Evolving Association Index

Sacha Berger

December 14, 2001

Abstract

In this report indexing data using typed links instead of the commonly used keywords is examined. The purpose of indexing by linking is to group data items that are related as well as to discover documents of special interest without the need of providing search criterias (such as indexing categories or keywords) for those structures. What these structures are, is determined through the structure of the link graph. Therefore topics can be redefined as syntactic elements in graphs and can therefore be computed. An *Evolving Association Index* consists of data structure definitions for index authors and algorithms. Algorithms provide topic structures that can be used for categorizing and browsing data. In order to provide concrete examples and implementations, the approach is specialized to apply to online news publishing.

Contents

1 Introduction and Motivation **3**
1.1 Recall and Precision . . . 3
1.2 Indexing unnamed information . . . 4
1.3 The WWW – a decentralized and linked information pool . . . 4
1.4 Drawbacks of hyperlink based information retrieval . . . 5
1.5 Overcoming the shortcomings of hyperlink based information retrieval . . . 5

2 Defining Evolving Association Indexes **7**
2.1 Formal definition of *Evolving Association Index* . . . 7
2.1.1 Descriptors . . . 7
2.1.2 Associations . . . 7
2.1.3 Modelling Time . . . 9
2.1.4 Topics . . . 10
2.2 Intension . . . 11
2.2.1 Descriptors . . . 11
2.2.2 Associations . . . 11
2.2.3 Intension of topics . . . 12

3 Proposal of a few topic structures **14**
3.1 Accessibility components . . . 14
3.1.1 Formal definition . . . 14
3.2 Hubs and authorities . . . 15
3.2.1 Intension . . . 15
3.2.2 Formal definition . . . 16
3.3 Cluster topics . . . 17
3.3.1 Intension . . . 17
3.4 Collapsing topics . . . 18
3.4.1 Intension . . . 18
3.4.2 Formal definition . . . 19
3.4.3 Possible modifications of collapsing topics . . . 22

4 Implementation **23**
4.1 "Harvest" - the tool for retrieving articles on the web . . . 24
4.1.1 The descriptors data model . . . 24
4.2 "Linker" - the tool for editing associations . . . 25
4.2.1 The associations data model . . . 25
4.2.2 Completing the edited data to fulfill the definitions . . . 26

4.3 Implementing the topic structures . . . 26
4.3.1 Hubs and authorities . . . 27
4.3.2 Collapsing topics . . . 27
4.3.3 Cluster Topics . . . 32

5 Retrieving and analysing from a testbed **34**
5.1 The first test serie . . . 34
5.2 The second test serie . . . 36
5.2.1 Hubs and authorities . . . 37
5.2.2 "Cluster Topics" of the 2nd testbed . . . 38
5.2.3 "Collapsing Topics" of the 2nd testbed . . . 40

6 Integration of *Evolving Association Index* **44**
6.1 The use of isRelatedTo for merging topics . . . 45
6.2 The use of cites for merging topics . . . 45
6.3 The use of equivalent for merging topics . . . 45
6.3.1 Example . . . 45
6.4 The use of generalizes /specializes for merging topics . . . 47

7 Related Work **48**
7.1 The Web as a graph: measurement and models and methods[7] . 48
7.2 Mining the Link Structure of the World Wide Web[4] . . . 48
7.3 Authorative Sources in a Hyperlinked En- vironment[6] . . . 49
7.4 Finding Related Pages in the World Wide Web[5] . . . 49
7.5 A Web Community Chart for Navigating Related Communities[12] 49
7.6 Hypersearching the Web[3] . . . 50
7.7 Open Peer-Review as Web's Self-Organization Force[8] . . . 50
7.8 Metaphors We Surf the Web By[9] . . . 50
7.9 Description of Collections and Encyclopaedias on the Web using XML[10] . . . 51
7.10 The Connectivity Server: fast access to linkage information on the Web[1] . . . 51
7.11 Introduction to WordNet: An On-line Lexical Database[11] . . . 52
7.12 The Maximum Clique Problem[2] . . . 52

8 Conclusions **53**
8.1 Indexing music as possible application of *Evolving Association Index* . . . 54
8.1.1 Descriptor proposal for music . . . 54
8.1.2 Association proposal for music . . . 55

Chapter 1

Introduction and Motivation

This report describes *Evolving Association Index*, a method of indexing information through relations. This method tries to solve some of the problems found in classical indexing methods like indexing through keywords. The problems of indexing through keywords are related to the lack of an appropriate keyword taxonomy in some cases, and to ambiguosity and synonymy. Informations associated to a keyword may have different meanings due to different meanings of this keyword[1] - the keyword has ambiguous meanings. Also there may be different synonyms for a keyword and some information expected to be related to this keyword may be related only to one of its synonyms.

1.1 Recall and Precision

Information retrieval has two criterias of quality: recall and precision. Recall compares the amount of retrieved information related to a query to all information that should match the query (verified through human analysis). Precision is indirectly proportional to the amount of noise - non relevant information to the queries intension. Therefore in a keyword based system, ambiguous meanings of keywords affect the precision, and keywords containing a lot of synonyms affect the recall of information retrieval. To improve recall a thesaurus[2] of synonyms has to be used. To improve precision, keywords must be constrained to a set of non ambiguous words. Both improvements can not be automated, are error prone and produce a lot of work.

Recall and precision is also relevant for information retrieval based on link structures. This will be explained later when more details about *Evolving Association Index* are given.

[1] e.g. jaguar may be the keyword for an article about a rare specie of wildcats, or it could be the keyword for an advertisement of an expensive sports car

[2] A thesaurus can be seen as hierarchically organized categories of words. It is helpful for finding synonyms and related words - they are located in the "neighborhood" of each other in the hierarchy. While it seems quite simple to define hierarchies for narrow topics it turns out to be very difficult to define general purpose hierarchies or broad topic hierarchies. It might even be impossible to build up a whole world hierarchy.

1.2 Indexing unnamed information

As mentioned earlier, to be able to index by keywords it is nessecary to have keywords or categories. Case law is a good example for a domain where keyword indexing does not apply well.

Case Law is the law of reported judicial opinions. These are distinguished from statutes or administrative law. "Stare Decisis"[3] *(or precedent) is the basic concept of case law (common law) in which courts look to both the statute/regulation and prior court decisions to formulate opinions.*[4]

Obviously it is useful to have an index of all precedent decisions. Especially new topics are of interest for case law, because it is likely that the established laws do not apply too well to the new topics. If one law case concerns a new topic it is likely, that another law case already treated the same problem from another point of view. Because the problem is very new, the vocabulary is not established well, and therefore it is possible, that the two cases use quite different vocabulary – and therefore other indexing categories or keywords. Sooner or later, if there are enough cases concerned in this anonymous topic, the correlation between these related cases is realized and normative keywords or categories may then be derived and introduced in existing thesauri specialized in case-law.

Another example for unpredictable (and therefore difficult to categorize) informations are news articles. Some articles about news may be the only information about a event, while some campaigns or topics turn out to be interesting for longer time and lots of publications. Often this is not obvious at the beginning and even more often good naming or categorisation for these topics are found only after a while.

One of the ideas behind *Evolving Association Index* is to fill the gap between the nascency of a new topic and the awareness of its existance. This will be achieved by relating information items to each other.

1.3 The WWW – a decentralized and linked information pool

The WWW seems to be actually the biggest, fastest growing and evolving pool of information. Furthermore due to its decentralized, unorganized nature it is quite difficult (maybe event impossible) to provide a high quality keyword index according former improvements. Some recent publications[7, 4, 6, 5, 3, 1] specialize on information retrieval on the WWW without focusing on the use of keywords.

The methods described in these papers use the hyper linked structure of the Internet to retrieve and rank valuable information. The heuristic used is, that links are provided by a lot of different authors regarding the linked document as relevant. A often linked document seems to be more valuable information than lesser linked documents. Therefore the text of the linked document as well as the links text itself may be analyzed. The analysis of the link texts relate the documents in a very valuable way to a lot of occurrences of important keywords and their synonyms – often much more than the document itself (looking again

[3] STARE DECISIS - Lat. "to stand by that which is decided." The principal that the precedent decisions are to be followed by the courts. [http://www.lectlaw.com/def2/s065.htm]

[4] [http://www2.lib.udel.edu/subj/godc/resguide/caselaw.htm]

at the Web page of jaguar cars, we could find nice images and opulent phrases like "the dream of speed" or "the power of a wildcat" but not a lot of profane expressions like car, sports car). A document containing a lot of "outbound" links (links to other documents) is called a "hub". A "hub" for a topic is a document being interesting for navigation and a good starting point for a (maybe broader) topic. A document containing a lot of "inbound" links (links from other documents pointing to it) is called a "authorative source" (or authority) meaning that this document is rated (through hyper links) by a lot of other people as interesting or authorative in the sense of the keywords found in the link text.

These kind of structures and others introduced later will form the basics for *Evolving Association Index*.

1.4 Drawbacks of hyperlink based information retrieval

In a hyper linked network we have one type of associations between documents - the hyperlink. Obviously information retrieval using hyperlinks is often disturbed by hyperlink usage not related to the content (e.g. `<a>You need this Software to view our Page</a>` vs. `<a>this page describes it in more detail</a>` - while the latter link provides us with valuable information related to the document containing the link, the former link doesn't). Links are often used for navigation or advertisements. Navigation links are often used when the content tends to be rather long for use on screen display or to link to different representations of the same content (e.g. screen presentation and printer version). Advertisements are often not related to the topic at all.

1.5 Overcoming the shortcomings of hyperlink based information retrieval

This leads to the idea of using typed links providing information about their special meaning. New Internet standards like XLink [5] provide the basic infrastructure for typed links, but does not define semantic associations usable for information retrieval. Anyway XLink could be used as standard technology framework upon which semantic associations could be built.

In this report I will explain my efforts in defining and evaluating a small set of relations for indexing through links using XLink. The properties of these links should be quite application independent although the used names for them are specially tailored for use of indexing within the domain of news articles. This set should be expressive enough to derive interesting structures described later in this report. Here a preview on the relations I suggest: "relatedTo", "specializes", "generalizes", "equivalent", "cites". These relations are all results of semantic between documents and therefore an *Evolving Association Index* author will have to provide these relations. Further base relations are used to represent the publication date of documents. They are called "before", "after" and "sameDate". These relations may for sure be derived automatically.

[5] This web standart is specified at `http://www.w3.org/TR/2000/PR-xlink-20001220/`

For the testbed described later I will use different german newspapers available online.

Chapter 2

Defining Evolving Association Indexes

2.1 Formal definition of *Evolving Association Index*

Evolving Association Index can be defined as a language of PL1.

2.1.1 Descriptors

Descriptors will be used as place holders for any kind of document. Because most properties of these documents are irrelevant for the formal definition of *Evolving Association Index*, only time will be associated to descriptors. This will be necessary for deriving the temporal associations later. Descriptors will be defined as functions. We will use constants d_i for descriptors. $d(t)$ is a descriptor published at the time t, and d_t is the publication time of d.

2.1.2 Associations

Associations are relations between descriptors. Some associations provide properties making it possible to derive other associations in these cases the author of *Evolving Association Index* does not need to provide the derivable associations.

The properties of the associations

- isRelatedTo
  The arity of this relation is 2 (in fact it could also be of any arity, but it does not change the expressiveness). The properties of this relation are reflexivity, symmetry and non transitivity[1] .

[1]Note that "antisymmetric" in contrast to "nonsymmetric" denies any symmetric occurrence of this relation. 'In contrast to "symmetric", "nonsymmetric" allows symmetric use of this relation, but it is not enforced (mandatory?). Similar rules apply to other occurrences of "not" and "non".

- CITES
  The arity is 2. This relation is non reflexive, non symmetric and non transitive.
- EQUIVALENT
  The arity is 2. This relation is reflexive, symmetric and transitive. The properties of this relation are these of the equivalence relation.
- SPECIALIZES
  The arity is 2. These relations properties are not reflexive, antisymmetric and transitive.
- GENERALIZES
  The arity is 2. These relations properties are not reflexive, antisymmetric and transitive.
- BEFORE
  The arity is 2. These relations properties are not reflexive, antisymmetric and transitive.
- AFTER
  The arity is 2. These relations properties are not reflexive, antisymmetric and transitive.
- SAMEDATE
  SAMEDATE is of arbitrary arity. This relation is reflexive, symmetric and transitive. The properties of this relation are these of an equivalence relation.

