concept of scope is important to avoid ambiguities between topics and their characteristics. Any assignment of a characteristic to a topic is considered to be valid within certain limits, which may or may not be specified explicitly. The limit of validity of such an assignment is called is scope A scope is defined in terms of themes and themes are topics. E.g. to distinguish between “Paris” in France and “Paris” in Texas assign the scopes “France” and “USA” to the two topics. This removes any ambiguity and reduces the chance of errors, for example when merging Topic Maps. I fact, the well-designed, consistent and imaginative use of scope in Topic Maps does much more than simply remove ambiguity, it can also aid navigation by providing different views to the underlying resources.

3.2 Using Topic Maps for knowledge representation

Due to its generality and expressive power Topic Maps go far beyond meta-information modelling, in fact Topic Maps are a base technology for knowledge representation and knowledge management. Topic Maps can express facts, procedures and fairly complex relations between concepts. As such they come close to the knowledge representation field of Artificial Intelligence. Knowledge representation techniques like semantic networks or conceptual graphs turn what we know about a particular domain into a form, which a computer can understand. Searching in a Topic Map can be compared to searching in such knowledge representing structures. [18] [19]. The close similarity to semantic nets gives an idea of how Topic Maps, even without any occurrences connecting them to an information pool, can become valuable resources in their own right. Next it will be shown how Topic Maps can be used to represent the interrelation of roles products etc. that constitute corporate memory by mapping up the “Business Landscape”.

3.3 Mapping up the “Business Landscape” using Topic Maps

It seems like a natural approach to use a Topic *Map* to map up the “Business Landscape” that makes up the external environment of an organisation. The Business Landscape should embody all the important properties of the current business such as products, trademarks, ownerships, people, locations, employees, managers, competitors, affiliations, memberships, subsidiaries, technologies, investors, players, markets, events, etc. It is such Business Landscapes that will be used throughout this report to demonstrate the use of Resource Discovery application.

Topic map graphs are abstractly described in terms of nodes and arcs and visualisation of a Business Landscape, exemplified through the Topic Map community could look like this:

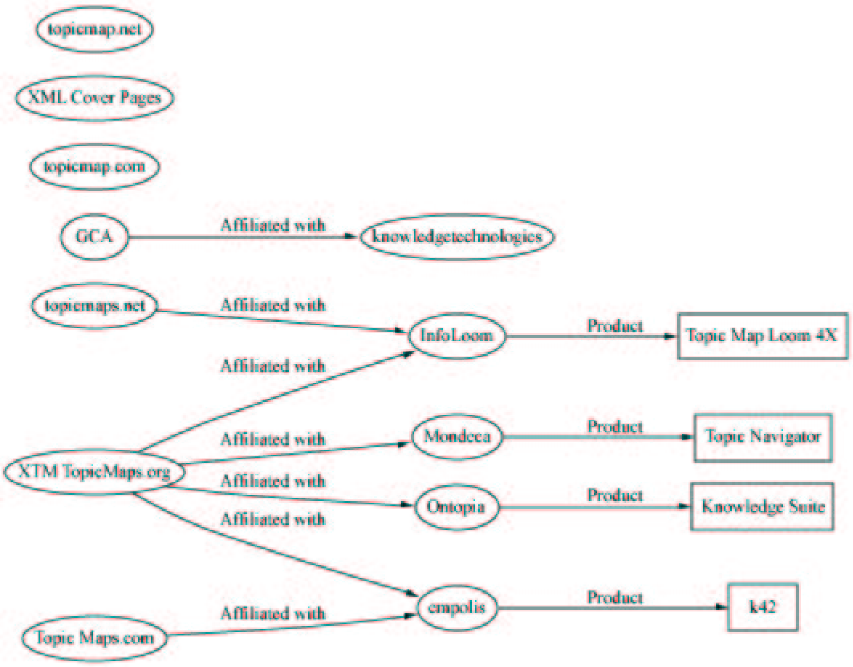


Figure 3-1. Topic Map visualisation.

In figure 3-1, boxes represents products and circles represent players such as companies, organisations or industry portals. Keep in mind that associations as defined by the Topic Map standard is inherently multidirectional and the use of arrows in the figure is thus meaningless and should strictly speaking be replaced by undirected arcs.

By first glance the graph in figure 3-1 looks fairly incomplete, with many of the nodes completely unconnected to the rest of the map. This is true, but later on we will se how the Business Landscape could be expanded.

3.3.1 Business Landscape Topic Map model

This section provides a tabular linear representation of the underlying Topic Map model visualised in figure 3-1. The model is also presented in the appendix in both LTM format [20] and SGML format.

Topic Types

Topic	**Base Name**	**Sort Name**	**Display Name**
Topic types			
Player	Player	-	Player
Product	Product	-	Product
Key Intelligence Topic	Kit	-	Key Intelligence Topic
Occurrence types			
Homepage	Homepage	-	Homepage
Press-release	Pressreleases	-	Press-release
News	News	-	News
Association types			
Produced-by	produced-by	-	Product
Subsidiary-of	subsidiary-of	-	Subsidiary
Member-of	member-of	-	Member
Affiliated-with	affiliated-with	-	Affiliated with

Table 3-1

Topics

Topic	**Base Name**	**Sort Name**	**Display Name**
Kit	Topicmap	Topic Map	Topic Map
Kit	Topicmap	Topic Maps	Topic Map
Kit	Xtm	Xtm	XML Topic Maps
Kit	Semanticweb	semantic web	Semantic Web
Player	Ontopia	-	Ontopia
Player	Mondeca	-	Mondeca
Player	Infoloom	-	Infoloom
Player	Empolis	-	Empolis
Player	Gca	-	GCA
Player	topic-maps-org	-	XTM TopicMaps.org
Player	topic-maps-com	-	Topic Maps.com
Player	topic-maps-net	-	Topicmaps.net
Player	topic-map-com	-	topicmap.com
Player	topic-map-net	-	topicmap.net
Player	Coverpages	-	XML Cover Pages
Player	knowledgetechnologies	-	Knowledgetechnologies
Product	Knowledgesuite	-	Knowledge Suite
Product	Topicnavigator	-	Topic Navigator
Product	topicmaploom4x	-	Topic Map Loom 4X
Product	K42	-	k42

Table 3-2

Occurrences

Occurrence role type	TopicID	Occurrence
Homepage	Ontopia	**http://www.ontopia.net**
Homepage	Mondeca	**http://www.mondeca.net**
Homepage	Empolis	**http://www.empolist.com**
Homepage	Empolis	**http://www.empolist.co.uk**
Homepage	Empolis	**http://www.empolist.de**
Homepage	Gca	**http://www.gca.org**
Homepage	k42	**http://k42.empolis.co.uk**
Pressreleases	Mondeca	**http://www.mondeca.com/site/news_events/press_releases.html**
Pressreleases	Empolis	**http://www.empolis.com/englisch/presse/index.html**

Table 3-3

Identities

TopicID	Identity
Ontopia	**http://www.ontopia.net**
Mondeca	**http://www.mondeca.com**
Infoloom	**http://ww.infoloom.com**
Empolis	**http://www.empolis.com**
Gca	**http://www.gca.org**
topic-maps-org	**http://www.topicmaps.org**
topic-maps-com	**http://www.topicmaps.com**
topic-maps-net	**http://www.topicmaps.net**
topic-map-com	**http://www.topicmap.com**
topic-map-net	**http://www.topicmap.net**
Coverpages	**http://xml.coverpages.org/topicMaps.html**
knowledgetechnologies	**http://www.knowledgetechnologies.com**

Table 3-4

Associations

Association type	TopicID:role	TopicID:role
Produced-by	Knowledgesuite:product	Ontopia:player
Produced-by	Topicmapnavigator:product	Mondeca:player
Produced-by	Topicmaploom4x	Infoloom:player
Affiliated-with	Ontopia:player	Topicmaps-org:player
Affiliated-with	Mondeca:player	Topicmaps-org:player
Affiliated-with	Infoloom:player	Topicmaps-org:player
Affiliated-with	Empolis:player	Topicmaps-org:player
Affiliated-with	Infoloom:player	Topicmaps-net:player
Affiliated-with	Empolis:player	Topicmaps-com:player
Affiliated-with	Knowledgetechnologies:player	Gca:player

Table 3-5

Note that this is a very simple Topic Map utilizing only the basic constructions in the Topic Map standard. Things that have not been modelled are in particular Scope. The Topics could also be provided with properties such as the significance of a player, easily modelled using the Facet construct. There are also a few other elements in the Topic Map "Business Landscape" model that requires further explanation.

Key Intelligence Topics (KITs)

Key Intelligence Topic is a commonly used term used in the Competitive Intelligence community to express critical intelligence needs. KITs are supposed to model interests to be used in the Resource Discovery Process. Basically any Topic could be defined as a KIT using multiple type declarations.

The KIT sort name is used to define the search word or phrase. Also note that a few KITs have more than one sort name such as ‘topic name’ and ‘Topic Maps’. This is required because the Focused Crawler used later on does not account for language independence or word stemming.

Identities

In order to be able to merge Topic Maps later on, as many Topics as possibly has been anchored to an identity. This is because merging of Topic Maps requires a way of establishing the identity between seemingly disparate topics from different maps. The specification of identity attributes on the topic elements that address the same public subject is the explicit solution the standard offers for merging Topic Maps. The other solution is implicitly through the topic naming constraint, which states that any topics that have the same name in the same scope refer to the same subject. In most cases the homepage of the companies or products in question are used as identities.

Chapter 4 Proposed approach

In this chapter we present the proposed approach for a semi-automated Resource Discovery architecture based on interactive learning principles. Figure 4-1 is a model of the proposed solution architecture.

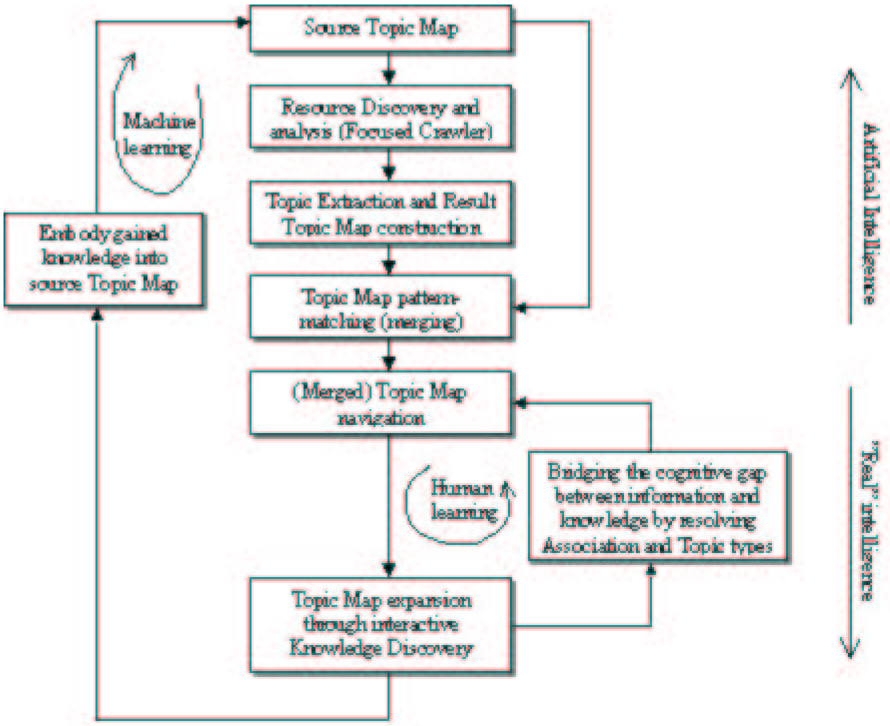

Source Topic Map
Resource Discovery and analysis (Focused Crawler)
Topic Extraction and Result Topic Map construction
Topic Map pattern-matching (merging)
(Merged) Topic Map navigation
Topic Map expansion through interactive Knowledge Discovery
Bridging the cognitive gap between information and knowledge by resolving Association and Topic types
Embody gained knowledge into source Topic Map
Machine learning
Human learning
Artificial Intelligence
"Real" intelligence

Figure 4-1. Model of the proposed approach to semi-automated resource discovery

This is supposed to be a generic model, but for the purpose of making it easier to comprehend it will be explained in relation to a given area of employment. Since we are already familiar with Topic Maps and the notion of Business Landscapes it will be explained according to those terms.