The following axioms define further properties of these relations.

$$
\begin{aligned}
\textsc{equivalent} &\subseteq \textsc{isRelatedTo} \\
\textsc{cites} &\subseteq \textsc{isRelatedTo} \\
\textsc{specializes} &\subseteq \textsc{isRelatedTo} \\
\textsc{generalizes} &\subseteq \textsc{isRelatedTo} \\
\textsc{specializes}^{-1} &= \textsc{generalizes} \\
\textsc{generalizes}^{-1} &= \textsc{specializes} \\
\textsc{specializes} \cap \textsc{generalizes} &= \textsc{equivalent} \\
\textsc{before}^{-1} &= \textsc{after} \\
\textsc{after}^{-1} &= \textsc{before} \\
\textsc{before} \cap \textsc{sameDate} &= \emptyset \\
\textsc{sameDate} \cap \textsc{after} &= \emptyset \\
\textsc{after} \cap \textsc{before} &= \emptyset \\
(A, B) \in \textsc{before} &\Leftrightarrow t_A < t_B \\
(A, B) \in \textsc{sameDate} &\Leftrightarrow t_A = t_B \\
(A, B) \in \textsc{after} &\Leftrightarrow t_A > t_B
\end{aligned}
$$

where t_d is the publication time of d

$$\textsc{before} \cup \textsc{sameDate} \cup \textsc{after} = \mathcal{D} \times \mathcal{D}$$

where $\mathcal{D}$ is the set of all descriptors.

$$(A, B) \in \textsc{equivalent} \wedge (A, C) \in R \Rightarrow (B, C) \in R$$

where $R \in \{\textsc{isRelatedTo}, \textsc{equivalent}, \textsc{specializes}, \textsc{generalizes}\}$

More derived relations

These associations are all derived using the base relations. Authors do not need to provide them. They are mainly intendet to be usefull for definitions of topic structures described later.

- $\textsc{isWeaklyRelatedTo}(d_1, d_2) \Leftarrow \textsc{isWeaklyRelatedTo}(d_1, d_3) \wedge \textsc{isRelatedTo}(d_3, d_2)$
- $\textsc{isWeaklyRelatedTo}(d_1, d_2) \Leftarrow \textsc{isRelatedTo}(d_1, d_2)$
- $\textsc{isStronglyRelatedTo}(d_1, d_2) \Leftrightarrow \textsc{isRelatedTo}(d_1, d_2)$
- $\textsc{serie}_2(d_1, d_2) \Leftarrow \textsc{isStronglyRelatedTo}(d_1, d_2) \wedge \textsc{before}(d_1, d_2)$
- $\textsc{serie}_n(d_1, ..., d_{n-1}, d_n) \Leftarrow \textsc{serie}_{n-1}(d_1, ..., d_{n-1}) \wedge \textsc{serie}_2(d_{n-1}, d_n)$

2.1.3 Modelling Time

The publication time in descriptors is used to derive the associations BEFORE , AFTER and SAMEDATE . Therefore we need to define a representation of time and how to calculate time intervals due to this representation. It is not possible to provide the same accuracy of time for all descriptors: While some descriptors may have publication time specified in day, hour and minute (e.g. online news publishing) others may only provide the month they where published in. I will now describe the syntax used for modelling time, duration (or time spans) and time intervalls and also the rules to derive BEFORE , AFTER and SAMEDATE .

Informal descriptions

- Time can be specified with different degrees of precision from years to seconds.
- Durations should be normalized (e.g. 1 hour 10 minutes instead of 70 minutes)
- Precision of durations is specified through its smallest unit.
- Time intervals are modelled through start time and end time. The least precise time specifies the intervals precision, durations may be roundet as expected.

Syntax

$$\begin{array}{r}
\hline
year \leftarrow 0... \\
month \leftarrow 1..12 \\
day \leftarrow 1..31 \\
hour \leftarrow 0..23 \\
minute \leftarrow 0..59 \\
second \leftarrow 0..59 \\
\hline
\end{array}$$

$$\begin{array}{lcl}
\hline
time & \leftarrow & year \\
 & \vee & year.month \\
 & \vee & year.month.day \\
 & \vee & year.month.day[h] \\
 & \vee & year.month.day[h:min] \\
 & \vee & year.month.day[h:min:sec] \\
\hline
\end{array}$$

$$\begin{array}{ll}
\hline
span & \leftarrow year\mathbf{Y} \\
 & \vee year\mathbf{Y}month\mathbf{M} \\
 & \vee year\mathbf{Y}month\mathbf{M}day\mathbf{D} \\
 & \vee year\mathbf{Y}month\mathbf{M}day\mathbf{D}h\mathbf{h} \\
 & \vee year\mathbf{Y}month\mathbf{M}day\mathbf{D}h\mathbf{h}min\mathbf{m} \\
 & \vee year\mathbf{Y}month\mathbf{M}day\mathbf{D}h\mathbf{h}min\mathbf{m}sec\mathbf{s} \\
\hline
\end{array}$$

$$\begin{array}{lcl}
\hline
timeIntervall & \leftarrow & time - time \\
\hline
\end{array}$$ [2]

Rules for relations between times

- **same time**
  Two events occur at the same time, if the duration is 0 after reducing the precision of the more precise time to that of the less precise.
- **before/after**
  If two events do not occur at the same time one occurs before the other and vice versa.

2.1.4 Topics

Topics are documents or sets of documents.They are defined based on the structure of the graphs spanned through the associations. Topics may as well be single outstanding documents. Topics can also be regardet as documents - in this case topics may provide new virtual documents. These documents may for example represent folders containing other documents. Concrete topics will be discussed later and at this point there are no general properties of topics (except that they may be grouped together in classes according to their properties or specification).

[2] first time of a time intervall is the start time

2.2 Intension

This section will describe the semantics of all syntactic elements used in *Evolving Association Index*.

2.2.1 Descriptors

Again, descriptors are not explained further here. The semantic of descriptors is application dependent. The former example will interpret a descriptor as being information about[3] a news article .

2.2.2 Associations

The semantic of associations is application dependent – associations for indexing music or compositions may have completely different meanings than associations for indexing news articles, although they may share the same properties. It is especially important for an author of *Evolving Association Index* to know the semantic of the associations of his domain.

Nevertheless there are some guidelines on the meaning of the different associations I will describe now. They're tightly bound to the formal definitions:

isRelatedTo

The isRelatedTo association is the most general. It is not directed and only describes that the data items are semantically related (keep the non transitive nature of isRelatedTo in mind).

equivalent

Two descriptors are equivalent if they are linked together through an equivalent association. All of their associations will be the same. The data items represented by these descriptors are semantically exchangeable information. This link type implies a isRelatedTo association.

cites

One descriptor cites another if formers data item cites the others data item explicitly. This link type does not implie a isRelatedTo association.[4]

generalizes

d_1 generalizes d_2 when the content represented by d_2 is addressed in whole by the content represented by d_1 at same or less precise level of detail. This link type implies a isRelatedTo association.

[3] not the news article itself

[4] Imagine a catalog citing other documents. Why should these documents be related to a catalog they appear in?

specializes

d_1 SPECIALIZES d_2 when the content represented by d_1 addresses part or all of the contents represented by d_2 at same or more precise level of detail. This association implies a ISRELATEDTO association.

before , after and sameDate

These associations[5] are used in accordance to the time modeling and the publication date of the descriptors.

isStronglyRelatedTo and isWeaklyRelatedTo

ISSTRONGLYRELATEDTO is a synonym for ISRELATEDTO . ISWEAKLYRELATEDTO is the transitive hull of ISSTRONGLYRELATEDTO . ISWEAKLYRELATEDTO can be used to find groups of descriptors being all accessible from each other.

ISWEAKLYRELATEDTO does not represent the authors intension of ISRELATEDTO - e.g. we could not determine, if a descriptor is especially representative for this group, if we only had transitive ISRELATEDTO associations. Having only transitive ISRELATEDTO associations would also mean that adding a bridge between two topics would merge these topics together, loosing all information about the former subtopics. This would reduce the precision of queries to an *Evolving Association Index*.

serie$_2$ and serie$_n$

We will look at data items represented by associated descriptors in SERIE$_2$ or SERIE$_n$. Each of these data items - except the first - where published explicitly related to (=ISSTRONGLYRELATEDTO) a former released data item. Maybe these data items are about a quite actual topic[6]. This topic is evolving while the data items are published and the most recent data item (one end of the serie) is maybe also the most representative.

2.2.3 Intension of topics

Topics are meant to be sets of documents all sharing in part the same semantic. Assuming that a reader searching for a special topic found the right topic we expect all documents in this set to be relevant to the reader. On the other hand it is possible that topics provide meta information about the documents making it easier for the reader to decide which documents are the most interesting in his special case. Documents of series for example can be read following the publication date. If only a short summary is required, the first and last document of the series may turn out sufficient as summary. Topics could for example provide a way to navigate through specialization of some subjects. Concrete intentions of special topic structures will - like the properties - be discussed later.

[5] BEFORE , AFTER and SAMEDATE are base relations, but they do not have to be specified by the author

[6] actual means actual in the authors view, not in the readers view

isWeaklyRelatedTo and serie are relations rendered based on other relations - therefore these relations may also be concerned as topics what was not done at this point. Nevertheless they will be used as base for construction of topics.

Chapter 3

Proposal of a few topic structures

In this chapter I will describe a few topic structures of interest. These structures were found by looking at the testbed described later. Some structures were found directly by looking at a visualization of the index – hubs and authorities for example are very obvious and easy to see in the testbed . Other structures were found indirectly: Cluster topics (described later) for example were found as strategy to compensate the negative impact of bridges on accessibility components (also presented here).

To rate the quality of a topic structure it is necessary to implement the algorithms and apply them to the testbed. While performing this kind of proof of concept it often turns out that there are a lot of parameters to fine tune to obtain good results.

The structures described here are certainly not a complete representation of possible *Evolving Association Index* use, rather than some first experiences with the possibilities of indexing through typed links.

3.1 Accessibility components

The most obvious topic structures are probably accessibility components. The network of documents and relations normally falls apart in components consisting of documents that are all accessible through chains of relations. Apparently sometimes documents exist relating to a lot of different things – everything seems to be related to each other. Therefore accesibility components sometimes span documents, we would expect not to belong together, but in general accessibility components provide a simple clustering of related documents. Later I will describe methods to overcome the shortcomings of accessibility components.

3.1.1 Formal definition

Accessibility components over the isRelatedTo relation are already defined through isWeaklyRelatedTo . If we are interested in components spanned through cites relations, we can simply adopt the same method. Note, that cites in contrast to isRelatedTo is directed, which affects the accessibility.

Because the other relations all imply isRelatedTo , it is not nessesary to define accessibility components spanned through them.

3.2 Hubs and authorities

These terms and structures are already described earlier and their quality in the domain of information retrieval is by now proven by very successful projects like the search engine Google[1] and crawler engines like the Clever Project[3].

Returning to the properties of hubs and authorities: Hubs are nodes in a graph having a lot of outbound edges and authorities are nodes with a lot of inbound links.

In an *Evolving Association Index* we have typed links and therefore we need to analyze the impact of these types (and their properties) on hubs and authorities:

- cites : This is the case examined in all papers about hubs and authorities – therefore the behavior and meaning of hubs and authorities described by these links should be exactly like described in these papers. The lack of symmetry in cites associations result in a clear separation of hubs and authorities. The lack of transitivity ensures that the associations are not automatically spread across the graph and therefore hub/authority structures directly represent the intension of the link authors.

- isRelatedTo : Because these associations are symmetric, it is not possible to distinguish hubs and authorities. The terms hubs and authorities are therefore obsolete here and I will call these kind of nodes "relation centers" pointing out that these nodes are central nodes in a kind of relation star structure. Again, the lack of transitivity ensures that these structures are tightly bound to the intension of the *Evolving Association Index* authors .

- equivalent : Nodes linked together through this type of association do not provide outstanding hubs or authorities. This is due to the transitive, symmetric nature of the equivalent relation: all equivalent nodes have the same amount of in and outbound links. Therefore the equivalence relation should not be used to retrieve hubs and authorities.

- specializes / generalizes : Hubs and authorities built through these relations are quite interesting because of the semantic of these links and because of the dual nature of specializes /generalizes – A very special authority is also a hub for more general nodes and vice versa. The transitivity of these associations does not equalize hubs/authorities and other nodes due to the antisymmetry. The central nodes of hub/authority stars get also enforced through the transitivity when their relation partners get related to other nodes the same way (this is due to the transitivity).

3.2.1 Intension

It is quite obvious that authorities seem to be a good information source due to the fact that a lot of linkers (people or automatic generators writing hyperlinks)

[1] see http://www.google.com

rate this document as relevant (through linking it). When analyzing authorities centered in a star of SPECIALIZES associations (respectively GENERALIZES), it is obvious that we will get a document about a very narrow (respectively very broad) topic, but while navigating along the rays of the star the documents topics get broader (respectively narrower).

Hub nodes are in contrast not as explicit about the contents quality – a authority was (implicitly) rated as relevant through a lot of associations pointing to it. In contrast hubs provide a lot of associations but (maybe) no other node (author) rate this document as relevant. Nevertheless hubs tend to pull together related documents – they can be assumed to be some kind of index or catalog documents.

If we look at the publication date [2] and the derived temporal associations it sometimes turns out that hub documents have a lot of references to nodes in the future. This is especially interesting, because these documents tend to be good entry points to a topic maintained, to always represent the actual state of the topic – in online news publishing the editors often provide these kind of hubs as a convenient overview about an actually evolving topic.[3]

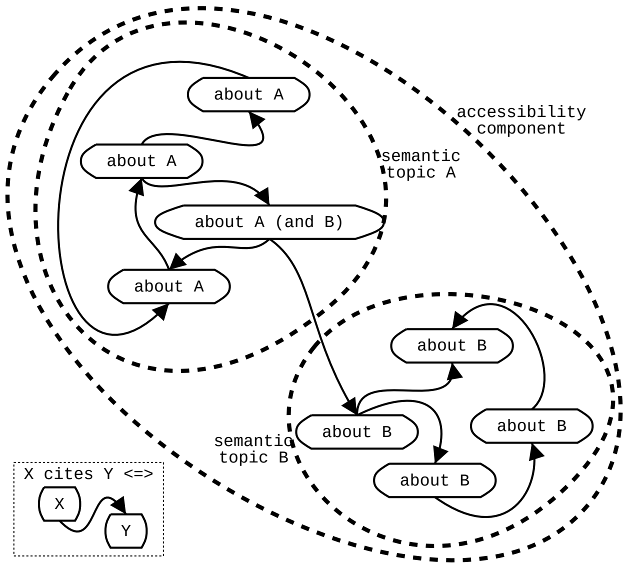
3.2.2 Formal definition

The formal definition is quite obviously based on the number of inbound and outbound associations. The selection, if a node having n in/outbound links is a hub/authority on the other hand is not obvious at all and is surely application dependent. Therefore the filter function will not be defined here. It is likely that the amount of authority and hub documents in online news publishing is relative to the amount of all documents constant, which would enforce a kind of rule like "10% of all Documents are good hubs/authorities". Concrete filters have to be found due to experiments on testbeds and real data and will therefore not be formalized here.

[2] the publication date is meant to be the date of the first publication in contrast to a possible revision date

[3] Note that we registered the publication date, not the date of the last revision

3.3 Cluster topics



3.3.1 Intension

As obvious in the example, sometimes unrelated topics[4] tend to be connected through relations. This is based on the fact that some facets of a document may be off topic. Because these kind of bridges are typically spread all over the document space, in the end everything could be connected to one big topic, if we look at simple accessibility components (e.g. found through spanning tree algorithms) as topic. One solution might be to further atomize the information retrieved: e.g. the documents could be dividet into paragraphs or facets. It is likely that this does not solve the problem, but to transfer it to another level where unrelated structures of paragraphs get associated to each other.

At a first glance it seems obvious that not all types of links tend to build bridges. Let's look at the different types of relations:

- EQUIVALENT relations for example (if not misplaced) would never provide inter topic relations, due to their semantic constraint to only relate semantically exchangeable information (topics again are meant to be some

[4]In this case topics are meant to be sets of semantically related documents – not fortunately the result of a topic structure

kind of semantically related documents).

- Neither SPECIALIZES nor GENERALIZES neither would build bridges. We remember that the relations where defined syntactically antisymmetric and transitive. This means, that other topics would only be related to each other, if one topic completely specializes a facet of one document of the other topic. In this case the complete former topic can be seen as a facet of the latter one which is an interesting situation – worth to be called a topic – and not a problem. Therefore we should introduce a subtopic. In fact Collapsing and Cluster topics will find similar nested topics – but based on other ideas.

- ISRELATEDTO as the most general relation is a possible source for bridges. The example above could have been designed with ISRELATEDTO relations in exchange.

- CITES is the most obvious candidate for potential bridge production (see the example), although due to the nonsymmetric nature of CITES , the bridges impact of citations is not so dangerous as the impact of ISRELATEDTO .

When looking at the phenomena of inter topic bridges connecting unrelated topics to big graph components, it turns out that off topic (or inter topic) relations are quite weak: this means that while topics tend to be glued together by a lot of relations (sometimes topics are even linked transitively, meaning that every node of a topic is pairwise adjacent to all other nodes of this topic), inter topic relations are often the only bridge between topics. Cluster topics are based on this property of topics. Bridges are found and removed from the *Evolving Association Index*. The resulting graph components are called clusters. Normally bridges are not unique and there are quite a lot of possibilities to remove bridges. The idea here is to remove the bridges connecting the biggest possible components (in contrast to remove bridges connecting single documents to topics). After having clustered the graph through removing single bridges it is maybe useful to recluster the first order clusters through deletion of stronger inter topic bridges. This means to build clusters through deletion of 2 (or later more) bridges. The re-clustering could be organized in a tree structure to provide a kind of browse-able directory with more and more specialized topics according to the depth of the clusters in the tree.

3.4 Collapsing topics

Collapsing topics can be seen as the opposite of Cluster topics: we do not group documents by removing weak relations and searching for graph components, but we look for strongly related subgraphs of the *Evolving Association Index*.

3.4.1 Intension

As seen in the testbed, strong linked structures tend to belong to one topic. The strongest possible relations between nodes are transitive (therefore the EQUIVALENT relation is transitive!). If we find a transitively completely linked subgraph it is likely to be semantically related. Therefore we define these structures as topics. Transitively connected subgraphs are also often called cliques. The most interesting cliques are the biggest ones – this problem is widely known as the maximum clique problem[5]. Once such a topic is found all it's nodes can be collected into a folder (or directory) node. This step is called collapsing a topic. Of course the new topic folder is connected to the rest of the graph by all links connecting the clique to the rest of the graph. After this step we can again search for the next maximum clique and re-collapse, until the maximum cliques size drops below a minimum (obviously a clique of only two nodes may not provide a very interesting topic). The collapsed folders construct a kind of browse-able directory structure. Again this structure specializes in depth, but further more nodes near the root may tend to contain less elements (documents and topic folder nodes) than the leafs (and other deeply nested topic folder nodes). The leafs only contain documents and inner nodes may contain a mixture of documents and topic folder nodes.

Now let's again talk about the different types of associations and their impact on collapsing topics:

- isRelatedTo relations are very well suited to this topic structure. They are wide spread and undirected.[6]
- cites is a directed relation and therefore it is less likely to produce transitive subgraphs – only if every document in a subgraph cites the other, they are transitively linked together. It may turn out useful to ignore the direction of cites while collapsing topics in order to find more clusters.
- specializes / generalizes implies isRelatedTo and therefore already received consideration.
- equivalent implies equality of all relations of its members and can therefore also be neglected.

3.4.2 Formal definition

The collapsing topics structure $collTop(G)$ is a function applied to a Graph

$$G = (V_G, E_G)$$

The vertices V_G are defined as

$$x \in V_G \Rightarrow x = v \vee x = \{\overline{v_1}, ..., \overline{v_n}\}$$

where v is a normal vertex in G, but $\overline{v_i}$ aren't vertices in G (but in another Graph) - the set $\{\overline{v_1}, ..., \overline{v_n}\}$ is the vertex in G. It will be called folder vertex. The edges E_G are defined as

$$\forall e \in E : (e = (v_{start}, v_{end}) \wedge v_{start} \in V_G v_{end} \in V_G) \, .$$

$collTop(G)$ is a recursive function:

$$collTop(G) := \begin{cases} \texttt{if} nextCollTop(G) = G : G \\ \texttt{else} : collTop(nextCollTop(G)) \end{cases}$$

[5]Later in this report more will be said about the maximum clique problem
[6]therefore it is easier to obtain transitive subgraphs

$nextCollTop(G)$ performs one step of the recursion – one collapse of some vertices into a folder vertex:

$$nextCollTop(G) = \begin{pmatrix} V_{nextCollTop_G} \\ , E_{nextCollTop_G} \end{pmatrix}$$

$nextCollTop(G)$ therefore describes a new Graph based on G composed of

$$V_{nextCollTop_G} = V_G \setminus V_{bC_G} \cup \{V_{bC_G}\}$$

and

$\forall (v,w) \in E : [(v,w \in \{V_G \setminus V_{bC_G}\} \Rightarrow (v,w) \in E_{nextCollTop_G}) \land (v,w \in V_{bC_G} \Rightarrow (v,w) \notin E_{nextCollTop_G}) \land (v \in V_{bC_g} \land w \in \{V_G \setminus V_{bC_G}\} \Rightarrow (\{V_{bC_G}\},w) \in E_{nextCollTop_G}) \land (w \in V_{bC_g} \land v \in \{V_G \setminus V_{bC_G}\} \Rightarrow (v,\{V_{bC_G}\}) \in E_{nextCollTop_G})]$.

Note that $E_{nextCollTop_G}$ is the smallest set conforming the former definition.

$V_{bC_G} \in \overline{C_G}$ where V_{bC_G} denotes one of the biggest cliques collected in $\overline{C_G}$.

$$\begin{array}{lll} \forall V \in \overline{C_G} : (& & V \in C_G \\ & \land & |V| \geq t \\ & \land & \forall W \in C_G \setminus \overline{C_G} : |W| < |V| \\ ) & & \end{array}$$

with

$$\forall V \in C_G : ((V \times V) \subseteq E_G)$$

Therefore C_G is the set of all subsets of V_G which are connected transitively in G. They are called cliques. $\overline{C_G}$ are all elements of C_G with maximal cardinality. They are the maximum cliques of G. The constraint $|V| \geq t$ provides the possibility to set a treshhold parameter t. Below this cardinality we are not interested anymore in big cliques. A useful value would be maybe 2: this prevents collapsion of cliques containing only 2 vertices.

A simplification and it's impact on the semantic and the resulting data

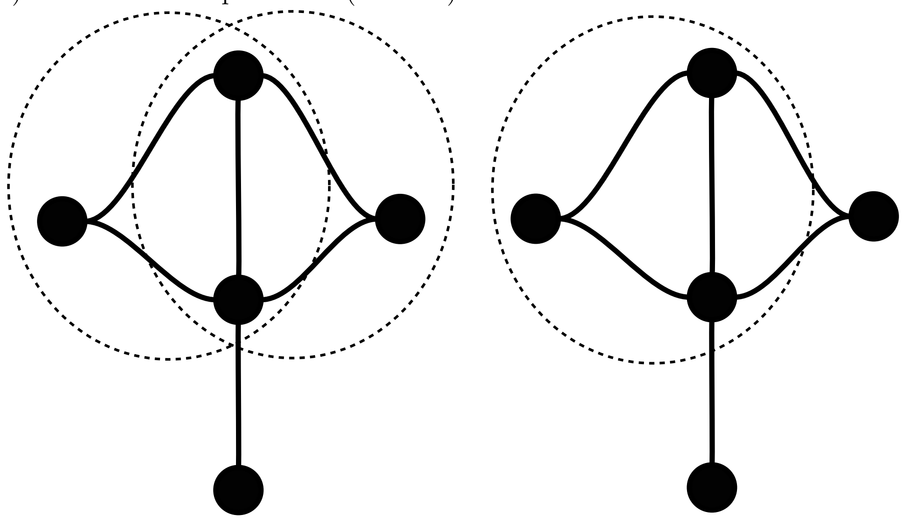
Note that earlier we defined $V_{bC_G} \in \overline{C_G}$ (or more verbose $V_{bigClique_G} \in \overline{C_G}$). This implied, that if $\overline{C_G}$ contains more than one element, every $V_{bigClique_G}$ is collapsed in the same step. Semantically this makes sense, because every $V_{bigClique_G}$ of same size in the same iteration is processed equally. Especially overlapping cliques are collapsed equally, resulting in their shared documents to be contained in all the folder vertices of the overlapping cliques[7]. Therefore the vertices are not arranged in a proper hierarchy. The number of vertices in the result of a collapsion may be much larger than in the unprocessed data.

A simplification is to only collapse one $V_{bigClique_G}$ in each iteration. Therefore V_{bC_G} is exactly one set of $\overline{C_G}$. The advantage would be a cleaner hierarchical resulting data structure without duplicate nodes. On the other hand cliques of the same cardinality in the same iteration are not treated equally, and therefore

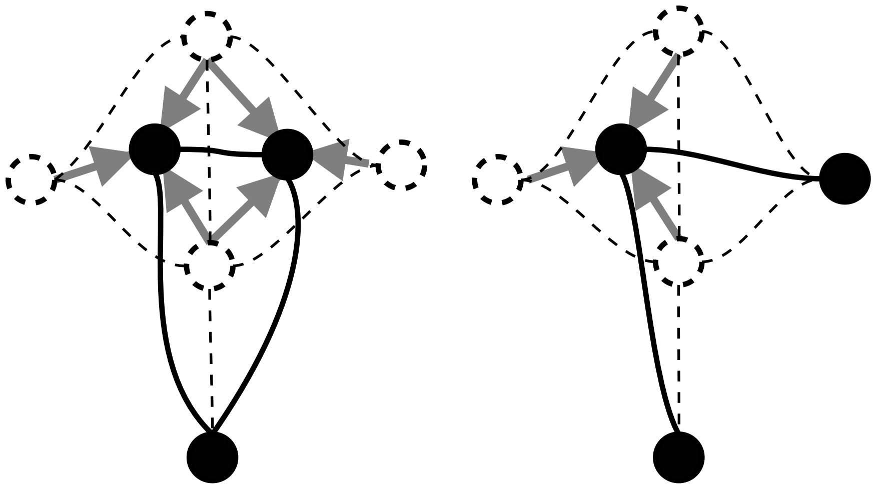
[7]This again leads to multiplication of out-/inbound edges to shared vertices.

maybe some collapsions get impossible. The results of using this simplification or not are not identical (see the example below).In the example implementation I used the simplification. [8]

The latter Illustrations show a comparison of collapsing topics with (right side) and without simplification (left side).