4.1 Source Topic Map

Starting at the top with the Source Topic Map, this is the Topic Map that describes the current Business Landscape. Topics like product, company, player and associations such as affiliated-with, subsidiary-of etc. are modelled in the Business Landscape Topic Map. The Business Landscape is the input to a Resource Discovery application. In this particular project the Resource Discovery application is implemented using a Focused Crawler. The Resource Discovery application automatically extracts Topics from the Business Landscape and uses those Topics as guidelines to find new resources not yet embodied in the Business Landscape. Extracted Topics from the Source Topic map includes all occurrences (Uniform Resource Locators), Identities and Key Intelligence Topics and thus the input to the Focuses Crawler is a list of URLs and a list of Key Intelligence Topics.

4.2 Topic Extractor

When the Resource Discovery application has finished searching for new resources the Topic Extractor automatically extracts entities and relations from the discovered resources. Since everything in a Topic Map is a Topic we call it Topic extraction even if the extracted Topics are modelled as Facets, Identities, Associations or any other Topic Map construct. The straight-forward entities to extract would be hyperlinks, and Meta-data. However by utilizing sophisticated extraction technology [22,23,24,25] it is also possible to extract entities such as noun phrases, peoples names, company names, geographical names, addresses, telephone numbers, sales figures, locations, etc. Topic extraction would also comprise additionally properties that could be inferred about a resource such as type and category, quality, popularity, reputation etc. E.g. inferred resource types could be classified as company homepage, personal homepage, press releases etc. Note in particular that all classes, categories and types are already modelled in the Business Landscape using the Topic Type construct. This means that if the Topic extractor is unable to resolve the Topic type for a particular Topic it should rather assign to it an empty or at least generalised Topic type and then attach to the Topic as many properties as possible e.g. using the Facet construct and leave it up to the human to resolve the ambiguities as will be explained shortly.

4.3 Result Topic Map construction

Based on the extracted Topics the Result Topic map is automatically constructed. The simplest Result Topic map constructed without using any sophisticated entity extraction tools consists only of the generalised Topic type “player” and the Association type “link-to”. All resources would thus be modelled as players and all links would be modelled using the link-to Associations. An advanced Result Topic map would use inferred properties enabling the Result Topic Map generator to assign more specific Topic types and Association types such as company instead of player and subsidiary instead of link-to.

4.4 Merging Topic Maps

After Topic Map construction the really exciting part follows – merging Topic Maps. As seen from the figure the *Source* Topic map is fed forward to the “merge” block where it is merged with the *Result* Topic Map. As has already been mentioned, the Topic Map standard has built in constructs enabling Topic Maps to be merged. Merging may be regarded as an abstract level of pattern matching. What happens in practise is that by using relationships already established between Topics in the Source Topic Map, and the newly discovered Topics in the Result Topic Map could be semantically coupled to the Source Topic Map. This demands for further explanation using an example. The Association, using Linear Topic Map notation (LTM), produce([CompanyA]:player, [Product1]:product) is established in the Source Topic Map and the Association affiliated-with([CompanyB]:player, [CompanyA]:player) appear in the Result Topic Map. If we had only consulted the Result Topic Map by its own we would not have known about the affiliation between CompanyB and CompanyA. By merging the two maps, we put meaning into the results of the Resource Discovery Application because we can in fact see them in relation to our current Business Landscape.

Graphically this could be illustrated by overlaying two half images to gain the complete picture. In the next figure boxes and circles represent Topics and the arcs represent Associations. Association types are labelled using some arbitrary types. In practice the circles could be companies and the boxes could be products and the associations could be product-of or affiliated-with.

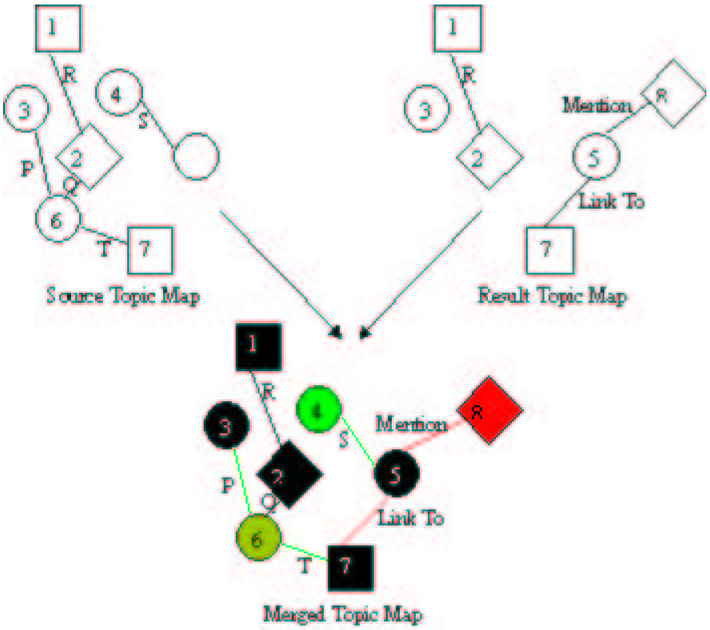


Figure 4-2. Merging Topic Map illustration

Figure 4-2 shows how the Merged Topic Map provides us with a more complete picture. The black coloured objects represents the Topics and Associations present in both the result and the source. The red coloured objects are meant to indicate new Topics and Associations not present in the Source Topic Map, and last the green coloured objects and links are meant to represent those Topics and Associations established in the Source Topic Map not found in the results.

4.5 Topic Map navigation

After merging the maps it is time for the human analyst to do some work. The human involved could be an analyst or anyone with a reasonable level of domain specific knowledge comprised by the Business Landscape. Instead of navigating the results from the Resource Discovery in terms of unrelated bits and pieces we will navigate the *Merged* Topic Map by correlating the results with the Business Landscape. This is very valuable since we could now partially ignore the black and green objects and concentrate on the red ones. However since a Business Landscape could be rather large and unhandy to navigate we might consider just looking at the part of the Business Landscape that concerns the discovered resources i.e. the red objects. In practise this would mean that we would match just a few resources with the Business Landscape at a time. The Topic Map Scope and Facet constructs enable this type of filtering.

4.6 Topic Map expansion and resolving Association- and Topic Types

From figure 4-1 we see that the next step after *Merged* Topic Map navigation is Topic Map expansion. Topic Map expansion describes the activity concerned with incorporating knowledge gained from the Resource Discovery results into the Business Landscape through interactive Knowledge Discovery in the Topic Map. The interactive exploration is represented in the figure by an iterative "human learning" process involving the steps Topic Map navigation and Resolving Association and Topic types.

This demands for further explanation using the figure 4-2 as an example. We see that the Merged Topic Map has two new Associations indicated by the red lines and one new Topic indicated by the red filled circle. To start with all new red Topics and Associations are unresolved because, as already mentioned, a computer is not very good at resolving potentially ambiguities and that requests for human assistance. To guide the decision any properties attached to the unresolved Topic and Association types such as keywords or descriptions extracted from the discovered resource should be readily available for the human operator. E.g. each object in the navigable Merged Topic map could be click able in order to obtain more information on an object. Besides this, implementing intuitive user interfaces is a challenging exercise and at this point in time it has only been investigated briefly.

As mentioned would the simplest Result Topic map only consist of generalised Topic types and in order to resolve a Topic we have to assign a proper Topic Type such as company or product. E.g. if the Resource Discovery Application discovered a link to the homepage of CompanyB from the homepage of CompanyA and the Topic extractor was unable to tell what type of relation exists between the two companies it would show up as plain link-to association in the Merged Topic Map. Imagine that due to constraints set by the Topic Map there could only exist two types of relationships between Topics of type "company". They are "subsidiary-of" and "affiliated-with". This self-descriptive property of Topic Maps works in a similar way as a power-point template file – all that needs to be provided is the content. To resolve this relationship all that is required by the operator is to decide which one of the two types it is! An imagined user-interface for resolving relationships is shown in figure 4-3.

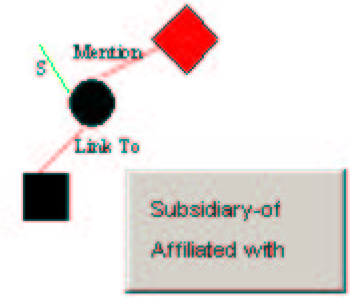


Figure 4-3. Imagined user interface for resolving Association types

As soon as the Association type has been resolved it should be reflected in the navigable Topic Map by changing the colour from red to black and labelling it with the proper Association type. Note that a link on the homepage of a company does not always indicate a tight relation and it should therefore also be possible to mark the Association in the Topic Map as weak or even non-existing e.g. using the Facet construct which could be reflected in the interface visually by removing the relation completely. The same procedure as for resolving Association types would adapt to resolving Topic types. Note that Topic types should be resolved first, because due to constraints set in the Topic Map this would reduce the number of available options when resolving the Association types.

As long as we are able to make sense of the discovered knowledge, movement ahead is possible. Each resolved Topic or Association type would bridge the cognitive gap between information and knowledge. The process is explained as iterative “human learning” in the model because resolving ambiguities requires repetitive navigation and knowledge discovery by the operator.

4.7 Back-propagating

Not to be confused with the neural network BP learning algorithm, the Merged Topic Map is propagated back to the initial starting point in order to embody the gained knowledge into the Source Topic map. This is performed when all Topic and Association types are resolved. In practice this means that the Merged Topic Map will act as the new Source Topic map. The next time we use the Resource Discovery application it will know about all the Topics and Associations discovered last time and thus the machine has learned as well as the human. The idea is that in each of the “machine learning” iterations, knowledge is gained by expansion of the Business Landscape using Resource Discovery techniques. This would thus enable the Focused Crawler to discover more resources in the next iteration because it is better informed prior to the crawling.

Chapter 5 A module based Testbed framework.

In this chapter the design and architecture of a module based Testbed is discussed.
The experimental focused crawler utilised in the Web Resource Discovery and analysis prototype presented in chapter 6 is built using the Testbed. At the end of this chapter we will see how the Testbed is used to analyse hyperlink structures, and how this can be applied in an experimental study on syndication in the online publishing industry.

5.1 What is the purpose of this Testbed?

The purpose of this Testbed is for use in experiments concerned with web information retrieval, processing and analysis of HTML document structures, and analysis of connectivity and hyperlink structures in collections of retrieved web pages.

The Testbed can be used on its own for experimental purposes such as evaluation of crawling algorithms, weighting algorithms for information retrieval and information extraction. In this way if desired, instead of spending time on the protocol level and low-level database design, the time could be spent on activities deemed more interesting.

5.2 Design goals for the Testbed

The overall goal is a generic architecture. The Testbed should be extensible to fit the needs of any particular experiment. As a result of the generic approach, the Testbed will not be optimised for use with any specific type of experiment.

The idea is that application prototypes should be built on top of the Testbed framework, and as thus the application and the Testbed will need to communicate efficiently. This may be done through the use of interfaces. The use of interfaces will clearly separate the Testbed from the application prototype and it will arrange for a seamlessly replacement of the Testbed. Such a replacement could be a further optimised module streamlined to perform any given functionality needed by the application.

A Testbed should also provide a unified method for measuring performance and because all applications or algorithms are built using the same generic building blocks it is easier to compare the results, in particular tie performance. The built-in Python profiling tools can be used to conduct such measures [26,27].

It is evident that all of these goals could not be met within the time constraints bound by this project, however they are provided here to act as guidelines for further work on the Testbed.

5.3 Testbed functionality

The basic functionality of the Testbed should provide generic means of retrieving web pages, processing the content of these pages and storing the processed data in a database. At the lowest level this involves working with Internet protocols such as HTTP, HTML document parsing, and database handling. In short the current functionality of the Testbed implementation includes:

Features

- URL redirection
- Meta-refresh redirection
- HTML document parsing with handlers for the most common tags
- Basic segmentation of HTML documents
- Frameset handling

Limitations

- Restricted to retrieval of plaintext or html document types
- Limited to pages accessible through hyperlinks (i.e. restricted to the "visible" web)

5.4 Testbed architecture

Any web information retrieval or web crawling task may be broken down into four individual activities: *Retrieve, Process, Analysis* and *Select*. These tasks also make up the major building blocks of the Testbed. The *Retrieve* activity is primarily concerned with fetching web pages and caching. The *Process* activity is concerned with parsing web pages, entity extraction and database storage. The *Analysis* activity is concerned with performing knowledge discovery on the retrieved data. The last activity is *Select*, which based on the analysis performed is concerned with selecting what pages to be retrieved next. This is with no regards to if this is done manually, automatically or semi-automatically.

Retrieving web pages often involve retrieving more pages than the initially requested page. This could be due to the use of framesets or redirecting pages. In these cases the frameset sources and the redirected page also needs to be retrieved. This is achieved by iterating over the four building blocks. First all initial pages are *retrieved*, then they are parsed and *processed* e.g. to discover meta-refresh tags used for redirecting purposes. The only *analysis* performed is to find all pages contained in framesets and redirected pages. The select activity is performed automatically by marking all the pages found in the analysis to be retrieved in the next iteration. The iteration ends when there are no more pages found fulfilling the frame or redirect criteria.

By substituting the Retrieve activity with the iteration over all four blocks and by using more advanced analysis functions and different criteria's for the selection of pages to be retrieved next, the same iteration cycle as described in the previous section could be used to describe any crawling algorithm. The diagram shown in figure 5-1 captures the entire crawling activity.

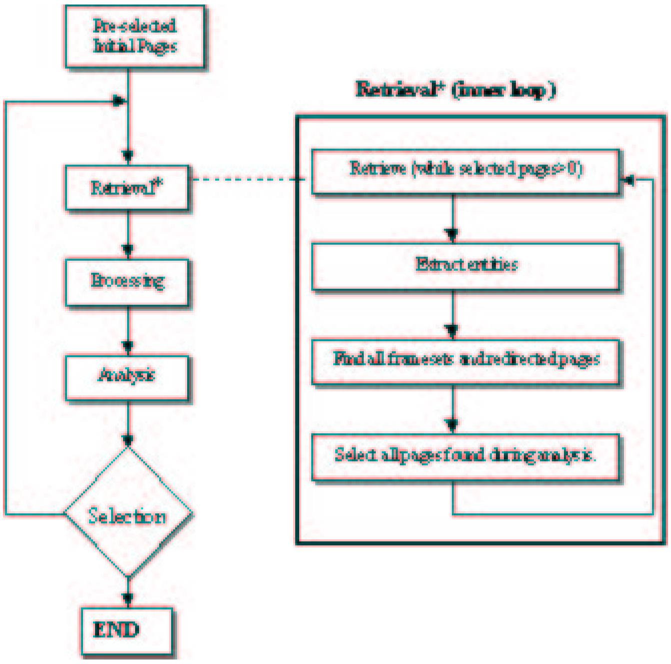

Pre-selected Initial Pages
Retrieval* (inner loop)
Retrieval*
Retrieve (while selected pages>0)
Processing
Extract entities
Analysis
Find all framesets and redirected pages
Select all pages found during analysis.
Selection
END

Figure 5-1. Testbed building blocks

Each building block is carried out in a sequential manner. This is first of all due to dependencies between the blocks, however it also makes the design clearer and it lowers the complexity of the implementation considerably. The sequential execution of blocks also arranges for human intervention in an experiment running at the Testbed. The Retrieval and Processing blocks have few dependencies and would benefit from running in parallel, as they would do in most streamlined implementations. This is due to the parallel slackness phenomenon described later on in this chapter. Internally in each block especially in the retrieval block tasks are performed in parallel.

5.5 Testbed implementation

The Testbed is implemented exclusively using the Python programming language and the MySQL relational database [28]. Both Python and MySQL are cross-platform compatible and the Testbed should thus work on a variety of platforms, however so far it has only been tested on an Intel platform running Linux. At the moment MySQL is used, however any relational database with a Python DB API-2.0 [29] compliant interface could be used instead.

Database model

The database model is URL centric, and what is meant by that statement is illustrated by the following figure.

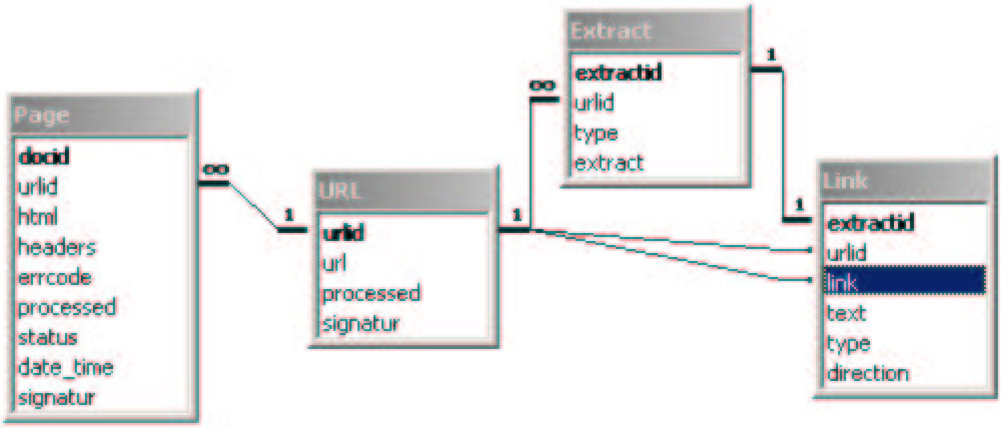


Figure 5-2. Tables and their relations.

We see from figure 5-2 that each URL may be related to an entry in the Page table and none or more extracts in the Extracted table. Each Extract is related to none or more entries in the Link table.

The Page table is used to store all web pages to be processed by the Testbed. The results of the processing is one or more extracted entities such as keywords, description, title, anchors or segments, which are all stored in the same Extract table only separated by dedicated entity-types. Any extracted entity may also consist of hyperlinks, which are stored in the Link table with one column for originating URL and one for destination URL. Since hyperlinks may point to pages not yet retrieved this demands for an entry in the URL table for each known URL.

5.6 The Testbed building-blocks

The aim of this section is to give an introduction to the Testbed building blocks. We discuss implementation issues for each individual block.

5.6.1 Retrieval

Retrieval is basically concerned with fetching web pages from remote servers. Some of the important features of retrieval include threading and caching. These techniques are discussed in this section.

Threading

Because of relatively low data dependency, fetching web pages, is an excellent task to run in parallel. This is especially important as page download size and bandwidth vary greatly. The Testbed retriever opens many connections (page requests) simultaneously. While it is waiting for an answer from the server to its request, other requests can be processed at the same time. This technique is referred to as parallel-slackness* [30].

**Parallel slackness*
Hiding the latency of communication by giving each processor many different task, and having them work on the tasks that are ready while other tasks are blocked (waiting on communication or other operations).

To achieve multiple connections in the Testbed retriever-class a thread handles each connection. The threaded retriever is a modified version of a skeletal web spider presented by Aahz Maruch at the Ninth Python Conference [31]. Only the threading code is used and the modifications are mainly concerned with error checking, such as using a TimeoutSocket, and exception handling. The Timeoutsocket is important to implement because otherwise connections could potentially have infinite timeouts, however this report will not discuss this particular problem in more details.

The retriever is using a thread pool, passing in each URL to be retrieved to a queue. This is in contrast to the brute-force thread method, spawning and killing one thread per retrieve operation. Using threadpools are thus more efficient, and provides better utilization of resources, because threads are re-used and because there's no polling on thread completion.

Threadpools was used for this project because there was a reason to believe that this method was especially suitable when retrieving many small independent objects such as web pages. However no experiments were conducted to compare the performance between the methods. Probably when working with small objects, using a threadpool will have an advantage over spawning and killing threads but this difference in performance would be less significant when dealing with larger objects because each thread would be kept alive for a longer period of time.

Fair use of remote servers (hosts)

Another important issue that was taken into consideration when designing the web-page retriever was to be fair to the servers by avoid issuing to many requests to the same server at a time. This issue is particularly important when using threads, which could potentially flood a server with requests if not taking proper precautions. When webcrawlers requests too many pages from one server, this is called rapid-fire and may take servers down, or at least make them very slow [30]. [30] solves this by putting all newly found links from different servers into a workpool from which new links are randomly drawn an examined next. The Testbed solves this in a similar fashion by storing new links in dictionaries. Because dictionaries are implemented using hash-tables, the links are queued in a randomly fashion. The distribution of requests is of course dependent on the number of unique hosts.

The Robot Exclusion protocol and Robots <META> tag

The Testbed respects the Robots Exclusion Protocol [31], the de facto standard (although unofficial), for limiting Web crawlers access within websites. It was particularly important to implement this feature in the Testbed because of the intended general-purpose usage.
The robot exclusion standard is implemented in the Testbed using a built-in Python library for parsing the Robot Exclusion Protocol. To decide weather or not a server allows a URL to be visited, a set of rules based on regular expressions is constructed for each server visited. These rules are stored in persistent objects using marshalling techniques provided by the Python Pickle module. Thus there is no need to build the rules each time the retriever is run.

The Testbed is also compliant with the Robots META tag, *NoIndex* and *NoFollow* directives, which is also used to restrict Web Crawlers access rights to index individual pages and the rights to follow links from those pages. It is important to notice however that all features discussed in this section are optional and that they may be switched off if the user of the Testbed for some reason finds it necessary not to comply with these standards.

Caching

When experimenting with crawling algorithms, the same web page would most likely be crawled several times. This votes for some type of caching mechanism so that instead of retrieving the same page from the web each times, instead a local copy is returned if it has been crawled previously. At the moment the local file system is used for caching, which is fairly straightforward to implement for small-scale systems. As filenames, the hexadecimal representation of the message digest signature (md5) on the URL is used. Each time a new page is requested both the html body and the returned headers from the server are cached, and all consecutive requests for that page are returned directly from the file system. In addition the files are compressed using Zlib [33] to save space. The use of compression does not give any noticeably loss in time performance.

Using the file system for caching could in worst-case scenarios (not unlikely to occur), result in directories filled with 100 000 and more files. For most file systems this would probably result in decreased performance. One solution to this problem could be the Reiser File system (ReiserFS) [34] which is based on fast balanced trees. ReiserFS claims that there should be no problem putting 100 000 files in one directory. ReiserFS also claims to have an improvement in small file space and time performance compared to other file systems. At the moment the default file system under Linux (EXT2) is used, but ReiserFS should be considered to replace EXT2 in the future.

Apart from the purely implementation considerations, one reason for using the file system to cache a collection of web pages (or other units of data), is that files are very transportable and platform independent compared to a database which would be the most obvious alternative. This also means that a collection of web pages used in an experiment could be easily shared and distributed among researchers and distributed on cd or dvd-roms or as file-archives on the Internet. The widely used TREC document collection [35] used in text information retrieval and natural language processing research is amongst those using this method.

5.6.2 Processing

The Testbed does provide basic processing functionality, such as HTML parsing, cleaning text, normalising URLs etc. However the HTML parser is designed to allow for any modification needed to suit a particular test configuration.

The basic HTML parser used in the Testbed is extended from the HTML parser provided with the Python standard module library. The parser provides handles for tags such the title, <Meta> tags, anchors, headings, typographic tags, list tags, paragraph-tags and table tags. By default the parser does also handle the classes of tags described in chapter 6 for use in weighting algorithms.

The HTML document processing also includes basic segmentation of HTML pages to be used in fine-grained analysis of document and link structures. Boundaries used for segmentation are at the moment tables and paragraphs with that corresponding precedence order. An example of basic segmenting is shown in figure 5-3.

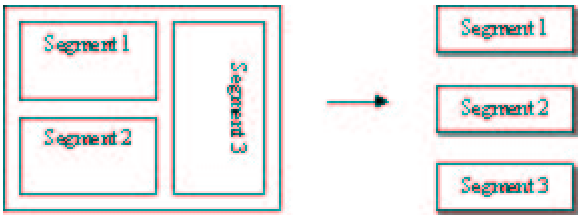


Figure 5-3. Basic segmentation process

Other processing functions performed are handling metarefresh and location headers used in redirecting, frameset handling and determination of hyperlink direction and type. The types determined are absolute and relative referencing and the directions determined are those shown in figure 5-4.