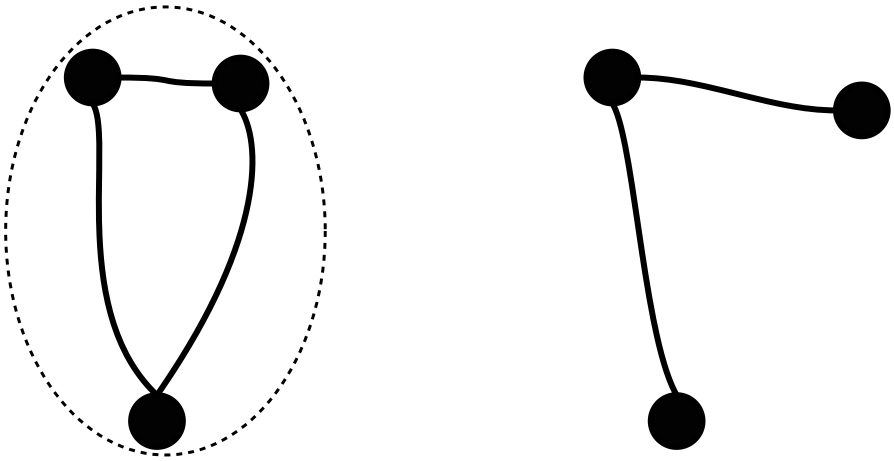


(1)At this state collapsing topics finds two big cliques. The simplification only finds one.



(2)Without simplification two of the nodes get duplicated to be able to collapse in two folders. Therefore there is also an edge that is duplicated.

[8]especially because the biggest clique algorithm turned out to be simpler to implement this way



(3)Obviously there is a difference in the results with and without simplification. Note that the simplification is not confluent.

3.4.3 Possible modifications of collapsing topics

The fact that reality tend not to be perfect also affects *Evolving Association Index*. Certain groups of nodes may semantically provide a topic but not be linked together transitively - maybe almost transitively, but not completely. Therefore an interesting extension would be to modify the biggest clique algorithm to be tolerant to a certain amount of missing associations. This amount may for example be proportional to the cardinality of the (unperfect) clique.

Chapter 4

Implementation

In order to find derived associations it is necessary to use a real world testbed. Therefore a prototype of *Evolving Association Index* was implemented . Its descriptors are specialized on news articles on the web. Parts of the prototype are implemented as server side web application, accessible through normal web browsers. The advantage of designing the application using this server/client concept was the ability to store the data in a consistent way at a central place, without restriction to one machine or operating system for the collection of the testbed. Any web browser could be used for this task.

Almost every part of the prototype was built using XML based technologies like XSLT [1], and XLink. Further web standards used are HTTP[2] and URL[3] encoding[4]. The software components for collapsing topics and cluster topics were implemented in haskell. I For the implementation open source components were used exclusively. The different components are not 'glued' together in one application. Access can be obtained either through small unix shell- and python scripts or through the WWW.

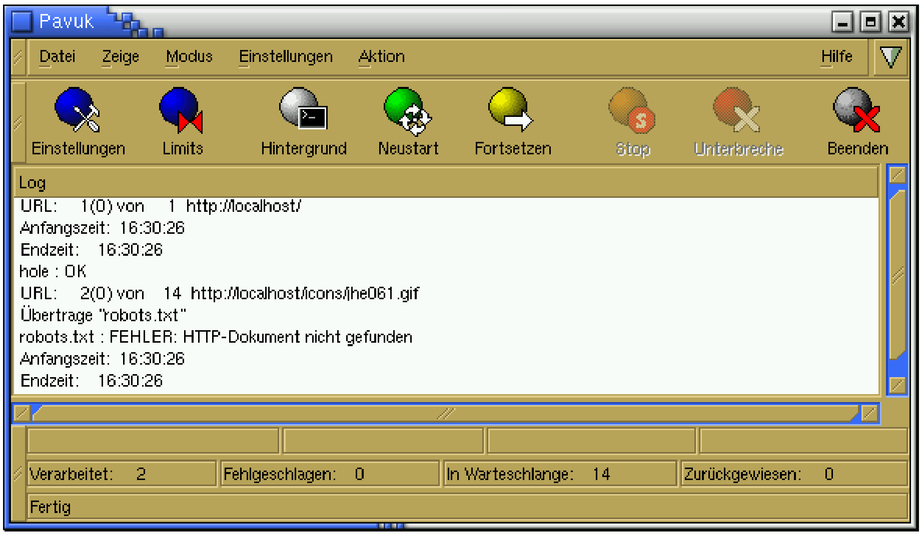
The prototype is separated in different parts: A tool for collecting and copying articles on the web, and another one for editing the associations. The data produced by these tools can be integrated into one *Evolving Association Index* file conforming with the DTD proposed later. This data can be visualized using two different tools – one specialized on *Evolving Association Index* and another for generic XML visualization. Both tools produce XHTML displayable in most modern web browsers. Another tool produces a 2 dimensional topographic graph visualisation of the associations that can be printed. Two more tools are able to calculate collapsing topics or cluster topics. The output of these tools is plain text. An appropriate visualisation for those structures is not implemented in the prototype.

[1]This web standart is specified at `http://www.w3.org/TR/1999/REC-xslt-19991116`
[2]This web standart is specified at `RFC 2616`
[3]This web standart is specified at `http://www.w3.org/Addressing/URL/Overview.html`
[4]This web standart is specified at `RFC 1738`

4.1 "Harvest" - the tool for retrieving articles on the web

This tool is used to mirror pages by providing their URLs. The publication date and the title of the articles must be specified (the default date is the actual date). It is also necessary to enter the publisher or provider of the article. Because a fixed set of providers was given, a pull down menu with fixed values for this information was convenient. Submitting this information leads to a (server side) check, if all fields are entered correctly. If this check passes, a descriptor file is created and the URL is mirrored with all its images, sub frames and other contents. The links found on this page are not followed recursively. All components of one page are downloaded in their own separate subdirectory (these folders are named according to the actual time in order to obtain a unique name).

The tool used for mirroring is pavuk.[5] Pavuk is a UNIX program used to mirror contents of WWW sites, locations or files. It gathers documents through HTTP, FTP, Gopher and as option through HTTPS (HTTP over SSL) servers. Pavuk has an optional GUI[6] based on the GTK+[7] widget set, but for the web application this GUI cannot be used.



4.1.1 The descriptors data model

The descriptors are saved as XML documents – every descriptor in one file. This is safer in case of application crash, because then only one descriptor is lost, not the complete data file. The name of the document is the articles title in URL encoded format – this means that spaces are replaced by "+", and most special characters are transformed in "%xx" where "xx" is the hexadecimal representation of the characters ASCII code. More about URL encoding can be read in RFC 1738.

[5] actually pavuks Homepage can be found on `http://www.idata.sk/ ondrej/pavuk/about.html`
[6] Graphical User Interface
[7] Homepage: `http://www.gtk.org/`

Further steps showed that it was easier to access all descriptors at once, instead of processing them separately. Therefore a list is generated using XSLT and some other small tools (e.g. for generating a list of all files in a directory) The URL encoding of the single descriptors files was necessary to access these documents through XSLTs `document()`-function.

The following DTD explains the structure of this descriptor list called `allDescriptors`.

```
<!ELEMENT allDescriprors  (descriptor*)>
<!ELEMENT descriptor      (title?,
                           origin,
                           mirror,
                           keywords?,
                           comment?,
                           publicationDate?
                          )>
<!ELEMENT title           (#PCDATA)>
<!ELEMENT mirror          EMPTY>
<!ATTLIST mirror url      CDATA #REQUIRED>
<!ELEMENT publicationDate (#PCDATA)>
<!ELEMENT origin          EMPTY>
<!ATTLIST origin url      CDATA #IMPLIED
                 name     CDATA #IMPLIED>
<!ELEMENT keywords        (keyword*)>
<!ELEMENT keyword         (#PCDATA)>
<!ELEMENT comment         (#PCDATA)>
```

4.2 "Linker" - the tool for editing associations

After having mirrored a web page using Harvest, descriptors can be linked together using this tool. The user is asked for a source, and a destination URL, and for the type of association required. The URLs are those pointing to the mirror pages, but they will later be used to determine their descriptor. It was better to use the mirrors URLs instead of URIs to the appropriate descriptors, because it is possible to drag the mirror URLs directly from a web browser window after verifying the content of the mirrored page. A small problem appears because the URLs copied out of a browser are always absolute (including the servers location) and the mirror URLs used in the descriptors are relative.

4.2.1 The associations data model

Associations are stored individually in an XML document. The names of these files are calculated according to the actual time (in seconds since 1970 Jan.1st). This was a common way to ensure the creation of unique filenames and it can be replaced by any other unique ID algorithm[8]. For common use these documents can be merged together with the descriptors to provide an XLink conforming link base of extended links. This link base will be used as base for any work on *Evolving Association Index*.

[8]In fact timestamps as IDs are only useful while the editing tool is used by only one person at a time.

This link base contains all information nessesary for *Evolving Association Index* related processing, like collapsing topics. The references to external files are references to mirrored documents. They are all assumed to be beneath a folder called `/mirrors/` located beneath the folder containing the link base. Therefore all these references start with `../mirrors/`.

4.2.2 Completing the edited data to fulfill the definitions

This step is responsible to preserve the properties defined for the different association types. Because it turned out, that the generation of many associations was not nessesary for most algorithms, this step is not precompiled. This means, that the applications by now are responsible that the associations fulfill all constraints. This means, for example, that for two nodes A and B and and EQUIVALENT relation between them, all associations of A have to be duplicated and linked to B (and vice versa).

Because mainly CITES associations were analysed, this step hardly became nessecary.

4.3 Implementing the topic structures

The prototype implementations of the topic structures are realized either in XSLT or in Haskell[9], a strongly typed, lazy evaluated functional programming language. I chose not to implement everything in XSLT because of the lack of sophisticated libraries or frameworks for more complicated structures – in fact XSLT was never intended to be a full featured programming language but rather a transformation language for XML. Besides the fact that it is an official web standard becoming more popular, the XML awareness is one of the biggest advantages of using XSLT for simple computations[10]. This means, that it is not nessesary to integrate XML parsers and generators for the implementation of simple tools – simply apply a XSLT sheet on some XML data (a file or a URL) and the resulting data is well formed XML.

To implement the more complicated computations Haskell was chose due to its good readability (in my opinion) and the availability of useful 3rd party libraries supporting the tasks. The readability is due to the fact that the implementation is mostly a simple conversion between mathematical formulations of properties of sets (found in the formal definitions) into Haskell formalisms of set operations and properties. Haskell and its libraries are mostly created in an academic surrounding and are therefore tailored to be quite expressive.They can be used for the implementation of well readable algorithms. Haskell calculations turned out to be much more performant than equivalent XSLT transformations with available XSLT processors.

Further libraries or tools used in the context of implementing the topic struc-

[9] see http://www.haskell.org

[10] according to http://lists.xml.org/archives/xml-dev/200103/msg00942.html XSLT is touring complete. At http://www.unidex.com/touring/ there is a Project called TMML: TouringMachineMarkupLanguage. It consists of a DTD for modelling Touring machines and an XSLT stylesheet for emulation of this touring machine. This proves that XSLT can be used for any computable problem – but this does not imply that it is very practical

tures are FGL[11] and GraphViz[12]. GraphViz was used to produce the graph structures printed in this paper and for debugging. It provides automatic layouting and visualization of graph structures based on a simple graph specification language. The visualizations can be generated in a big variety of graphic formats like PostScript, GIF and many more.

FGL is a Haskell (and ML) framework for applications using graph structures. It provides efficient data structures for representations and queries on graphs. Furthermore there is a set of standard algorithms for common graph operations like spanning trees, search in graphs, maximum independent node sets, voroni diagrams, shortest paths and many more.

4.3.1 Hubs and authorities

As mentioned earlier, no concrete hub and authority definition is given here. Therefore there is also no concrete implementation of hubs and authorities. For the analysis of topic structures, it is enough to sort the descriptors acording to their in- and out- degree. This is done through a short XSLT stylesheet, that sorts the output of the visualisation according to the in- or out- degree. Now it is simple to descide manually which document to rate as hub or authority.