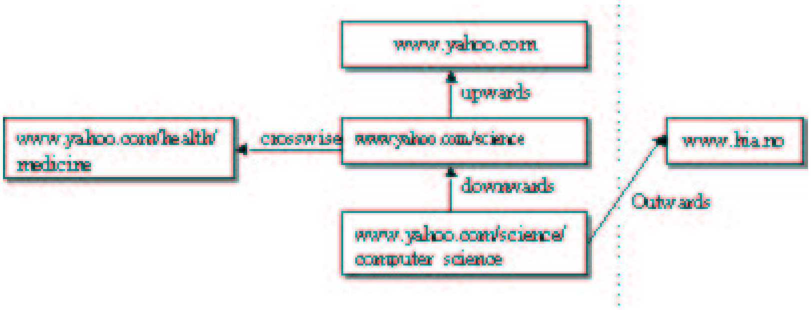


Figure 5-4. Direction types

<u>Pseudo-code describing how the data is processed:</u>

```
For each page processed:
        For each extracted entity from page:
                Insert entity into the extracted table
                        For each hyperlink in the extracted page:
                                Insert hyperlink into the link table
```

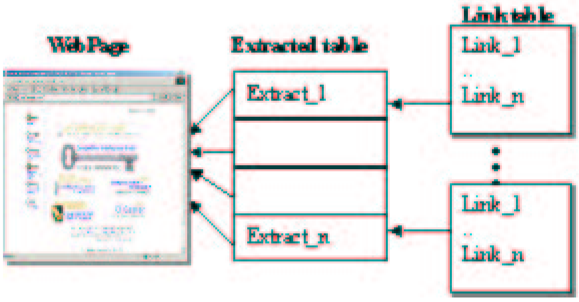


Figure 5-5. Figure showing how the data is stored

5.6.3 Analysis

By default the Testbed does not provide any analysis functionality. The user of the Testbed should provide the analysis functionality needed. The Testbed does thus initially only provide a skeleton for the analysis class.

5.6.4 Selection

Selecting pages to be retrieved can be performed either manually or automatically. In the case of manually selection, the user would be presented with a collection of pages and any information about those pages, present in the database such as metadata, or external meta-information that can be inferred about a document but is not contained within it, such as category, type, language etc. When performing selection automatically it would be based on some type of rule such as, “select all pages previously not retrieved that are in the top ten of authorities or hubs”.

5.7 Using the Testbed for experiments

5.7.1 Introduction to some simplified link analysis tools

In this section some of the requirements to carry out simple link analysis. In particular we discuss need to group pages according to some common feature such as domain and frameset.

Clustering pages based on their frame-membership and relation to redirecting pages.

Visually a web page is represented as one page, however in reality that page could be composed of several underlying frame pages. Usually we would like to treat these pages as one unit. Another technique widely used on the Internet is to page redirection. Similarly it should be considered equivalent if a hyperlink points to a page or if it point to the page it is directed to.

To illustrate this phenomenon, a link structure found during experiments using the Testbed is shown in the figure below.

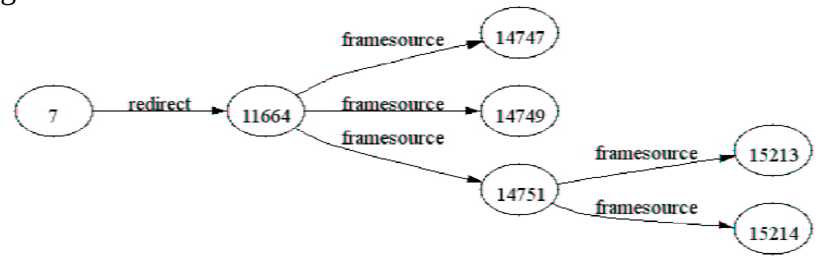


Figure 5-6. Pennet link structure

URL	Ref §
http://wi.pennnet.com/	7
http://wi.pennnet.com/home/home.cfm	11664
http://wi.pennnet.com/content/homepage_content.cfm?Section=Home&unique=WI	14747
http://wi.pennnet.com/home/topnavn/searchform_frame.cfm	14749
http://wi.pennnet.com/home/topnavn/main_topnavn.cfm	14751
http://wi.pennnet.com/home/topnavn/large_logoframe.cfm	15213
http://wi.pennnet.com/home/topnavn/topnav_main_banner.cfm?Section=home	15214

Table 5-1

To group pages considered equivalent together a simple clustering was developed. It takes as input parameter a list of tuples consisting of an ID of the referring page and an ID of the referred page. The output is a tuple consisting of, as the first item, the rootID which is the ID of what is considered as the root of the cluster and as the second item, a list of the other ID's in the cluster. With the use of this algorithm it was possible to make a lookup table that can translate an ID into its corresponding rootID. The lookup-table for the example above would look something like {7:7, 11664:7, 14747:7, 14749:7, 14751: 7, 15213:7, 15214:7}. This type of lookup-table is implemented using the Python dictionary type

Group pages according to domains, domain-names and top-level domains.

Often it is important to know what pages are in a specific domain or what pages link to a particular domain. To accomplish this, all distinct domains, domain names and top-level domains need to be found. By a domain it is mean e.g. "topicmap.org" and correspondingly by domain-name and top-level domain it is meant "topicmap" and "org". Each unique name is given a nameID and in this way a lookup-table could be constructed, which in a similar way as the previous example could map an ID to its nameID. Using such lookup-tables it is possible to group any list of URLs by domain, domain-name or top-level domain.

5.7.2 Analysis of syndication in the online publishing industry

The first experiment using the Testbed, was analysis of the link structure between online newspapers and magazines (the online publishing industry). The discovery of who link to whom could be considered as an indication of syndication or at least some kind of affiliation.

At first a rootset consisting of the homepage of 235 online newspapers and magazines was fed to the Testbed. The Retrieval block fetched all these pages, including framesets and redirected pages. The Processing block parsed the pages and extracted entities, which in turn were inserted into the Link table and the Extract table.

By utilizing the lookup-tables introduced in the two previous sections an extended version of the Testbed link table was constructed. The extended table consisted of the following columns [url, root, domain, domainname, tld, link_url, link_root, link_domain, link_domainname, direction, type].

By querying the extended link analysis table and group the result according to domain, it was possible to construct a list of what newspapers linked to each other, counting all links pointing within a domain as equal. E.g. hyperlinks pointing to **http://www.cnn.com/** and **http://www.cnn.com/technology/** were considered the same.

Note that none of the pages were crawled beyond the first page. Conducting a full experiment however would of course include fetching pages beyond the first page to discover more relationships. However surprisingly many relationships were discovered just using the hyperlinks found at root page of the websites.

Visualizing the result of the link structure analysis using Graphviz

Graphical visualization provides a strong intuitive representation of the results from the link-analysis. Graphical visualization of the link structures was constructed using the Graphviz tool [36]. The generation of graphics is done automatically by translating results in tabular form into Graphviz scripting files. The use of Graphviz will be discussed in further detail in the Resource Discovery Application prototype chapter.

The following graphs displays a selection of the link structures discovered in the experiment. Note in particular how some of the pointers are bi-directional, which may be considered as an indication of the presence of a strong relationship.

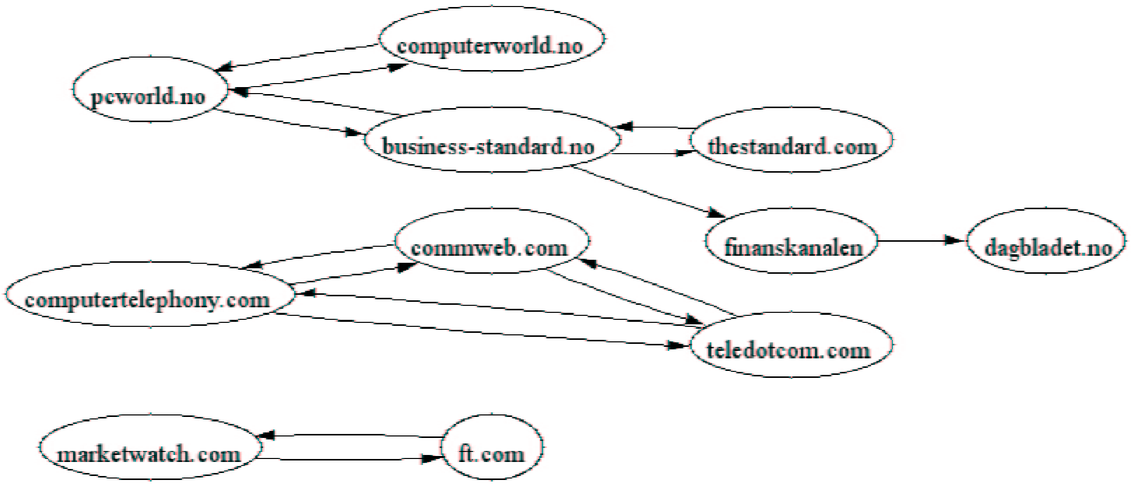


Figure 5-7. Relationships discovered using link analysis

No further conclusions will be drawn from these results. Besides the visualization of the results found in the experiment the results are provided as is and they are not discussed in further detail.

Chapter 6 Implementation of an experimental focused crawler algorithm

As mention in the previous chapter the experimental focused crawler was build using the Testbed. This was achieved by extending the default analysis functionality provided by the Testbed. The extensions include methods for fine-grained analysis of link structures and HTML document structure analysis.

6.1 Background

This section introduces some of the fundamental characteristics of the Web. In particular the Web graph structure and Social networks are discussed.

6.1.2 Web graph structure

A strongly connected core at the heart of the Web graph contains almost one-third of all Web sites, but besides this the graph structure is very complex. Although there are unconnected nodes in the web graph, areas saturated with information on similar topics tend to cluster and Web crawlers as well as Web surfers can travel between these sites via hyperlinks. For more in depth theory on this subject refer to the study on the graph structure in the web [37] presented at the Ninth World Wide Web Conference [38].

6.1.3 Social Networks

Social network theory is concerned with properties related to connectivity and distances in graphs. Social networks are formed between web pages by hyperlinking to other web pages.

Social networks and in particular the collaborative effort constituted by the Internet community incorporates a lot of semantically valuable knowledge about the web. In particular this includes the knowledge preserved in online bookmarks, portals and other types of structured and unstructured compilations of resourced. One way of exploiting social networks to gain semantic knowledge is the idea that hyperlinks to similar pages are placed spacially near each other in a Web page. This phenomenon is commonly known in bibliometric terms as co-citing.

Another idea, due to Kleinberg [4] is that there are to types of useful pages. An authority page is one that contains a lot of information about a topic. A hub page is one that contains a large number of links to pages that contain information about the topic. The basic idea is the mutually reinforcing relationship between hubs and authorities [4]. A good hub page points to many good authorities and a good authority page is pointed to by many good hub-pages. This fundamental idea is deployed by a wealth of search engines and in link analysis research.

Much human thought has gone into creating each hyperlink and labelling it with anchor text. Other valuable relational information can be gleaned from the structure, hierarchy, and similarity of peaces of text [39]. Authors of Web pages leave behind a multitude of other traces, explicitly by using Meta tags, hyperlinks and structural elements and implicitly by the tacit textual and visual expression. If it is possible to identify some form of heuristics based on qualitative and quantitative properties of the social network, the hypothesis is that these heuristics could be used to disclose those tacit but semantically valuable relations, thoughts and associations present in the authors mind at the time of creation.

6.2 Focused crawlers vs. Standard crawlers

In the following figure the difference between a Focused Crawler and a standard crawler is illustrated.

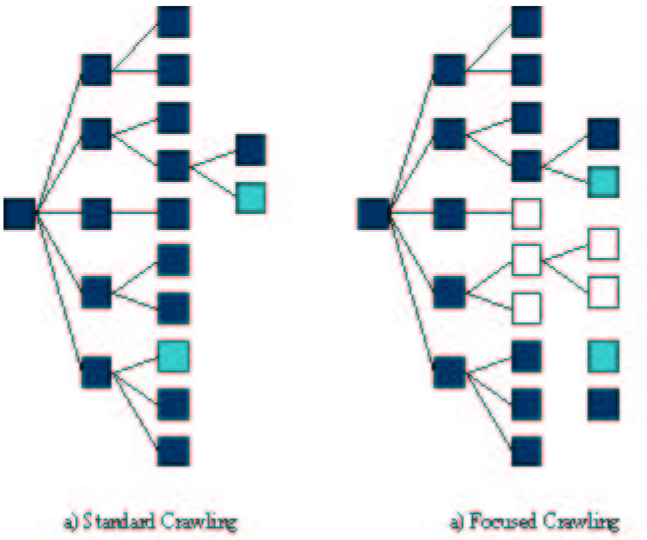


Figure 6-1. Standard vs. Focused crawling illustration

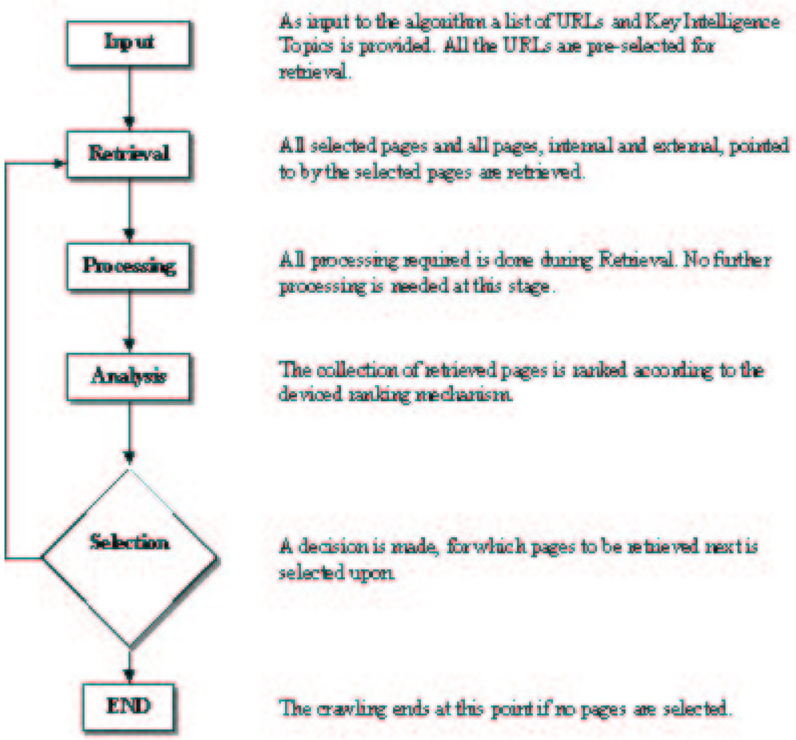
Figure 6-1. **a)** A standard crawler follows each link, typically applying a breadth first stratagy. If the crawler starts from a document which is *i* steps from a target document, all the documents that are up to *i-1* steps from the starting document must be downloaded before the crawler hits the target. **b)** A focused crawler tries to identify the most promising links, and ignores off-topic documents. If the crawler starts from a document which is I steps from a target documents, it downloads a small subset of all the documents that are up to *i-1* from the starting document. If the search strategy is optimal the crawler takes only *i* steps to discover the target [40].

6.3 The algorithm

The algorithm exploits the following properties:

- Social networks (hubs and authorities)
- Hyperlink co-citation through the use of fine-grained micro-hubs
- Topic concentration (Key Intelligence Topics)
- Inheritance of properties from pages in the neighbourhood

The following figure captures the focused crawling algorithm in accordance with the model used in the Testbed chapter.



As input to the algorithm a list of URLs and Key Intelligence Topics is provided. All the URLs are pre-selected for retrieval.

All selected pages and all pages, internal and external, pointed to by the selected pages are retrieved.

All processing required is done during Retrieval. No further processing is needed at this stage.

The collection of retrieved pages is ranked according to the deviced ranking mechanism.

A decision is made, for which pages to be retrieved next is selected upon.

The crawling ends at this point if no pages are selected.

Figure 6-2. Focused crawling algorithm

6.4 Analysis of link structures

In the previous chapter an application utilizing simple link analysis was introduced in connection with the experiment on syndication in the online publishing industry. Further fine-grained techniques, to be used in the focused crawler algorithm, are discussed in the next section.

6.4.2 Finding significant hubs and authorities

If a significant hub is defined as one pointing to many authorities such hubs could be found by utilizing the extended link analysis table introduced in chapter 5. By querying the mentioned table it is possible to produce a list of hubs ranked according to significance. Similarly, a list of authorities may be produced, by using inverse link frequency, which is also embodied in the extended link analysis table. The inverse link frequency is found by counting external links pointing towards a particular page. Significant authorities would be those with many links directed towards them. Both hubs and authorities may be considered in terms of one single page, a group of pages or a whole site for that matter.

6.4.3 Finding significant hub segments

While carrying out basic link analysis, normally hubs are considered on a per page basis, however by considering hub-segments, or micro-hubs if you like, more fine-grained link-analysis could be undertaken. The Testbed has by default the ability to do basic segmentation of HTML documents. This makes it possible to consider each document at a micro-level, in terms of segments as oppose to the macroscopic level denoted in the previous section. Utilizing the extended link analysis table introduced in chapter 5 makes it is possible to query the Testbed for those segments pointing to authorities and rank the each segment according to how many authorities they point to.

One of the hypothesis raised in chapter 1 was that hyperlinks to similar pages are placed near each other in a document. If hyperlinks are spacial close to each other they would also be captured in the same segment. If each segment is considered separately a crawler may decide not to follow links in other segments. This is an important property, which is to be exploited in the experimental focused crawling algorithm to prevent topic drift caused by links pointing to disparate sources by pages discussing heterogeneous topics.

6.5 Analysis of HTML structures

Traditional information retrieval has been based solely on the indexing of textual content. It was not until the introduction of hypertext and hypermedia that link analysis was utilized to improve retrieval. Although methods such as link analysis may improve retrieval in terms of relevance and recall, the traditional methods should not be disregarded completely, as they appreciate important properties not taken into account by the other methods. Content and context is as such by far the most important properties available for the evaluation of relevance.

With the introduction of hypertext and hypermedia the markup languages followed and HTML today is the standard format web publishing used by all Internet browsers. By exploiting the structures inherently within web pages constructed using Hyper Text Markup Language (HTML) it is possible to weight content according to the enclosed structural elements. In other terms it is possible to value the importance of document headings and plaintext differently. At the moment seven classes of markup elements (tags) are used. These classes are the same as used by a study on using HTML structures to improve retrieval [41,42]. The classes are enclosed in table 6-1.

Class Name	**HTML tags**
Title	TITLE
Header	H1, H2, H3, H4, H5, H6
List	DL, OL, UL
Strong	STRONG, B, EM, I, U
Anchor*	A
Plain Text	None of the above
Meta*	META

Table 6-1. The classes and associated tags

In addition to the six classes used in [41,42], the content present in the document Meta keywords and Meta description were introduced as one further class called Meta*. Also note in particular that the anchor class of page *p* is meant to contain all the term occurrences that appear in anchor tags of hyperlinks in pages *q* that point to it, and as such it contains term occurrences from *other* documents. In addition to the text appearing in the anchor tags a window of text surrounding the hyperlinks of pages *q* is included in the Anchor* class. More accurately the segment that enclose the hyperlinks in pages pointing into *p* is used as the window and it thus a dynamical window. In other words a page inherits properties from pages in its neighbourhood. The reason for doing so is the belief that if text descriptive of a topic occurs in the text of pages pointing into *p* this should reflect on page *p*.. A similar windowing technique using a static window has been successfully deployed by [50]. Refer to [41,42] for more information on how and why the other HTML tags were grouped into classes.

The Testbed provide handles for all the classes by default and the term occurrences are stored in the Extracted table of the Testbed. A special table in the Testbed was constructed for the analysis of text. But one major obstacle needed to be forceed because the implemented database (MySQL) has no possibility for individual term occurrence weight assignment. This was solved in a rather ad-hoc manner by repeating each occurrence term, weight times, which consequently restrict term weights to discrete values. The ad-hoc solution was chosen because of it simplicity, but most of all because it could be implemented very fast which was an important considering the time constraints bound by the project.

To be indexed a weighted representation of every document was constructed using the repeated term method, but before indexing the documents some pre-processing was required. All Key Intelligence Topics (KITs) from the Topic Map model constructed by more than one word, such as a phrase, had to be merged, in order to be searchable after indexing. The merging is done by replacing all non alpha-numeric characters in a phrase with a character not counted as a tokenizer by the indexer (the underscore character). This is necessary because otherwise the tokenization process performed by an index-engine would split up phrases before indexing and they would thus not be searchable. Note that all these constraints are mainly due to the adopted ad-hoc solution and would have been avoided if another more appropriate solution had been used.

The table used for the analysis of text were loaded with the weighted representation of each web page. The table is indexed using the full-text indexing feature shipped with version MySQL 3.23 and above. MySQL full-text index implements the vector space model with no stemming of words. Provided a list of Key Intelligence Topics it is now possible to produce a ranked list of all documents in a collection using the text analysis table with those documents having a large concentration of KITs being ranked higher than those with less. Relevance is computed by MySQL based on the number of occurences in a documet, the number of unique occurences in that document, the total number of occurences in the collection, and the number of documents that contain a particular occurrence.

6.5.1 Weigth assignment

The weight assigned to each class are comprised by a Class Importance Vector CIV = [civ_1, civ_2, civ_3, civ_4, civ_5, civ_6] where civ_i is the importance factor assigned to class i. The CIV used for the experiment is [1 8 1 8 8 2], which is one of the best CIVs found by [41,42] using a genetic algorithm. The importance factor for the Meta* class was set to $\mathrm{civ}_7=3$ and assigned part by rationale and part by trial and error. The extended CIV [1 8 1 8 8 2 3] includes the Meta* class. Conveniently all the importance factors are discrete values. For more information on how the weights were assigned refer to [41,42]. Note also that the introduction of an extra class and the modification of another is not as uncomplicated as it may seem, and it should be looked into more carefully how the Meta* class and extended Anchor* class affect the importance factor of the other classes.

6.5.2 URL ordering

The focused crawler implements a ranking mechanism that associates a score with each link in the pages it has retrieved. This mechanism is used to ensure that the crawler preferentially pursues promising crawl paths.

By combining analysis of link structures and analysis of text, a simple mechanism for assigning scores was built. This type of ranking mechanisms is sometimes referred to as URL ordering [69].

The current algorithm used accounts for the concentration of KITs, micro-hubs and the global inverse link frequency (authority). The global inverse link frequency has low semantic value compared to the other two measures, but may indicate to a certain extend the degree of authority for a page. The low semantic value is due to the fact that common sites, such as search-engines and large companies, is often linked to by many pages, and more often than not this is with no regards to the topics discussed in these pages. Still, the inverse link frequency may be used to adjust for those pages not properly intercepted by the other two methods.

The score for link *l* is calculated using the following measures:

1. The number of authorities co-cited with *l* in a micro-hub *m*.
2. The number of micro-hubs citing *l*
3. The concentration of KITs in the page holding *m*
4. The inverse link frequency of *l*

Note in particular that the concentration of KITs in a page *p* is also influenced by the pages citing *p*. This is due to the windowing technique explained previously in this chapter. This means in practice that each link inherits properties, in this case the KIT concentration, not only from the pages that cite it, but also from the pages that cite the page that cite it.

An experimental rank algorithm

m is a micro-hub pointing to at least one authority.
s_l is the score for link l
m_{cited} is the number of authorities cited by micro-hub m
m_{kit} is the concentration of KITs in the page holding m
l_{gilf} is the global inverse link frequency of l
c_1 is the importance factor for co-citation
c_2 is the importance factor for concentration of KITs
c_3 is the importance factor for the inverse link frequency

Accumulating the weights from the micro-hubs

```
For each $m$
    For each link $l$ in $m$:
        $s_l = s_l + (m_{cited}*c_1 + m_{kit}*c_2)$
```

Adjusting for global inverse link frequency*

```
For each $sl$
    $s_l = s_l + l_{gilf}*c_3$
```

**This last adjustment could potentially increase scores for pages not always co-cited with an authority.*

In order to combine all three measures in particular the inverse link frequency needed to be normalised. The micro-hub citation measure (1) takes discrete values ranging from one up to a discrete value counting the number of authority pages it links to. The concentration of KITs (2) takes values, ranging from zero meaning no occurrences up to a non-negative floating-point number. The global link frequency (3) is also a discrete value measure, but the value depends on the number of pages in the collection analysed. As the number of known pages increases the inverse link frequency count is bound to increase proportionally. To compensate for this dependency we use a multiplying factor c_3 related to the number of pages in the collection. (1) and (2) are mutually reinforcing [4] and as long as the number of authority pages in the collection analysed is fixed, these measures could be combined with no further normalisation required.

With the use of trial and error an importance factor was also assigned to each of the three measures. (2) was given the highest importance, primarily because content is after all what we seek in relevant pages. (1) was assigned the next highest importance and (3) the lowest importance. At the moment the importance factors used are fixed to [8 2 1].