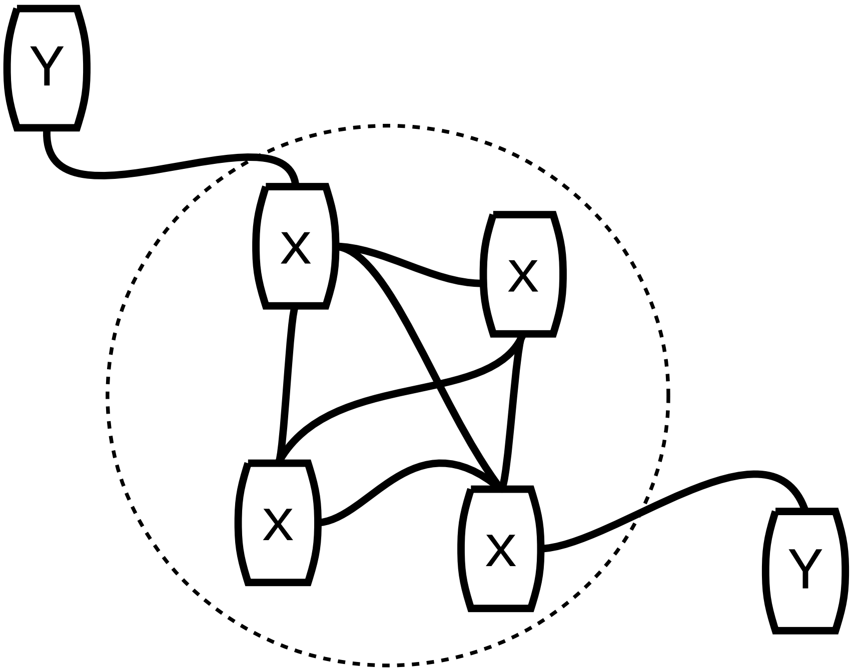
4.3.2 Collapsing topics

The algorithm for collapsing topics is based on the maximum clique problem. Assuming we have a solution to the maximum clique problem providing us with a subset of all nodes representing the maximum clique, the algorithm would look like this:

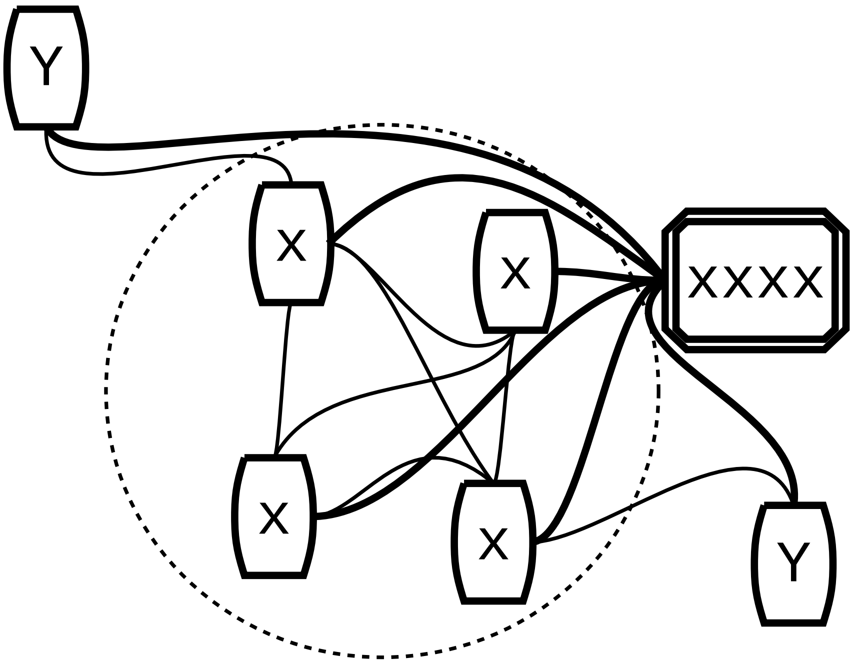
- while `maximumClique` $> n$
  - insert a folder node representing `maximumClique` into the index
    - connect this folder node to all nodes of all edges containing nodes of `maximumClique`
  - remove `maximumClique` from the index (all nodes and associated edges)

[11] It is maintained by Martin Erwig `erwig@fernuni-hagen.de` at the FernUniversitaet Hagen

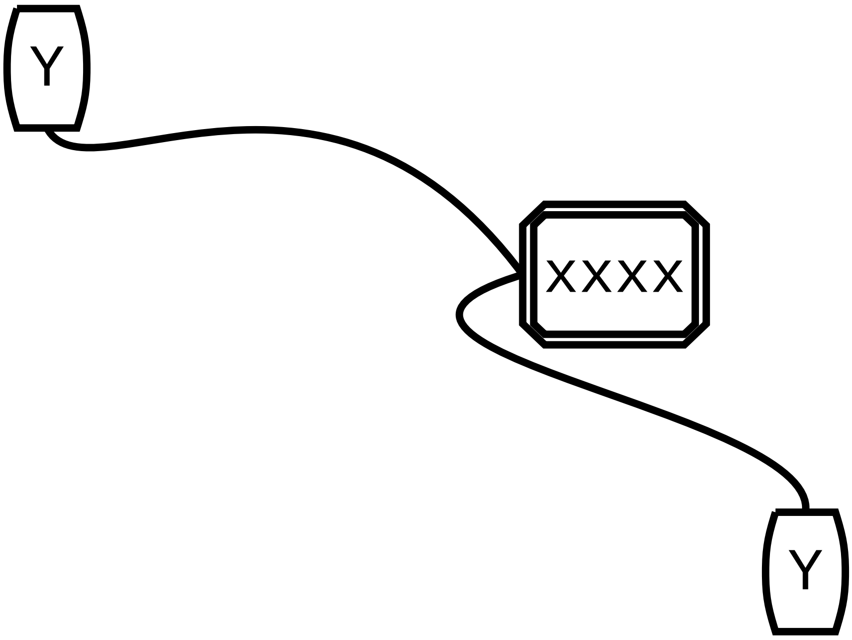
[12] see `http://www.graphviz.org`



(1)The `maximumClique` is found



(2)The new folder node is created and linked to all nodes found in edges of the `maximumClique`.



(3)Now the `maximumClique` can be deleted. (The information to reconstruct the graph can be stored in the folder node)

Unfortunately the maximum clique problem is one of the first problems shown with NP-complete time complexity, which means that for exact algorithms the time used to compute the maximum clique increases exponentially with the number of nodes[2]. Nevertheless approximations to this Problem with polynomial time complexity[2] exis. In the implementation no approximation was used to keep the algorithm simple and readable. According to Immanuel M.Bomze - *The Maximum Clique Problem, pg.3*[2] a set of nodes S is a maximum clique of a graph G if and only if S is a maximum independent set of nodes in $\overline{G}$ where $\overline{G}$ is the complement of G. It is a complement according to the edges – for $G = (V, E)$ with nodes V and edges E the complement is $\overline{G} = (V, (V \times V)/E)$. This definition was chosen, because the maximum independent set is implemented in FGL and the complement is easy to calculate. An obvious and simple optimization would be to first divide the graph into it's components (also provided in FGL – a forest of spanning trees provide us with components) and to search for cliques in these components. This optimization is applicable because cliques obviously never span over more components – nodes of different components are unreachable for each other.

Haskell code of the implementation

For understanding of the algorithm it is nessecary to understand the imported symbols (in the first 18 lines). Therefore please look at the FGL documentation.

```
module Main where

import Graph ( mkUGraph, embed, delNodes
```

```
             , newNodes
             , nodes, edges
             , Graph, Node, UGraph
             , Context
             , pre, suc, neighbors
             , match
             )
import Indep (indep) {- indep calculates
              biggest independant set -}
import DottyVis (view)
import TestEAI (testEAI
               , CollapsingTopicData
                 (Folder)
               )

main :: IO() -- System hook

main = (putStrLn
        . view
        . buildCollapsingTopics)
       testEAI

buildCollapsingTopics
    :: Graph CollapsingTopicData ()
    -> Graph CollapsingTopicData ()
buildCollapsingTopics g
    = if (length btc) <  minCliqueSize
      then g
      else buildCollapsingTopics
           nextIteration
    where
    nextIteration
        = collapseTransitivePartGraph
          btc g
    btc = biggestTransitiveClique g
    minCliqueSize = 3

biggestTransitiveClique
    :: Graph CollapsingTopicData b
    -> [Node]
biggestTransitiveClique g
    = (indep . complementGraph)
      g

complementGraph
    :: Graph CollapsingTopicData b
    -> UGraph
complementGraph g
    = mkUGraph
      (nodes g)
```

```
      (complementEdges g)
    where
    complementEdges g
        =  [ e |
             e <- allPossibleEdges g
           ,  notElem e (edges g)
           ]
    allPossibleEdges g
        = [ (n1,n2) |
            n1 <- (nodes g)
          , n2 <- (nodes g)
          ] -- crossproduct of nodes

collapseTransitivePartGraph
    :: [Node]
    -> Graph CollapsingTopicData ()
    -> Graph CollapsingTopicData ()
collapseTransitivePartGraph someNodes g
    = delNodes
      someNodes
      ( embed
        ( contextEdgesToAll pre
               -- precessors of someNodes
        , newTopicNode -- creates a node
        , mkTopicFolderData
          newTopicNode
          g
          someNodes -- label is Folder
        , contextEdgesToAll suc
                -- succesors of someNodes
        ) -- topicNode and it's context
        g
      ) -- embed the new topicNode in g
    where
    newTopicNode = head (newNodes 1 g)
    mkContextEdges = (zip . repeat) ()
    contextEdgesToAll selectorFun
        = mkContextEdges
          (concatMap
           (selectorFun g)
           someNodes
          ) {- edges to someNodes
               selected by selectorFun -}


mkTopicFolderData
    :: Node
    -> Graph CollapsingTopicData ()
    -> [Node]
    -> CollapsingTopicData
```

```
mkTopicFolderData topicNode g someNodes
    = Folder
      (show topicNode)
      ( map
        (\n -> ( myLabel g n
               , neighbors g n
               )
        )
        someNodes
      )

myLabel
    :: Graph CollapsingTopicData ()
    -> Node
    -> String
myLabel g n
    = show ctd
    where (_,_,ctd,_)=contextFor g n

contextFor
    :: Graph a b
    -> Node
    -> Context a b
contextFor g n
    = cf (match n g)
    where
    cf (Nothing, _)
        =  error "Something went wrong"
{- This error should never occur -}
    cf ((Just res),_)
        = res
```

4.3.3 Cluster Topics

Because the Cluster topics algorithm is less readable, only the functionality and some details will be explained here. For further details please consult the implementation.

Clusters are found by trying to separate one accessibility component into two accessibility components. Therefore first a graph structure representing *Evolving Association Index* is separated into a list of accessibility components (this can be achived with the `component` function from the `DFS` module of `FGL`). Now each accessibility component can be searched for weak bridges. Depending on how many edges are to be removed, a set of edge sets is created, with all possible combinations of edges from this component. This means, for example, that if we search for all possible single bridges, that must be removed to split the component, we have a list containing n lists of length 1, if the component has n edges – every edge must be checked. If the component falls apart in two components, we must rank the quality of these two components. Because we do not want to split off single nodes, we rate the split inverse to the size difference of these two components. For x and y as size of two clusters that were built through removal of weak bridges, the quality of the split is $|1 - x/y|$. The test with the lowest result of this ranking is the best possible collapse.

Chapter 5

Retrieving and analysing from a testbed

The initial idea was to retrieve about 200 to 400 articles from different newspapers available online, copy them and index them and this during the course of approximately six weeks. Indexing consists of building a descriptor per article and relating some descriptors through associations. A descriptor consists of publication date, original URL, URL of the copy, publication date, title of the article and the site this article came from.

It was time consuming to retrieve, read and index the articles, wich lead to a small set of documents in the end.

5.1 The first test serie

For the first test series German language newspapers from Germany, Austria and Swizerland were chosen. Random selection of the articles was used in the beginning. This strategy was unsatisfactory, because not enough connections between the articles were found. It turned out that in relation to the amount of articles published in 5 different newspapers, 54 selected articles in this series were not significant. Therefore a big variety of completely unrelated topics were involved, and as a result a lot of documents remained unrelated. This problem may not have occured if many more documents from perhaps less online newspapers would have been retrieved .

The following tables summarise some statistical information about the first testbed. The articles where collected from the German newspapers online versions from *Süddeutsche Zeitung* (**SZ**), *Frankfurter Allgemeine Zeitung* (**FAZ**[1]), *Die Zeit* (**ZEIT**), the Austrian paper *Der Standart* (**STD**) and the in German published Swiss newspaper *Neue Züricher Zeitung* (**NZZ**).

[1] at this time only the english version of this newspaper was available online but now as last big german newspaper FAZ is also available online.

Collection date	13.8.2000-5.9.2000
Total number of articles	54
Articles from SZ	38
Articles from NZZ	5
Articles from FAZ	4
Articles from STD	4
Articles from ZEIT	3

In the associations table two values are provided: the first number corresponds to the number of associations after deriving new associations according to the properties of the special associations. This means for example that if the *Evolving Association Index* author enters the association A SPECIALIZES B, B GENERALIZES A is derived. The second number in brackets corresponds to the number of associations provided by the *Evolving Association Index* author.

Sum of all associations	33 (26)
ISRELATEDTO associations	26 (24)
CITES associations	0 (0)
GENERALIZES associations	3 (1)
SPECIALIZES associations	3 (0)
EQUIVALENT associations	1 (1)

As we can see no CITES associations were found. This is due to the semantics of the CITES relation at the time of the collection of the first testbed: only articles cited in the text of the article were meant to be real citations, not hyperlinks after the text. These were regarded as navigation links and therefore not considered. Later it turned out, that at the time the online newspapers were only extracts of the paper edition and therefore no hyperlink citation was used. The other associations were not provided by the journalists – they were deduced while reading the articles.

The following table summarizes the number and size of relation centers found. The size is the in- plus out-degree of the node and therefore the number of documents standing in ISRELATEDTO relation centers. The other relations CITES , GENERALIZES and SPECIALIZES do not spawn relation centers, but hubs and authorities. Due to no CITES and almost no SPECIALIZES and GENERALIZES associations these are not accounted here.

relation center size	**count**
4	1
3	3
2	3
1	14

As we can see the first test series did not provide very extensive relation centers, but the four biggest of them are built by half of the associations available.

The next table summarizes the information about the accessibility components of this testbed. The size of a component is the number of documents involved in this component.

Components found	8
Biggest component size	7
Smallest component size	2
Average component size	3.625
Unlinked documents	23

The content of the components is related to each other, which is not surprising due to the fact that they are so small. Therefore the use of Cluster Topics and Collapsing Topics does not make sense and is omitted in this small testbed.

5.2 The second test serie

The bad connectivity and implementation problems lead to a stop and a second series was started later. The goal for this series was to find strongly related structures, and therefore the documents collected were constrained to two actual topics[2] making headlines in the daily press.

- BSE, the "mad cow disease", mouth and foot disease and other articles concerning health problems derived from food production.
- MP3 related news. Napster and MP3.com were at the time in lawsuit. The mayor players in the music industry felt attacked by the Internet as a new media market.

The CITES relation was extended in its interpretation to contain any hyperlink related to this topic (this excluded navigation related links on sidebars). One of the maingoals while retrieving the documents was to follow the links as extensively as possible without leaving the formerly described informal "topic of interest" definition. This was obtained by breadth searching the cited documents – this was not automated because of the necessity to read the articles to decide if they were related or not. In fact it turned out that most of the articles were retrieved due to their citation relation to other documents, and not due to the fact that they were published at the time of the second test series. Therefore not only actual newsarticles, but also older ones got collected. In fact most of the associations in this second series are CITES associations.

The new goal was to discover new structures in a more complicated association network where simple accessibility components are too big for useful navigation help. In the end the dense network structure of the second series lead to the idea of "Collapsing Topics" and "Cluster Topics".

The following tables are similar to the ones of the first test series, but some more are added to summarize the results of "Collapsing Topics" and "Cluster Topics". The online news providers changed a little bit. Due to the fact, that articles of a computer related topic were searched, the German online computer newstickers heise.de and computerwoche.de were added. Heise releases the magazines C't, iX and telepolis (all on paper) which are quite popular (for the non academic society) in Germany. Computerwoche is an economic and financial newspaper related to IT business. The publisher of Computerwoche also publishes Young Professional – this newspapers is meant for the IT job market.

[2]Topics this time are not meant to be structures in an association network, but topics in the eyes of the newspapers and the readers (this is often called a campain in terms of journalism)

Both, Heise and Computerwoche maintain an online newsticker. This service provides much articles than the magazines, but are therefore very up to date. Furthermore, the news is often enriched with hyperlinks to related documents (not only links for navigation or marketing purpose) which is a rich source for CITES associations.

Some of the newspapers of the first test series were dropped due to complicated reimplementations of their websites. The problems were often related to non standard compliant URLs (browsers tend to be more tolerant than the tools used to copy the web pages) or JavaScript.

The articles were collected from *Süddeutsche Zeitung* (**SZ**), *Die Zeit* (**ZEIT**), *Neue Züricher Zeitung* (**NZZ**), *die tageszeitung* (**taz**),*www.computerwoche.de* (**CW**) and *www.heise.de* (**heise**).

The collection span is the timespan in which the articles were retrieved, while the publication span is the time span in which the collected articles were published. Because hyperlinks between related articles were followed, these two spans differ.

Collection span	21.2.2001-9.4.2001
Publication span	12.2.1998-9.4.2001
Total number of articles	220
Articles from SZ	102
Articles from NZZ	4
Articles from taz	2
Articles from CW	13
Articles from heise	94
Obviously broken	5

The associations table is coded like the previous one: the first number corresponds to the number of associations after deriving new associations according to the properties of the special associations, the second number in brackets corresponds to the number of associations provided by the *Evolving Association Index* author.

Sum of all associations	307 (298)
ISRELATEDTO associations	13 (8)
CITES associations	286 (286)
GENERALIZES associations	3 (1)
SPECIALIZES associations	4 (2)
EQUIVALENT associations	1 (1)

5.2.1 Hubs and authorities

The latter tables summarize statistics about hubs and authorities.

authority size	**count**
16	1
7	1
4	5
3	12
2	40
1	113

hub size	**count**
23	1
9	2
8	1
6	2
5	3
4	15
3	22
2	23
1	33

While analysing the hubs and authorities it turned out, that these node properties were well separated. This means, that hubs were never authorities at the same time and vice versa.

For more details about the components it is useful to look at a visualization of the net structure. In contrast to the first test series there are two tables, one for big components and another for small ones. Small components are trivial graphs, like two or three documents linked together in a chain or simply one hub linking to some other documents.

Small components found	9
Biggest small component size	5
Smallest small component size	2
Average small component size	2.67
Big components found	5
Biggest big component size	61
Smallest big component size	13
Average big component size	35.8
Unlinked documents	18

5.2.2 "Cluster Topics" of the 2nd testbed

The second testbed provided big enough accessibility components to apply the "Cluster Topics" algorithm. Please note: "Cluster Topics" was meant to find weak bridges in accessibility components that can connect semantically unrelated topics. The constraints on data collection for the second test serie were to retrieve only documents on some recent campaigns – MP3, BSE and mouth and foot disease. This already provided an efficient filter for these kind of bridges, because semantically unrelated (but maybe cited) documents were not retrieved. The two campaigns "MP3" and "food" miraculously did not provide inter topic bridges. Less obvious: "mouth and foot disease" and "BSE" did also not provide any CITES relation although both are related to contaminated meat, mass slaughtering of animals and also occured at about the same time. Therefore "Cluster Topics" could not proof it's reliability in dividing unrelated topics.

Nevertheless bridges were found but their meaning was less obvious. My interpretations of these bridges is not necessarily valid for bridges in general.

The bridge between 984422231 and 984422792

This section with a rather mystic (or maybe only cryptic) name is about one special bridge found in the testbed. It connects the documents with the IDs 984422231 and 984422792. These documents are connected through a bridge building a cluster of 33 documents. The removal of this bridge produces two clusters, one of 13 documents and the other of 20 documents.

- **984422231**, *Universal und Sony machen Napster Konkurrenz* (heise.de, 22.2.2001)
  - Cites
    - *Hintergrund: Das Establishment frisst den Revoluzzer* (heise.de, 31.10.2000)
    - *Napster will gegen Verfügung in Berufung gehen* (heise.de, 13.2.2001)
  - Cited by
    - *Napster-Prozess geht in die nächste Runde* (heise.de, 24.2.2000)
    - *Realnetworks plant Musik-Aboservice* (heise.de, 30.3.2001)
    - *Napster: Neue Perspektiven durch EMI, Bertelsmann und Warner* (heise.de, 2.4.2001)
    - *Bertelsmann: Ein Durchbruch für Napster* (heise.de, 3.4.2001)
    - *Napster mit Alanis Morissette gegen die Plattenlabels* (heise.de, 4.4.2001)
- **984422792**, *Napster-Prozess geht in die nächste Runde* (heise.de, 24.02.2000)
  - Cites
    - *Brüchiger Friede zwischen MP3.com und Plattenlabels* (heise.de, 18.11.2000)
    - *Napster muss nicht (vollständig) vom Netz* (heise.de, 12.2.2001)
    - *Napster bietet Musikindustrie eine Milliarde Dollar* (heise.de, 21.2.2001)
    - *Die Milliardenwette der Musiktauschbörse* (heise.de, 22.2.2001)
    - *Universal und Sony machen Napster Konkurrenz* (heise.de, 22.2.2001)
  - Cited by
    - *Napster blockiert nicht-lizenzierte Musikstücke* (heise.de, 2.3.2001)

The translation of these headlines is not important at this point and will follow later. For detailed informations about these documents environment please consult the HTML *Evolving Association Index* browser, the visualization or the print out in the appendix.

Reading of the artiles in detail brought up some differences about the focus of the documents in the two clusters joined by the bridge: The environment of "*Napster-Prozess geht in die nächste Runde*" in it's cluster provides precise information on the lawsuits between companies relating to new technologie/Internet music on the one hand and the traditional music industry on the other hand. Announcements of the related companies are written straight forward, surrounded with other important events that occured recently. Therefore this cluster seems to represent the actions that took place.

The cluster containing "*Universal und Sony machen Napster Konkurrenz*", or and the environment of this article is mostly related to reflections about the events of its publication date. The documents emphasize on the conflicts between the established music industry and competitive companies in new market positions. The articles tend to predict more than the documents in the other cluster. This cluster is slightly more focused on the matters of the established music industry.

5.2.3 "Collapsing Topics" of the 2nd testbed

As mentioned earlier, while introducing collapsing topics, the algorithm is NP-complete. To be able to calculate collapsing topics it is nessecary to reduce the number of nodes[3]. Without influencing the result, this can be done by applying the algorithm on the separated accessibility components, excluding the other nodes.

"Collapsing Topics" was first applied on the biggest accessibility component (61 articles)[4]. It had to do with MP3. Opposed to "Cluster Topics", the results seem to be easier to analyse. The collapse was iterated as long as possible, producing some trees of clusters and documents. Already the titles of the documents shows that clusters represent a specialization of topics: the Napster and the MP3.com cases are clearly separated.

- ⊙ **985030666**, *MP3.com mit neuem Service für Musiker*
- ⊙ **985030880**, *My.MP3.com ist wieder online - mit Jahresgebühr*
- ⊙ **Folder 1**
  - **983463861**, *Musikindustrie gewinnt gegen MP3.com*
  - **983463680**, *MP3.com sucht Einigung mit Plattenlabels*
  - **Folder 2**
    - **983305401**, *MP3.com kündigt Einigung mit EMI an*
    - **Folder 3**
      - **983305469**, *Richter: Prozess gegen MP3.com ist ein klarer Fall*
      - **983305633**, *MP3.com zahlt 40 Mio. Dollar an Musikkonzerne*
      - **983304586**, *Hintergrund: Plattenlabels vs. Internet - die Revolution findet nicht statt*
    - **Folder 4**
      - **985090440**, *MP3.com-Chef verneint Vorsatz*
      - **985090361**, *MP3.com feiert Zwischen-Entscheidung*

[3] nodes are articles

[4] The implementation behaved useable in time and memory consumption – the haskell prototype took about 1 minute on a typical desktop computer

* **Folder 5**
  - **985081053**, *MP3.com einigt sich mit Sonys Plattenlabel*
  - **985082199**, *Universal will von MP3.com 450 Millionen US-Dollar*
  - **985081979**, *MP3.com zu Schadensersatz in Millionen-Höhe verurteilt*
  - **985030775**, *Internet-Musikdienst MP3.com schließt Frieden mit Universal*

The former tree is the biggest collapsed topic found in the analysed accessibility component. This collapsed topic consists of documents and further collapsed topics. After recognizing the whole structure as a semantically well formed topic[5], the quality of the tree structure was analysed. When looking at the first level of the tree we can recognize, that the lawsuit is completely enclosed in the first folder, while the two documents 985030666 and 985030880 are related to other informations about MP3.com. To get a better view on the topic, the folders where enriched with titles and publication date oriented information – every folder was enriched by a timespan of all publication dates of the contained articles (and timespans of contained folders). The date information was used to sort the content of the folders as far as possible. As far as possible means, that if a documents publication date falls inside the span of a sibling folder, it is not sorted (this is shown by putting the publication date in paranthesis). This was done manually although the temporal aspects may have been done automatically. The automatic generation of titles for the folders is a problem out of the scope of this report. It may either be the title of the most important document, or a automatically generated title based on some important words found in the articles headings. This is a computer linguistical problem and will not be solved here[6]. This is the result of this process:

[**28.4.2000 - 15.11.2000**], `MP3.com`

⊙ [**28.4.2000 - 15.11.2000**], `MP3.com vs.the music industry`

- **28.04.2000**, *Musikindustrie gewinnt gegen MP3.com*
- **02.05.2000**, *MP3.com sucht Einigung mit Plattenlabels*
- [**05.5.2000 - 15.11.2000**], `lawsuit looks bad for MP3.com but friendly agreement occurs`
  - **(28.07.2000)**, *MP3.com kündigt Einigung mit EMI an*
  - [**05.5.2000 - 15.11.2000**], `lawsuit turns out bad for MP3.com but friendly agreement occurs`
    * **5.05.2000**, *Richter: Prozess gegen MP3.com ist ein klarer Fall*
    * **12.06.2000**, *MP3.com zahlt 40 Mio. Dollar an Musikkonzerne*

[5] all documents are clearly related to MP3.com, their service and their lawsuite

[6] Actually the "Second International Natural Language Generation Conference (INLG2002) is being planned – see htp://www.research.att.com/ rambow/inlg/first-announcement-01.html. A very extensive bibliography with about 1300 Publications may be found at http://liinwww.ira.uka.de/bibliography/Ai/nlg.html

    * **29.07.2000**, *Hintergrund: Plattenlabels vs. Internet - die Revolution findet nicht statt*
  - **[01.8.2000 - 15.11.2000]**, `More details about the case.`
    * **01.08.2000**, *MP3.com feiert Zwischen-Entscheidung*
    * **(29.08.2000)**, *MP3.com-Chef verneint Vorsatz*
    * **[22.8.2000 - 15.11.2000]**, `clearing of the MP3.com case`
      · **22.08.2000**, *MP3.com einigt sich mit Sonys Plattenlabel*
      · **06.09.2000**, *Universal will von MP3.com 450 Millionen US-Dollar*
      · **07.09.2000**, *MP3.com zu Schadensersatz in Millionen-Höhe verurteilt*
      · **15.11.2000**, *Internet-Musikdienst MP3.com schließt Frieden mit Universal*

⊙ **05.12.2000**, *My.MP3.com ist wieder online - mit Jahresgebühr*
⊙ **30.01.2001**, *MP3.com mit neuem Service für Musiker*

This tree structure seems to represent well what happend in the complete cluster, while the depth of the tree indicates the level of specialisation on details. This could for example be used for an automatic generation of summaries.