In order to improve ranking, the calculation of the combined measure needs to be revised using more scientific methods beyond the trial and error approach.

6.5.3 Crawling strategy

Several strategies were investigated during the experiments. The first strategy was to start with the initially provided pages and assign scores immediately after retrieving those pages. In addition to the initially provided pages all the corresponding root pages of the provided pages where retrieved if they had not been retrieved previously. This means that if e.g. **http://www.universimmedia.com/topicmaps/guide.htm** is selected the root page **http://www.universimmedia.com/** is automatically retrieved by the crawler. This is because often the root page provide more descriptive information about the pages on that site which could be used later on when evaluation the results.

Because all the experiments started with a relatively small set of initial pages it was necessary to revise the first strategy by expanding the number of pages retrieved before assigning scores. This was done by downloading all the pages linked to by the initially pages. This means that each iteration has two internal steps. Step one is to retrieve all the selected pages and step two is to expand the selected set of pages by downloading all the pages pointed to by the selected pages.

6.5.4 Selecting pages to be retrieved next

The most important part of any focused crawler is selecting which pages to retrieve. This is after all what separates a focused crawler from a standard crawling algorithm. To decide which page to retrieve next one solution is to use a threshold based on the score assigned by the rank algorithm. All pages above the threshold are selected and the number of pages selected will thus vary. Using a threshold based exclusively on the assigned score will result in virtually no control with the number of pages retrieved. One practical solution to this problem is to set a maximum number of pages to be selected. The threshold could also be fixed statically such that only a certain number of pages above the threshold are selected, but this would prevent the algorithm from ever converging. By convergence we mean that the number of pages that needs to be retrieved naturally decrease with each iteration. This happens because fewer and fewer pages meet the requirement of having score above the threshold.

Note in particular that using this threshold method, although a page does not meet the requirements for being pursued by the crawler in one iteration, this does not necessarily mean that it will not be considered in the next iteration. In practise this means that if the pages retrieved next have many hyperlinks to pages not pursued in the previous iteration, they may be pursued in the next iteration because they are assigned a higher score.

6.5.5. Setting the threshold

Setting the initial threshold could be a difficult exercise and at the moment this is done manually by investigating the ranked list produced by the implemented rank algorithm. Where to set the threshold depends on the number of pages that we want to retrieve in each iteration. From experience it looks most promising to set the threshold quite high such that only a few pages are retrieved in each iteration and rather have many iterations.

Automatically detection of thresholds is used in image processing to create bi-modal images from gray-scale images. The same algorithms as used in image processing for threshold detection might useful when developing methods for automatically setting the threshold value used in the focussed crawler.

6.5.6. Convergence

At some point we would like the crawler to stop. In some of the experiments conducted convergence occurred, but only for a few iterations. If the number of pages pursued does not decrease naturally the crawling may go on forever. This could be dealt with in two ways, manually and automatically. At the moment this is done manually by monitoring the crawler and adjusting the threshold after each iteration. The other way is to automatically force convergence. One form of “auto-focus” can be achieved without much effort by setting the threshold higher each time the number of pages selected exceeds the number of maximum pages set. Dynamically adjustment of the threshold would require more complex algorithms.

6.5.7 Results

When the focused crawler has finished, several types of results may be constructed by link analysis and text analysis of the retrieved pages in the same way as when assigning scores used for URL-ordering. First of all this includes ranked lists of all the page, but it could also be a list of all pages pointing towards a particular page or a ranked list of significant micro-hubs. These results however will not be used directly. Instead they will be utilized to construct navigable Topic Maps using the tools explained in the chapter to follow.

Chapter 7 Semi-automated Web Resource Discovery Application prototype

The architecture of the proposed approach is presented in chapter 4 and this chapter will thus primarily be concerned with implementation issues. In particular we discuss how to deploy the experimental focused crawler and miscellaneous Topic Map processing tools. The last part of the chapter is devoted to test cases and the results originating from the belonging experiments. First a few tools for Topic Map processing are presented.

7.1 Topic Map processing

A few simple tools were built in order to facilitate basic Topic Map processing. All the tools were developed using tmproc [43], which is a freely available Topic Map processor, implemented in Python, made by Geir Ove Grønmo. For the visualisation of Topic Maps a graph-rendering tool called GraphViz from AT&T Research is used, which is released under an Open Source license.

7.1.1 Extracting occurrences and Key Intelligence Topics

The first tool made using tmproc was used to extract occurrences and Key Intelligence Topics from source Topic Maps. Because the Focused Crawler do not read Topic Maps directly this tool is used primarily as a conversion tool to transform the source Topic Map into something the Focused Crawler can understand. The Focused Crawler requires two lists as input, one for the occurrences and one for the KITs. The KITs were extracted from the Topic Map using the *get_topics_of_type(tm[“Topic”])* construction provided by *tmproc.* In addition to extracting all occurrences of all topics, all Identities if available were also extracted. Thus the input to the tool is a Source Topic Map and the output is a list of URLs and a list of KITs ready to be passed on to the Focused Crawler.

7.1.2 Topic Map construction

Topic Maps may be authored manually by people or constructed automatically by computers. For a Topic Map to be constructed automatically some human effort is always required and as such it is probably more accurate to call the tools developed for Topic Map generation tools to avoid confusion.

7.1.3 Result Topic Map generator

The second tool developed was a small Python program used to generate a Topic Map from resources found by the focused crawler. Note that this tool was developed for use in the experiments specifically and considering what a Result Topic Map generator ought to be it is far from complete.

The result Topic Map is generated in two steps. First a Topic Map is generated using the using the Linear Topic Map notation. The LTM Topic Map is then transformed into an SGML Topic Map using the Linear Topic Map processor [44] made by Lars-Marius Garshol.

Merely a subset of the Focused Crawler results is modelled in the Result Topic Map.
At the moment one Topic Map is generated containing all sites found by the Focused Crawler pointing towards an authority site. An authority site would in this case be the same as a page present in the Source Topic map. In addition a subset of the Result Topic Map is generated for each resource discovered by the Focused Crawler. All sites are modelled as Topics and hyperlinks are modelled as associations.

7.1.4 Merging Topic Map

As explained in chapter 3, there are two constructs defined in the Topic Map standard that avails for the merging of Topic Maps. The methods are either explicitly through Identity attributes or implicitly through topic naming constraints. For the purpose of merging Result Topic Maps and Source Topic Maps in the application prototype a simple tool using the explicit method was developed. The Merging is done simply by comparing the URL of the discovered resources in the Result Topic map with the Topic identity attributes in the Source Topic Map. An approximate matching is used, allowing the Identity URLs to be considered equivalent as long as they are within the same domain name.

7.2 Topic Map visualisation

Visualisation of Topic Maps may be achieved using both 2D and 3D views. 3D visualisation allows a more efficient use of screen space, in particular because links between nodes do not intersect. In this project, however, only 2D visualisation techniques have been investigated.

7.2.1 GraphViz

Topic Map graph visualisation is implemented using a tool called GraphViz. GraphViz provides powerful tools for drawing directed and undirected graphs. The input to this tool is a description of the graph in the *dot* language and the output is a rendering of the graph in a choice of vector or bitmap graphics formats. Since a Topic Map graph is expressed using nodes and arcs GraphViz was an obvious choice for visualisation. GraphViz is also utilized in a couple of initiative on the visualisation of Resource Description Framework (RDF) graphs [45,46].

A powerful feature of GraphViz is its ability to render output in the Scalable Vector Graphic (SVG) format. SVG [47] is a language for describing two-dimensional graphics in XML. As of this writing, the status of the SVG standard is a World Wide Web Consortium (W3C) [48] Candidate Recommendation. At the moment a freely available plug-in downloadable from Adobe [49], is required in order to view SVG format files in a browser. However the avail of inline SVG browser support is expected to change as soon as the format moves into the Recommendation phase at W3C.

A few Python tools have been developed to construct scripting files in the dot language to be used with GraphViz. This is achieved by extracting Topics and Associations from a Topic Map. The extraction is as usual automatically done using tmproc allowing the Topics and relations to be easily translated into nodes and labels using the dot scripting language read by GraphViz. The dot language is as can be seen from the following examples very easy to use.

digraph test {
A [shape=box, URL=”http://www.hia.no/”]
B [shape=circle, Url=”http://www.grimstad.kommune.no/”
A -> B [label=”located in” url=”/tmengine/resolve?a+b]
}
	graph test {
	A [shape=box, URL=”http://www.hia.no/”]
	B [shape=circle, Url=”http://www.grimstad.kommune.no/”
	A -- B [label=”located in” url=”/tmengine/resolve?a+b]
	}

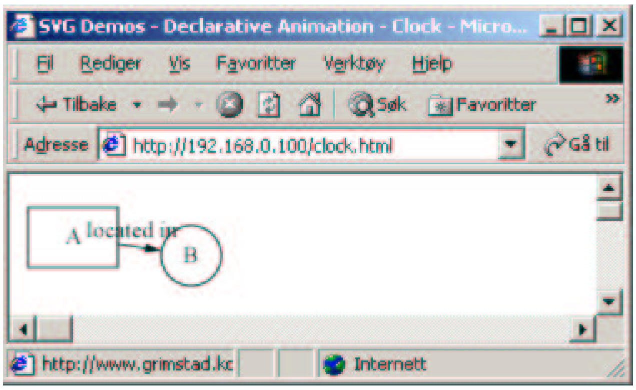
Directed graph dot language example syntax.	Undirected graph dot language syntax example

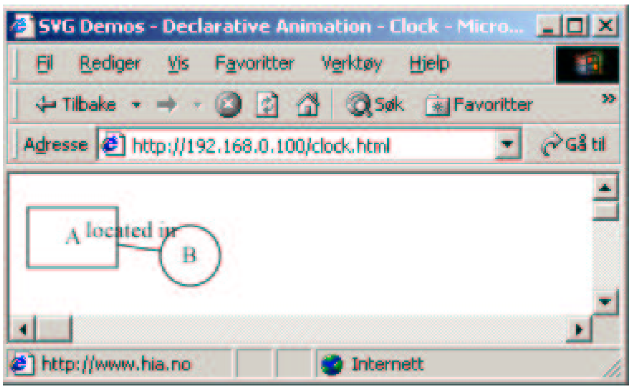
SVG output

```
<svg width="118pt" height="49pt" >
<a xlink:href="http://www.hia.no">
<polygon  style="fill:none;stroke:black" points="55,37
1,37 1,73 55,73 55,37 "/>
<text x="23" y="59" style="font-family:Times;font-
size:14.00" >A</text>
</a>
<a xlink:href="http://www.grimstad.kommune.no/">
<ellipse  cx="99" cy="66" rx="18" ry="18"
style="fill:none;stroke:black" />
<text x="94" y="70" style="font-family:Times;font-
size:14.00" >B</text>
</a>
<a xlink:href="/navigator/resolve?A+B">
<g  style="fill:none;stroke:black"><path d="M55 59C62
60 68 61 75 62"/></g>
<polygon  style="fill:black" points="71,64 81,63 72,59
71,64 "/>
<text x="36" y="55" style="font-family:Times;font-
size:14.00" >located in</text>
</a>
</svg>
```

```
<svg width="118pt" height="49pt" >
<a xlink:href="http://www.hia.no">
<polygon  style="fill:none;stroke:black" points="55,37 1,37
1,73 55,73 55,37 "/>
<text x="23" y="59" style="font-family:Times;font-
size:14.00" >A</text>
</a>
<a xlink:href="http://www.grimstad.kommune.no/">
<ellipse  cx="99" cy="66" rx="18" ry="18"
style="fill:none;stroke:black" />
<text x="94" y="70" style="font-family:Times;font-
size:14.00" >B</text>
</a>
<a xlink:href="/navigator/resolve?A+B">
<g  style="fill:none;stroke:black"><path d="M55 59C64 61
73 62 81 63"/></g>
<text x="36" y="55" style="font-family:Times;font-
size:14.00" >located in</text>
</a>
</svg>
```

Visualisation in browser





The output from GraphViz is in SVG format allowing the graphs to be viewed in a browser. Note in particular how the objects are used as hyperlinks by providing a URL label.

7.3 Test case

A few test cases have been conducted and the first test-case experiment was mapping up the Competitive Intelligence and Business intelligence industry Business Landscape. It was a rather successful experiment but it is not included in this report. A test case on the Topic Map community is documented in this report.