Automatic generated synopsis

The following method of generating a synopsis assumes that we have the ability to automatically generate sentences like the folder titles. Furthermore we may need a method to automatically generate the "glue" between facts and simple sentences to get a readable text. The last step should be a temporal modification of the sentences, so that they all fit together grammatically.

To generate a synopsis of a topic for example, it may be useful to produce a kind of "breadth first" serialisation of the tree. This provides the possibility to read the synopsis as far as desired without missing any facets of the topic, but also without being bothered by too much detail. Furthermore the publication date of an article is compared to its sister folders. If this date sits inside the time span of a folder, the article can be omitted. The idea is, that this article may already be represented by the folder. Although nothing proves this, for a synopsis about a topic this heuristic is useful.

The following synopsis is generated manually, but according to former description. The titles were only translated straight forward and rectified at some place to fit the same time. Only a few words were added, therefore the flow of the text seems unpleasant.

> ***Synopsis.*** *These are documents about MP3.com from 28.4.2000 to 30.01.2001. Its about MP3.com vs.the music industry [folder 1], their comeback with annual fee [985030880] and their new service for musicians [985030666]. In the lawsuit "MP3.com vs.the music industry [folder 1]" first the music industry wins against MP3.com [983463861]. Afterwards MP3.com looks for a friendly agreement*

with the music industry [983463680]. The lawsuit turned out bad for MP3.com, friendly aggreements occured [folder 2,folder 3]. After the judge explained that the case against MP3.com was quite clear [983305469], MP3.com payed 40 mio. $ dollars to the music industry [983305633]. After that, it became clear, that the big revolution from traditional music industry to internet industry would not happen [983304586]. The case in detail [folder 4]: First MP3.com celebrated that they had to pay less than first accused to [985090361]. Afterwards they assured that they did not want to act maliciously [985090440]. Then the MP3.com case was cleared [folder 5]: MP3.com found friendly agreements with Sony [985081053]. Universal wanted to have 450 million dollars from MP3.com [985082199]. MP3.com is then accused to pay a fee of some million dollars [985081979]. Afterwards found friendly agreements with Universal [985030775].

Chapter 6

Integration of *Evolving Association Index*

The creation of indices for a large amount of data has to be tackled by more than one authors. Therefore easy integration of *Evolving Association Index* is important. Integration in applications is not meant here – therefore literature about XLink is useful, because *Evolving Association Index* is XLink compliant. Integration of many instances of *Evolving Association Index* is meant here – that is the merging into one big document.

Please note: An *Evolving Association Index* is a network (or graph). Therefore it consists of nodes (descriptors) and relations between nodes (associations). Therefore the result of putting together several *Evolving Association Index* instances, should at least provide the mentioned properties. This can be accomplished quite easily: Let $E_1 = (V_1, E_1), \ldots, E_n = (V_n, E_n)$ be some *Evolving Association Index* instances, with V_x as set of descriptors and E_x as set of associations, then the integration E_{all} can be defined as $E_{all} = ((V_1 \cup \ldots \cup V_n), (E_1 \cup \ldots \cup E_n))$. This results in a new tuple of descriptors and associations – and is therefore syntactically a valid *Evolving Association Index*. Now the semantic quality of this structure has to be analysed.

Obviously merging indices with disjunct descriptors is not problematic. Before merging them, there was no relationship between the descriptors of the different indices and this property is retained after merging. On the other hand now it is possible to merge the topics by adding associations between nodes that were in different indices before the merge. When topics are merged, it is useful to be aware of some topic structures like hubs, authorities, collapsing topics and cluster trees – the insertion of associations affects these structures. Therefore it is useful to analyse in wich way the different associations can be used to achive well merged topics.

If the topics merged do not have completely disjunct descriptors, this means that there are equivalent descriptors. In this case it is useful to insert an EQUIVALENT association between equivalent descriptors. How this takes place and what it implies will be described later. Another way would be to reduce the two descriptors to one, but this would imply an information loss – it would not be possible to separate the merged indices again without losing information.

6.1 The use of isRelatedTo for merging topics

The isRelatedTo association is the weakest way to merge two topics. This weak association will always affect accessibility components. This means, that two accessibility components can be related to each other by simply creating an isRelatedTo association between two nodes – one in each accessibility component. This is a bridge that may be broken by cluster topics for example. If the *Evolving Association Index* author is aware of the existance of collapsing topics, he may use many isRelatedTo associations[1] to insert nodes into existing collapses. Therefore it may be nessesary to create a lot of new isRelatedTo associations because all nodes of one collapse are connected transitively to each other. This property has to be retained to merge collapsed topics. This means, that the number of associations grow almost exponentially according to the amount of nodes in a folder.

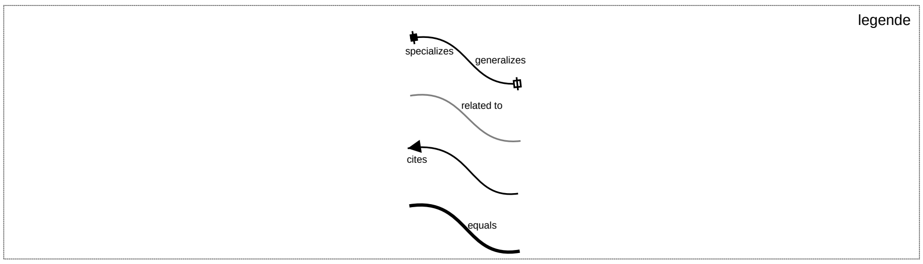
6.2 The use of cites for merging topics

Obviously the cites association can not be used to merge topics actively. This is because the cites relation is providet by the data author and not by the *Evolving Association Index* author. If one document cites another, the association must be inserted, but the *Evolving Association Index* author can not "invent" new cites associations to merge topics.

6.3 The use of equivalent for merging topics

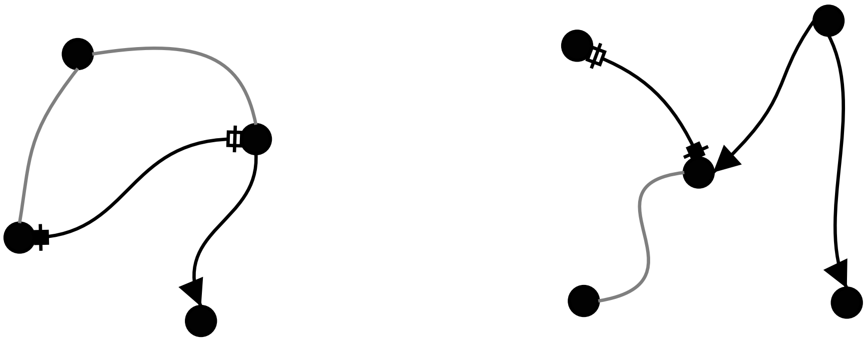
The equivalent association is especially useful for merging hubs or authorities. Because citations cannot be equalized, the hubs or authorities are transformed in some kind of relation centers. This is due to the implication of isRelatedTo through cites . Note, that the equivalent association results in equalizing the associations of two nodes. Therefore two hubs that are related through an equivalent association share the same specializes , generalizes and isRelatedTo associations.

6.3.1 Example

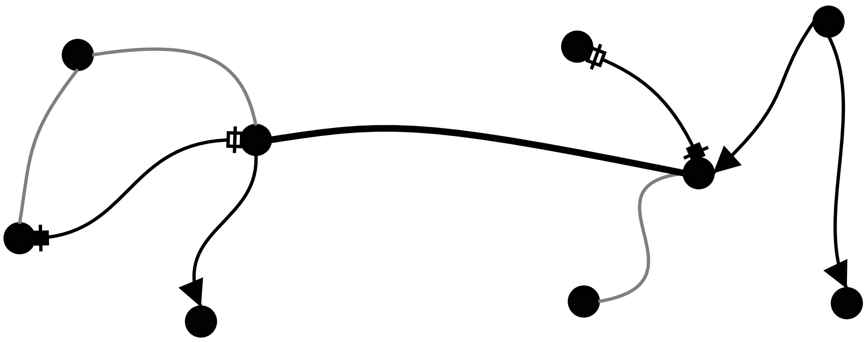


The following illustrations describes this:

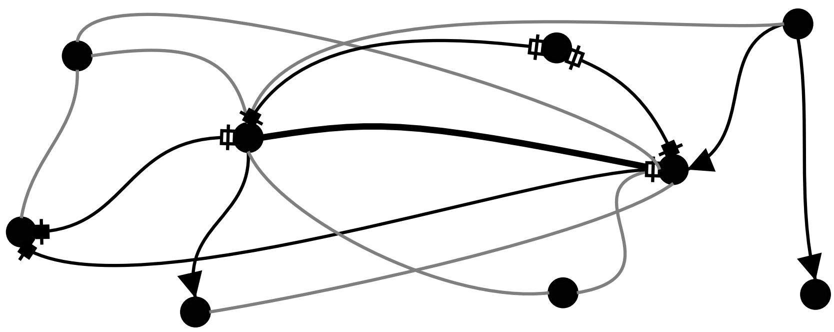
[1] This could also be done with more convenient isRelatedTo associations with higher arity – this would mean, that a lot of nodes can be associated by only one association



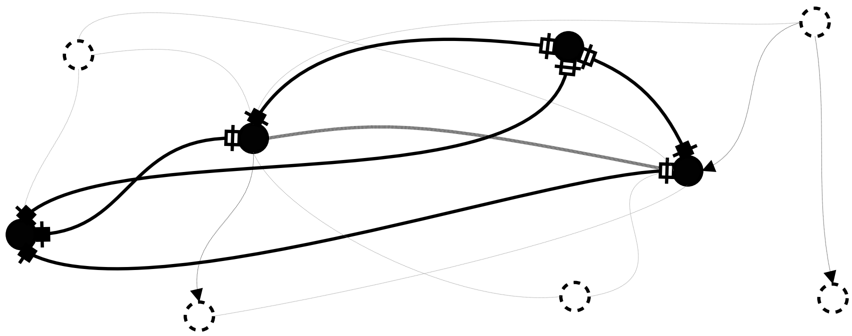
The first illustration shows two unrelated accessibility components. These may come from two different indices that were integrated into one.



The second picture shows that an EQUIVALENT relation was included. This is done by the author integrating two indices. He finds two documents with equivalent content being suitable for topic merge.



In the third illustration we see, that the associations are expanded according to the properties of EQUIVALENT .



In the end associations not directly mentioned in the new EQUIVALENT association may be spawned due to the properties of other associations.

6.4 The use of generalizes /specializes for merging topics

As seen earlier some algorithms for topic structures have bad computational complexity. For normal application heuristics reducing the factors influencing this complexity are useful. A good idea would be to keep the network small. On the other hand integration of indices is exactly the opposite of keeping the network small! What is needed to overcome this problem are ways to rate descriptors as more or less irrelevant to topic calculation. A possible hint for approximation of good topic structures could be to drop very specialized information. If a topic contains a general node, the specialized documents are likely to also be part of this topic structure. After the calculation of the topic structures the specialized documents may be included again without significant disturbance of the topics. Extensive use of GENERALIZES /SPECIALIZES is needed to support this heuristic. Therefore GENERALIZES /SPECIALIZES should be used as often as possible while integrating several indices.

Chapter 7

Related Work

Not much literature was available providing highly relevant information for the work on *Evolving Association Index*. Most of it was related to the connectivity analysis of hyperlinks.

7.1 The Web as a graph: measurement and models and methods[7]

The Web is seen as a graph composed of websides as nodes and hyperlinks as edges. Two algorithms are presented to automatically extract topics. The first algorithm (it's called HITS) is a search algorithm that distils high-quality pages for a topic query. The second algorithm is "thawling the web for cyber-communities" which means it distills topics. Therefore these two algorithms fit quite well together – one finds topics while the other finds the most relevant pages in that topic. The HITS algorithm works by finding and ranking hubs and authorities. The ranking of hubs is based on the ranking of authorities and vice versa[1] The thawling algorithm is based on ideas found in the domain of citation analysis. Subgraphs of the Web with special characteristics[2] are searched. In the end properties for new graph models suited to the properties of the Web are presented. The purpose of these models is to enlarge the gap between worst case and average case complexity for those (and other web-graph based) algorithms. The base for these new models are statistical measurements of the web-graph.

7.2 Mining the Link Structure of the World Wide Web[4]

Here, like in [7] HITS and "thawling" are presented. HITS is presented in more details and some extensions of HITS found in the "clever" project are described. These extensions are proposed to overcome some shortcomings of HITS. The first addresses the tendency to broaden very narrow topics. Therefore the text

[1] for example, a authority pointed to by a high ranked hub will also get good ranking – therefore ranking of hubs and authorities is mutually recursive.

[2] The Web is searched for bipatite cliques $K_{i,j}$, where all nodes of i are linking to all nodes of j

contained in links – the anchor text – is analysed. This is done by traditional textual content analysis. Perhaps this problem could be solvedby the use of "Collapsing Topics" without textual analysis. Another problem is called topic drift and is based on the fact that sometimes hubs may point to different topics. This problem is addressed by splitting hub pages in smaller subsections. It is possible, that this content analysis may become obsolete, if the results of HITS are refined by an algorithm similar to "Cluster Trees". The last problem is, that very popular big sites sometimes may take over topics: Imagine searching sites about doors and windows for house construction. It is likely that the term "windows" will enforce Microsoft related content to obfuscate the result. This is simply overcome through user supplied 'stop-sites'. Maybe the typed links of *Evolving Association Index* could turn out useful to automatically accomplish this task.