7.3.1 The Topic Map community

This test-case experiment aimed to map up the Topic Map community Business Landscape. The URL of a few of the major players an organisation and a few news-pages were provided as input together with 4 Key Intelligence Topics. There are two sections connected with this experiment; first the results from the Focused Crawling are presented and next the visualisation of the Merged Topic Map is presented.

Focused Crawler results

The following 16 URLs and 4 KITs were extracted from the Source Topic Map provided in chapter 3.

URLs
http://www.mondeca.com/site/news_events/press_releases.html
http://www.topicmaps.com/
http://www.mondeca.com/
http://www.infoloom.com/
http://www.mondeca.com/
http://www.knowledgetechnologies.net/
http://www.topicmap.net/
http://www.topicmaps.org/
http://www.ontopia.net/
http://www.empolis.com/englisch/presse/index.html
http://www.empolis.co.uk/
http://www.empolis.com/
http://xml.coverpages.org/topicMaps.html
http://www.empolis.de/
http://www.topicmap.com/
http://k42.empolis.co.uk/

Table 7-1

KITs
xtm
13250
Sematic_web
Topic Map(s)

Table 7-2

Explanation of the table used for the crawling experiment

Row 1	Number of retrieved pages in this iteration. The number in parenthesis is the total number of pages retrieved so far.
Row 2	Number of selected pages in this iteration. The number in parenthesis is the total number of candidates to be chosen from.
Row 3	The total number of URLs the crawler has seen so far.
Row 4	The number of unique sites found within the total number of URLs.
Iteration	In each iteration there are two internal steps. In the first step (column 1) the selected pages are retrieved which is then expanded in step 2 (columns 2).

Table 7.3.1.3

	Init	Iteration 1		Iteration 2		Iteration 3		Iteration 4	
Retrieved URLs	0	22 (22)	362[b] (384)	20 (404)	537 (941)	7 (948)	287 (1235)	4 (1239)	43 (1278)
Selected URLs	16*	338[a]	17 (468) [1]	486	7 (526) [2]	255	3 (521) [3]	37	-
Known URLs	16	396	8622	9120	16655	16920	22369	22409	23945
Known sites	13	63	1078	1088	1424	1441	1647	1657	1679

Table 7-3

*The 16 pre-selected URLs

a,b The difference between a and b is a result of the inner loop in the Testbed retriever which also fetches framesets and redirected pages.

The following tables are lists of the pages that were selected in each iteration. Having some domain specific knowledge of the Topic Map community the selected URLs seems like an appropriate choice. It is however very important to understand that the pages selected in each iteration will contribute to the final results equally to those many more pages selected automatically to be retrieved by the Crawler in the expansion phase. The automatically selected pages will show up directly in the final results.

1

#	URL	#	URL
1	http://www.datachannel.com/	10	http://www.coolheads.com/
2	http://architag.com/newsletter/	11	http://www.quid.fr/
3	http://www.hytime.org/	12	http://www.iso.ch/
4	http://www.phylis.com/	13	http://www.hytime.org/topicmaps/index.html
5	http://www.diffuse.org/TopicMaps/schema.html	14	http://www.universimmedia.com/topicmaps/guide.htm
6	http://www.xmledi.net/	15	http://www.infotek.no/
7	http://www.hytime.org/topicmaps/playsmap/	16	**http://www.lcc.gatech.edu/**
8	http://www.infotek.no/~grove/software/tmproc/index.html	17	**http://architag.com**
9	http://www.lcc.gatech.edu/gallery/rhetoric/terms/topoi.html		

Table 7-4 – Iteration 1

2

#	URL	#	URL
1	http://www.gnu.org/copyleft/gpl.html	5	http://sourceforge.net/projects/pyxml
2	http://sourceforge.net/mail/?group_id=10193	6	http://wxpython.org/
3	http://wxpython.org/download.php	7	http://sourceforge.net/
4	http://www.python.org/1.5/	8	

Table 7-5 – Iteration 2

3

#	URL	#	URL
1	http://www.xmleurope.com/	3	http://www.mcs.net/
2	http://www.mcs.net/~dken/esgml97.htm	4	

Table 7-6 – Iteration 3

Compared to a public search engine the Focused Crawler allows us to produce results in a number of different formats. In this particular case 1278 pages were retrieved and they could be ordered and clustered in a number of different ways providing different views on the results. A small program was made to illustrate how the results could be ordered and grouped according to their domain name. The score was assigned to each domain in a similar fashion as in the experimental ranking algorithm. The results considered are all pages retrieved excluding those provided as input to the crawler and only those sites with at least one hyperlink to one of the initially provided sites.

The table below is an ordered list of the top 30 unique domains as ranked by the experimental ranking algorithm.

#	URL	#	URL
1	Doctypes.org	16	yahoogroups.com
2	w3.org	17	architag.com
3	egroups.com	18	oasis-open.org
4	universimmedia.com	19	Seyboldreport.com
5	xml.com	20	infotek.no
6	hytime.org	21	computer.org
7	empolis.co.uk	22	garshol.priv.no
8	bonn.iz-soz.de	23	semantext.com
9	Techquila.com	24	idealliance.org
10	diffuse.org	25	ornl.gov
11	sgml.u-net.com	26	zdnet.com
12	sourceforge.net	27	xmledi.com
13	cogx.com	28	y12.doe.gov
14	Coolheads.com	29	isogen.com
15	datachannel.com	30	techno.com

Table 7-7

Note that it is evident that the algorithms made in order to rank the final results need to be revised in a similar way as the URL ordering algorithm. The results are provided as is, because there was unfortunately no time left in the project to perform recall or precision measures.

Merged Topic Map visualisation

In order to visualize and navigate the results, the Topic Map processing tools devices are used. First the Result Topic Maps are constructed, next the Result Topic Map is merged with the Source Topic Map and then finally the Merged Topic Map graph is constructed and visualised in a browser. At the moment only two colours are used in the graphs. Red is used to indicate discovered resources and relations. Black is to indicate known resources and relations. The green notation mentioned in chapter 4 is not used. Even though Topic Map associations are undirected the links are directed in the visualised graphs. Although meaningless from the Topic Map point of view, an understanding of who link to whom is an important fact to be aware of while resolving ambiguities. However as soon as they are resolved the arrows should strictly speaking disappear.

Because the graphs are supposed to be navigated using a browser they are rather big and required to be printed on individual pages. Explanation of the figures to follow:

Figure 1
This figure shows a Merged Topic map of a Result Topic map containing all the discovered resources that has at least one link pointing to one of the sites in the initial Source Topic Map (the Business Landscape). As can be seen this is a fairly complex graph, which is hard to get an overview of. The next figures try to solve this by just looking at a subset of the results.

Figure 2
This figure shows a Merged Topic map of a Result Topic map containing all links to and from one the resource discovered. The resource in question is “Semantext” as showing up in row 23 in table 7-7

.

Figure 3
This figure shows in the same manner as figure two a subset of the results. The resource in question in this figure is “Doctypes.org”, ranked as #1 in table 7-7. From the figure we can se that this resource has a lot of external links to other players in the Topic Map community business landscape and is as such considered an important resource by the system.

Chapter 8 Discussion

This chapter is supposed to provide comments on and propose alternative approaches that could have been pursued as oppose to those directions chosen along the way in this projects. Topics discussed in particular are alternative approaches to Resource Discovery, Topic Map graph visualisation and processing. Some ideas on automatically resolving ambiguities and some final thoughts on the proposed architecture model are also included.

8.1 Alternative approaches to Resource Discovery

The Focused Crawler yield best results if some form of hub-pages are provided as part of the initial set of URLs. This is because if only URLs detached from the web-graph are used as input, the Focused Crawler will not find enough interesting external hyperlinks to follow. By detached we mean that they do not have hyperlinks to external resources. To reach the unconnected parts of the web graph, methods must be adapted to provide additional starting points for the Focused Crawler. This has been solved in a number of other projects by utilizing the public web search-engines to expand the initial set of URLs [10,50]. This method was not pursued in this particular project, but instead alternative methods were investigated.

In the early stages of this project the possibilities of using the Open Directory Project (DMOZ) [51] in the Resource Discovery application was investigated. The goal of the Open Directory Project is to produce the most comprehensive directory of the web, by relying on a vast army of volunteer editors.

The basic idea pursued was that the initial provided URLs should act as pointers into the DMOZ database. For each URL the DMOZ database was to be consulted to se if they had been categories, and if this was true all other URLs in that corresponding category should be added to the initial set of URLs. This expanded set of URLs was supposed to be used as input to the Focused Crawler.

A fair amount of time was spent in order to parse the ODP data, which is described using Resource Description Framework (RDF) [52]. The ODP data was to be stored in two tables in the MySQL database, one for resources and one for categories. However due to some recent changes in the underlying data-format the available ODP parsers at that time did not parse the data correctly. Using a generic RDF parsers instead was the obvious alternative, however in order to come up with some results as quickly as possible a proprietary ODP dump parser was built and the data were stored using the mentioned tables in MySQL.

All well so far, but sadly the quality of the ODP directory is not what it was expected to be. This resulted adding irrelevant URLs to the initial set of URLs and the attempt of exploiting the ODP directory for Resource Discovery was thus terminated. The low quality may perhaps not be as crucial to a human but to the Focused Crawler irrelevant URLs as input would result in reduced quality of the results. The perceived low quality of the ODP directory is also backed up by a study on indexing consistency [53]. This study found that indexers assign the same term to the same document only 50% of the time. While indexers may agree on broad subjects area, they differ on which terms to assign to a specific document.

8.2 Supplementary repositories for use in Resource Discovery

A recent whitepaper [54] prepared by a working group on resource discovery asserted that there are potentially one million repositories on the Web today. This include tens of thousands of special purpose local search-engines that focus on documents in confined domains such as documents in an organisation or of a specific subject area.

A scalable alternative to Resource Discovery is the metasearch engine approach. A metasearch engine can be considered an interface on top of multiple local search engines to provide uniform access to many local search engines. The challenge lays in the ability to efficiently and accurately determine a small number of potentially useful local search engines to invoke for each user query [55].

Another approach to Resource Discovery is extraction of Web resource recommendation from repositories such as discussion groups (Usenet), news articles, e-mails and Internet Relay Chat (IRC). Phoaks [56] a system for sharing recommendations shows that 23% of Usenet messages mention web resources ad 30% of these are recommendations. In the Phoaks project fairly robust methods were developed to discriminate between different types of recommendations such as ignoring a URL that is part of a posters signature.

All of the above methods are likely contribute in Resource Discovery implementations, however they should be used with care as they may suffer from some of the same inconsistencies as the ODP dumps.

8.3 Topic Maps vs. RDF

The Resource Description Framework (RDF) specification is a W3C initiative to provide a lightweight ontology system to support the exchange of knowledge on the Web [48]. RDF is similar to Topic Maps in that they both attempt to alleviate the same problem of findability in the age of infoglut, and they are doing so by annotating information resources. One of the key differences is that Topic Maps take a topic-centric view whereas RDF takes a resource centric-view. Topic Maps start from topics and model a semantic network layer above the information resources. RDF starts from resources and annotate them directly and only by stretching the model beyond its real intent, the abstract layer can be represented [57]. An in depth comparison of the two technologies is provided in the paper “Topic Maps vs. RDF” by Eric Freese [58].

It was primarily the graspable and appealing Topic centric view and the ability to model knowledge without any reference to the underlying resources that guided the decision of using Topic Maps in this project. However the availability of a powerful and easy to use and Topic Map processor implementation in Python did also affect the decision.

More recently the concrete relation between RDF and Topic Maps is under investigation. Both the RDF and Topic Map comities are working together to develop a harmonized solution which could be one of the base layers for the Semantic Web [59].

8.4 The Semantic Web

The Semantic Web [61] is not a separate web but an extension of the current one. The essential property of the World Wide Web is its universality but to date, the Web has developed most rapidly as a medium for people rather for data and information to be processed automatically. The Semantic Web aims to make up for this.

Both RDF and Topic Maps are runner-ups to power The Semantic Web however there is a reasonable doubt that the envisioned "The Semantic Web" will revolutionise the Web any time soon. With this time perspective in mind, an in depth discussion of The Semantic Web is out of the scope of this report.

8.5 Final thoughts on the Resource Discovery architecture model

If we think about it the proposed model resembles quite well what people do while performing manually Resource Discovery. The difference is that it is done in a much more structured manner. Equally important is the discovered knowledge represented by the Business Landscape, which is readily available to be navigated and shared with anyone and thus enables a shared corporate memory.

What it mission critical is time spend in the “human-learning” process because everything besides this is performed automatically. To reduce time spent by a human operator it is important that the application prepare as much semantically valuable information as possible in order to resolve Topic and Association types as quickly as possible. This can be done using the mentioned sophisticated Topic extraction techniques.

8.5.1 Automatically resolving ambiguities

As more advanced Topic extraction tools are adapted in the Result Topic map construction more ambiguities will occur and demands for more automation of the ambiguities resolving process. Most of the ambiguities that are deemed to occur will most likely be that of concepts easily grasped by a human. If the computer had the ability to capture at least some of the obvious concepts it would arrange for the machine to automatically resolving of ambiguities.

The Cyc knowledge Base [61] is a publicly available computer based ontology that claims to capture “the most general concepts of human consensus reality”. A working draft of a technical report from Sun Microsystems [62], documents research and development of an XML Topic Map (XTM) representation of the Upper Cyc Ontology. On July 1, 2001 a greatly expanded version of the Cyc Common Sense Knowledge Base will also be made available in open access under the name OpenCyc.

In contrast to artificial intelligence (AI), the Cyc effort seems to focus on providing “real intelligence” to machines. This is very interesting and with the Cyc XTM initiative and the involvement of Sun Microsystems the possibilities of integrating Cyc with the Resource Discovery application in order to provide “real intelligence” is definitely an appealing though.

8.6 Topic Map visualisation and navigation

The area of Topic Map graph visualisation could have been an entire project on its own. The paper “Information Management – Topic Maps visualisation” [68] provides more in depth discussions on Topic Map visualisation.

The natural evolution of the plain graphs presented in this report would be the use of different colours, sizes, shapes and styles of nodes and arcs, inherently in GraphViz, in order to indicate significance, type, scope etc. Runner-ups include also 3D visualisation using VRML or the 3D initiative equivalent of SVG named Web3D.

Alternative graph rendering applications include Nicheworks [63], another tool from AT&T Research for exploring large networks providing zooming, focusing and filtering. The patented Star-Tree visualisation product from Inxight [24] use a focus+context technique based on hyperbolic geometry to visualize large hierarchies [64]. However, what may put constraints on alternative implementations is the fact that Topic Maps has a web structure and not a hierarchical tree structure.

When it comes to Topic Map navigation as mentioned nothing is implemented yet as far as this project is concerned. A Topic Map engine is required in order to interface with the Topic Map from the Web. Such an engine has most of the attributes of a Topic Map processor and as such tmproc could be used to create a Topic Map engine, however a Topic Map engine requires additionally handling of persistent objects and the ability to dynamically load large Topic Maps. A number of Topic Map engine implementations are available and most recently an extensible Topic Map engine framework was released by Ontopia [65], a Norwegian Topic Map company. This framework is a strong candidate to provide the required interactivity and due to the extensibility there should be no problem continuing using GraphViz to render the graphs.

Chapter 9 Future directions

The future directions for this project is in particular connected to further work on the proposed solution architecture model. This includes building a complete implementation of the model. To achieve this a Topic Map engine must be used for the provision of interactivity between the user and the underlying Topic Map. This interactivity is fundamental in that of resolving Topic and Association types.

As soon as we are able to back-propagate the gained knowledge represented by the Merged Topic Map, this will also allow us to find out how the Focused Crawler will benefit from being better informed prior to the crawling. Further work on the Focused Crawling algorithm needs in particular to focus on qualifying the assignment of scores. In order to improve ranking, the calculation of the combined measure needs to be revised using more scientific methods beyond the trial and error approach. Besides this, more appropriate methods for indexing with individual term occurrence weights is called for in the Testbed.

The Gartner Group [67] said of Topic Maps: “the paradigm is powerful, flexible and extensible, Topic Maps will become a mainstream technology by 2003.”

As the Topic Map standard receives more attention we will see more of applications implementing the standard. As the standard mature this will also be of considerably help in the future development of the application prototype.

Chapter 10 Conclusion

The way we look at information has changed and we have built a natural and deliberate way of looking at technology. We search for, use and relate to new knowledge and learning in a different light, and we utilize more and more information from digital medias. At present we are ready take the next step forward and the proposed solution initiated in this project aims to help us do so. This is equally important because current awareness in the continuously and rapid changing Business Landscape has become a “need to know” activity as oppose to “nice to know”.

The publicly available search-engines seem to primarily focus on quantitative measures such as an enormous coverage and fast response. Combined with the fact that they continuously lags several weeks and even months behind comparable to the current state of the ever-changing web, this calls for alternative methods for finding aids for the Web.

The greatest challenge in order to create such finding aids, lays in the ability to formalise knowledge and capture interests and information needs. From the experiences gained in this project, Topic Maps seem to have the necessary properties for this type of knowledge representation.

For the sake of the Focused Crawler, the experiments conducted so far shows promising results. Using as few as ten to fifteen Uniform Resource Locators (URL) supplied with Key Intelligence Topics, the algorithm manages to discover a number of new resources, comprised by players in the Business Landscape.

The proposed solution architecture model has yet to prove its usefulness in that of integrating human and machine learning. Still the current application prototype offer great help in order to bridge the gap between information discovered and knowledge. This is achieved by visualising the results discovered by the Focused crawler correlated with the Business Landscape.

References

1. R. G. Vedder, M. T Vanecek, C. S. Guynes and J. J Cappel. CEO and CIO Perspectived on Competitive Intelligence, Communications of the ACM, 1999

2. P. Lyman, H. Varian, J. Dunn, A. Strygin, and K. Swearingen. How much information? School of Information Management and Systems, Univ. of California at Berkeley, 2000.

3. Marc Najork, Janet L. Wiener. Breadth-first search crawling yields high-quality pages. Tenth World Wide Web Conference, 2001

4. John M. Kleinberg. Authoritive sources in a hyperlinked environment. In Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

5. Fuld & Company Inc. Intelligence Software: Reality or Still Virtual Reality? Intelligence Software Report 2000.

6. Fuld & Company. http://www.fuld.com/

7. Lagus, K., Kaski, S., Honkela, T and Kohonen, T. Browsing digital libraries with the aid of self-organizing maps. In Proceedings of the Fifth International World Wide Web Conference, 1996.

8. IBM Almaden Research Center. http://www.almaden.ibm.com/

9. Department of Computer Science, University of California, Berkeley. http://www.cs.berkeley.edu/~soumen/focus/

10. The Clever Project. http://www.almaden.ibm.com/cs/k53/clever.html

11. The Tenth Worl Wide Web Conference, Hong-Kong. 2001. http://ww10.org

12. Soumen Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. 10th International World Wide Web Conference, Hong Kong, May 2001.

13. SemanText. http://www.semantext.com/

14. Eric Freese. Using Topic Maps for the representation, management & discovery of knowledge. XML Europe 2000.

15. Michel Biezunski, Steven R. Newcomb. XML Topic Maps: Finding Aids for
the Web, Steven. April-June edition of IEEE Multimedia, 2001

16. Steve Pepper. The TAO of Topic Maps. XML Europe 2000, May 2000.

17. International Organization for Standardization. ISO/IEC 13250:1999 Document description and processing languages - Topic Maps. Geneva, 1999.

18. Rafal Ksiezyk. The answer is just a question [of Topic Maps matching]. XML Europe 2000, May 2000

19. Rath, H.H.: Making Topic Maps more colourful, in: Proceedings of XML Europe 2000 Conference, GCA, Alexandria, VA, 2000.

20. Lars Marius Garshol. The Linear Topic Map Notation. Definition and introduction, version 1.0. 2000-10-18

21. IEEE SPECTRUM. 2001

22. IBM Feature Extraction Tool. http://www-4.ibm.com/software/data/iminer/fortext/extract/extract.html

23. WhizBang! Labs. http://www.whizbang.com/

24. Inxight Thing Finder. http://www.inxight.com/

25. C. J. Godby, R.R. Reighart. The WordSmith Indexing System. 1998

26. Chimezie Thomas-Ogbuji. Transforming Python performance data. February 2001. http://www-106.ibm.com/developerworks/library/x-transpy/?dwzone=xml

27. The Python programming language. http://www.python.org

28. MySQL. http://www.mysql.com

29. Python DB API 2.0. http://www.python.org/topics/database/DatabaseAPI-2.0.html

30. Armin S. A Rehrl, Martin Frey, R. Alexander Rehrl. World Wide Web Robot for Extreme Datamining with Swiss-Tx Supercomputers. Interim Report, June 1999.

31. Ninth Python Conference. http://www.python9.org

32. Robot Exclusion Protocol. http://www.robotstxt.org

33. Zlib. http://www.info-zip.org/pub/infozip/zlib/

34. The Reiser Filesystem. http://www.namesys.com

35. Text REtrieval Conference. http://trec.nist.gov/

36. GraphViz. http://www.graphviz.org

37. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J. Wiener. Graph structure in the web: experiments and models. WWW9. 2000

38. Ninth World Wide Web Conference. 2000. http://www9.org

39. E. Spertus. Parasite. Mining structural information on the web. Computer Networks and ISDN Systems. 1997.

40. M. Diligenti, F. Coetzee, S. Lawrence, C.L. Giles, M. Gori. Focused Crawling Using Context Graphs. 26th International Conference on Very Large Databases, VLDB 2000

41.M. Cutler, Y. Shih, S. W. Meng. Using the Structure of HTML Documents to Improve Retrieval.

42.M. Cutler, H.Deng, S. S. Maniccam, and W. Meng. A New Study on Using the HTML Structures to Improve Retrieval. 1999

43. tmproc: A Topic Map engine. Geir Ove Grønmo. **http://www.ontopia.net/software/tmproc/**

44. LTM Processor. Lars-Marius Garshol. **http://www.ontopia.net/download/freebies.html**

45. The FRODO RDFSViz Tool. http://www.dfki.uni-kl.de/frodo/RDFSViz/

46. Rudolf: RDFViz. Exploring tools for RDF Graph Visualisation. http://www.ilrt.bris.ac.uk/discovery/rdf-dev/rudolf/rdfviz/

47. Scalable Vector Graphics (SVG). **http://www.w3.org/Graphics/SVG/Overview.htm8**

48. World Wide Web Consortium. **http://www.w3c.org**

49. Adobe Systems Incorporated. http://www.adobe.com/svg/

50. S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson and J. Kleinberg. Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. Proceedings of the Seventh International World Wide Web Conference (WWW7). Brisbane, Australia. April 1998.

51. The Open Directory Project. **http://www.dmoz.org**

52. Resource Description Framework (RDF). **http://www.w3.org/RDF/**

53. Susan Feldman "The Answer Machine.". Searcher, 8(1), January 2000: 58-78.

54. Resource Discovery in a Globally-Distributed Digital Library. Working Group Report, 1999 (http://www.iei.pi.cnr.it/ DELOS/ NSF/ resourcediscovery.htm)

55. W. Meng, C. Yu and Z. Wu. Towards a Highly-Scalable Metasearch Engine. WWW10. 2001

56. Terveen, L. Hill, W. Amento, B. McDonald, D. PHOAKS: A system for sharing Recommendations. Communications of the ACM. 1997.

57. Steve Pepper. Topic maps and RDF: A first cut. June 2000

58.Eric Freese. Topic Maps vs. RDF. Extreme Markup Languages 2000

59. Hans Holger Rath. Semantic Resource Exploitation with Topic Maps. GLDV2001

60. The Semantic Web. **http://www.semanticweb.net**, **http://www.w3.org/2001/sw/**

61. Cycorp. **http://www.cycorp.com**

62. The Upper Cyc Ontology in XTM. Sun Microsystems Technical Report 28 Feb 2001.

63. NicheWorks: Exploring Large Networks. http://www.bell-labs.com/user/gwills/NICHEguide/niche.html

64. J. Lamping, R. Rao, and P. Pirolli, A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. Proceedings of ACM SIGCHI'95, Denver, CO, 1995.

65. Ontopia. **http://www.ontopia.net**

66. Google. http://www.google.com

67. Gartner Group. **http://www.gartner.com**

68. Benedicte Le Grand and Michel Soto. Information management – Topic Maps visualisation.

69. S. Chakrabarti, M. van den Berg and B. Dom. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. WWW8.

APPENDIX