7.3 Authorative Sources in a Hyperlinked Environment[6]

As one of the earliest articles about link analysis of the WWW published by Kleinberg, this is a very good introduction to the vision of hub and authority based search of relevant information. The strategy described here is quite similar to the HITS-algorithm: The graph used to calculate the most authorative pages consists of the search result of a traditional search engine, extended by the pages linked to this initial set. The calculation is an iterative process where the ranking of authorities is calculated through the ranking of the hubs pointing to them and vice versa. The conclusion of this article is, that this algorithm works well for broad topics but tends to generalize very special search topics.

7.4 Finding Related Pages in the World Wide Web[5]

In contrast to [4] and [7] no concrete set of documents based on a traditional search query is used as seed for topic calculation, but a URL pointing to a document. Two algorithms (one is called "Companion") very similar to HITS and the cocitation found in [7] are used. The results are compared to the results of Netscapes "What's Related" service. They seem to provide better results, although only based on connectivity information opposed to netscape's service. Netscape's service is providet by Alexa [3] and combines content and connectivity analysis.

7.5 A Web Community Chart for Navigating Related Communities[12]

This short article describes a technique to create a community chart. A community chart is a graph of communities where related communities are connected.

[3] see http:/www.alexa.com

Communities consist of related documents. The explicit creation of relations between communities is different to the technique from the algorithms of [4] and [7]. Furthermore this article introduces a new relation besides the widespread hyperlink relation: the REO relationship – the Respect-Each-Other relation. This is imposed, if two pages are related to each other using a related pages algorithm called Companion. Companion is introduced in [5].

7.6 Hypersearching the Web[3]

The Clever project is another variant of the HITS algorithm and Googles random surfer model: again hubs and authorities are computed, but this time based on a seed set of Web documents (e.g. the result of querying a text based search engine) is used to get a portion of the web before applying hub and authority ranking. The seed set is extended by all referring pages (other pages in the WWW pointing to documents of this set) and all referred pages (pages outside this set which are hyperlinked by documents of this set). The advantage is, that hub and authority calculation takes place on a smaller set of documents which are already slightly specialized in terms of the query. The article gives a good overview of the Clever project without being technical. References to further relevant resources about finding authorities in a hyperlinked environment are provided and the influence of citation analysis and bibliometrics is explained.

7.7 Open Peer-Review as Web's Self-Organization Force[8]

In contrast to the fully automatic ranking and classification algorithms described earlier, the open peer-review is partly human driven distributed collaborative Web categorization. Users of the system are providing votes to rate and classify the Web. The self organization described here is based on ideas found in the open source movement. The voting model provides the ability to vote, give the vote to another trusted voter or waste the vote removing its influence from the system. Interesting is the fact, that in a certain way it is a kind of evolving association index: users can provide some kind of "related to" associations between documents and topics - therefore relating all documents of a topic to a former document. Furthermore hyperlinks (aka. citations) can be annotated and therefore typed. The article also describes a generic architecture able to translate for example Web directories like Yahoo! or DMOZ[4] into votes. Also pageranking could be translated into votes to enrich the system with another non directory based view of the Web.

7.8 Metaphors We Surf the Web By[9]

To be able to search interesting topic structures that could help a web user in finding information, it is interesting to know how the web is seen by users. This article analysed how people talk about their experiences in using the WWW. The way they remember what they were doing is quite close to their view of

[4] the open directory project, see http://www.dmoz.org

the web. An interesting fact is that the web is seen quite physical. Despite the fact, that information is queried and is transferred to the user, they see the web as a space in wich they move – not the information is coming to them, they are going there. This view of the web is used by novice and experienced users as well. Furthermore the analysed browsing experiences of the users showed that most of the time they actively move around towards information. Less often the user is viewed as passive recipient of information or as "passenger in some web vehicle". This suggests that the web lacks agency. The conclusion is that new browsing utilities should be aware of the users physical space metaphors of the web. Advantage can be provided by finding the differences between cyberspace and real space and then using those differences to make the boundary apparent.

7.9 Description of Collections and Encyclopaedias on the Web using XML[10]

This article was a big inspiration for finding useful association types. It describes the effort of modelling a distributed decentralized and collaborative catalogue for artwork. Different organisations and researchers are intended to build, use and extend this catalogue. It turned out impossible to find a general classification scheme for artwork providing enough acceptance by all involved members. This is due to the fact, that different cultural organizations prefer to develop their own taxonomy tailored to their own needs. The consequence is a "mapping" scheme able to map taxonomies against each other. It is called concept-mapping. Concept-mapping is a relation which exists between two classes. This relation is modelled through the fixed relations "equivalence", "narrower", "broader", "part-off" and "whole". Furthermore user defined relations with more specific semantic can be introduced. The realization of this catalogue is done using XML, and the DTDs can be found in the appendix (of [10]).

7.10 The Connectivity Server: fast access to linkage information on the Web[1]

For the implementation of algorithms based on the WWWs graph structure of hyperlinks – hyperlinks are in fact CITES associations – we require efficient storage of the graph. This paper describes the implementation of a connectivity graph server of all webpages indexed by altavista. The implementation is based on adjacency lists (representing the pages referred by the actual page), inverted adjacency lists (representing the pages linking to the actual page), delta encoded text files (used for a compressed mapping of URLs to shorter IDs and vice versa) and auxiliary data structures for modifications. The primary data structures are quite rigid and cannot be modified, therefore the auxiliary data structures represent the changes until the primary structures are regenerated. This connectivity information may be accessed for experiments with the graph structure of the WWW. The conclusion is, that it is possible to provide linkage information for a significant portion of the Web at a fairly high speed.

7.11 Introduction to WordNet: An On-line Lexical Database[11]

To understand the ideas on wich thesauri are based, it is useful to read this report. It is also an excellent example for an index based on relations. WordNet is a lexicon, where words are not organized in lexical order, but according to psycholinguistical theories respecting human associations with words. WordNet is based on four different semantically varying associations: synonymy, antonymy, hyponomy and meronymy. Synonymy means to associate words that have the same meaning in a certain context. Antonymy means to associate words with opposite semantic like "rise/fall". Hyponymy corresponds to the "is-a" relation in object-oriented knowledge representation and meronymy the "has-a" relation. This report also points out, that the search for the "roots" – the topmost levels of the hierarchies, is a complicated process. Therefore it may be useful to find *Evolving Association Index* techniques applicable for the creation of thesauri like WordNet. The WordNet database can be queried online and is available to the public at `http://www.cogsci.princeton.edu/ wn/`. It is a very useful tool for traditional text analysis and knowledge retrieval.

7.12 The Maximum Clique Problem[2]

The Collapsing Topics structure turned out to be an instance of the maximum clique problem. Because the calculation of the maximum clique is very wide spread it is useful to get an overview about this topic. This paper is a good introduction with different problem formulations of the maximum clique problem. Furthermore different implementations of exact methods and heuristics reducing the complexity are presented here. In the end some interesting applications of the maximum clique problem are shown. There is a big bibliography about related publications appended.

Chapter 8

Conclusions

Evolving Association Index is a useful strategy for classification of data items with unpredictable classification. It is possible to sort information according to their relevance[1], or to automatically generate hierarchies of specialization[2] without the notion of these structures through the author. The ways of indexing is either implicit (through CITES and the temporal associations) wich means that the index author is not responsible for these informations, or explicit (through the other associations) actively maintained by the author. Therefore the index is not the result of the work of one party, but three different instances are involved in the process of indexing – the index author, the information author and the time. This results in a healthy mixture of opinions about the relationship of information. Note, that if the information author would provide the associations, the opinion of other authors would provide even more diverity of opinions. Therefore *Evolving Association Index* is robust against mistakes of individuals.

Evolving Association Index is not well suited to unrelatable information like addressbooks. Furthermore information with known classification do not benefit from *Evolving Association Index*. This is mainly due because it takes some time for the topic structures to appear. Furthernore the topics or classifications are anonymous. This is a disadvantage, when the categories are known and immutable.

In this report implicit indexing was mainly analysed. This is because these associations were already given. Furthermore there was no other index worth while to integrate, which could have resulted in many more explicit associations. The new results of this analysis were cluster topics and collapsing topics. Cluster topics seem to provide hints about the authors environment or the authors focus on a topic. Collapsing topics provide tree structures representing some kind of specialization. Hubs and authorities turned out to indicate relevant documents or overview documents of a topic. All these informations were found without any content analysis, only using the graph structure.

Although not analysed here, explicit indexing seems to be a useful way to integrate several indices together. It is a way to annotate knowledge about relationship between information.

The temporal associations were not analysed, because the test series were not tailored to this analysis. It seems, that bigger testbeds with more similar topics

[1]for example through search for hubs/authorities/relation centers

[2]by using Collapsing Topics or directly through SPECIALIZES /GENERALIZES associations

are useful for this kind of work. On the other hand, many possible applications of *Evolving Association Index* that do not need temporal associations also exist.

Evolving Association Index is applicable to a lot of different domains. Descriptors and association names need marginal changes to be adapted to new domains. As an example music will be used for such a new domain.

8.1 Indexing music as possible application of *Evolving Association Index*

Evolving Association Index is perhaps better suited for categorizing music than fixed categories. This is because music is not well categorizable. On the one hand musicians do not like their artwork to be categorized and new music is normally new because it is not categorizable in existing categories. On the other hand the listener is often not an expert in music categories and does not know how to classify music – in the end it does not matter as long as he likes it. To classify music through *Evolving Association Index*, information items only have to be related to each other. This is normally not too complicated, because there is no need to unambiguously assign the information item to a category. One could for example relate two songs, because the listener likes to hear them when he is in a certain mood, or because they sound the same when he hears them. These categorizations are extremely anonymous, but tend to be the base of traditional categorization.

8.1.1 Descriptor proposal for music

Useful information items in the domain of music could be:

- Persons
  - Artist
  - Musicians
  - Band
  - Orchestra
  - Conductor
  - Composer
- Music
  - Song
  - Title
  - Composition
  - Edition
- Containers
  - Album
  - Compilation
  - Epoch

Each of the items may be an indexable information item represented by a suitable descriptor. They may all be associated to each other.

8.1.2 Association proposal for music

The associations proposed here have the same properties as those for indexing news articles, but some have other names or meanings:

- **relatedTo:** Two music items may be related to in a number of ways – either traditional relations like composer/composition, or psychological relations like "these songs sound similar" make sense.
- **cites:** In music "citation" is a well known aspect, wide spread through jazz music. Jazz is mostly based on citation and improvisation. Actual pop music also uses citation extensively.
- **equivalent:** Equivalence may be used for one artist using different names. It may also be used for various recordings, remixes, interpretations of the same song/composition.
- **partOf/wholeOf:** This association shares the same properties as the SPECIALIZES /GENERALIZES associations. It may either be used for an album/title association or artist/composition relation. Other use like "acapella version/with instrumentation" may also be useful – the choice of meaningful application is descidet by the author.

In the end *Evolving Association Index* is neither a closed domain nor complete application, nor a polished theory. It could be classified as a collection of ideas and vocabulary for knowledge discovery based on associations between information items. Therefore the Pageranking methods described in[7] and in[3] as well as the "Description of Collections and Encyclopaedias on the Web using XML"[10] could be formalized with the vocabulary of *Evolving Association Index*. The advantage would be a kind of simple integration of theories based on associations.

Bibliography

[1] Krishna Bharat, Andrei Broder, Monika Henzinger, Puneet Kumar, and Suresh Venkatasubramanian. The connectivity server: fast access to linkage information on the web. *Computer Networks and ISDN Systems*, 1998.

[2] Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. The maximum clique problem. Handbook of Combinatorial Optimization, 1999.

[3] Soumen Chakrabarti, Bryon Dom, S.Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, Jon M. Kleinberg, and David Gibson. Hypersearching the web. *Scientific American*, 1999.

[4] Soumen Chakrabarti, Byron E. Dom, David Gibson, Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Mining the link structure of the world wide web. *???*, 1999.

[5] Jeffrey Dean and Monika R. Henzinger. Finding related pages in the world wide web. *Proceedings of the International World Wide Web Conference*, 1999.

[6] Jon M. Kleinberg. Authorative sources in a hyperlinked environment. In *Proceedings of the Ninetn Annual ACM-SIAM Symposium on Discrete Algorithms.*

[7] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The web as a graph: measurement and models and methods. *Proc. 5th Annual Int. Conf. Computing and Combinatorics*, 1999.

[8] Maxim Lifantsev. Open peer-review as web's self-organization force. *http://citeseer.nj.nec.com/353131.html*, no pub.year.

[9] Paul P. Maglio and Teenie Matlock. Metaphors we surf the web by. *http://citeseer.nj.nec.com/maglio98metaphors.html*, 1998.

[10] Alain Michard and Giang Pham-Dac. Description of collections and encyclopaedias on the web using xml. *Archives and Museum Informatics*, 1998.

[11] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to wordnet: An on-line lexical database. Technical report, http://www.cogsci.princeton.edu/ wn/, 1993.

[12] Masashi Toyoda and Masaru Kitsuregawa. A web community chart for navigating related communities. http://www10.org/cdrom/posters/p1083/index.htm.