Characterization Hierarchy containing Augmented Characterizations

F.C. Berger, P. van Bommel*, Th.P. van der Weide
Computing Science Institute
University of Nijmegen
P.O. Box 9010
6500 GL Nijmegen
The Netherlands

Abstract

The focus of this paper is information retrieval and filtering in traditional retrieval contexts as well as on the Internet. Information Modelling techniques (e.g. NIAM, ER, OO) are used for the characterization of documents to be retrieved. This brings together the worlds of Information Modelling (IM) and Information Retrieval (IR, or: document retrieval). Although IM is in most cases used for traditional (non-document) databases such as relational databases (e.g. SQL), these techniques can be applied to IR in order to obtain different characterization levels for information objects.

The level of index expressions is discussed in detail and extended, yielding augmented index expressions containing additional (semantic) information. This is done in the following context. A searcher in a list of phrases serving as an index to documents often has problems finding the right words when the information sought for has to be described. Offering alternative phrasings and pointing to related concepts in the index could be a great help in this difficult process of query formulation.

Usually the index is obtained by characterizing documents. This paper describes the addition of semantic relations to the index. Various strategies for relating nodes in an index are discussed, and criteria for adding new index entries are introduced. The effects of adding relations on the process of offering support during the formulation process are treated as well.

Keywords: information retrieval; user modelling; query formulation.

Classification: 68P20 (AMS-1991); H.3.3, H.5.1 (CR-1991)

Published as:

Encyclopedia of Microcomputers, to appear.

Edited by A. Kent and J.G. Williams.

*Corresponding author. E-mail: pvb@cs.kun.nl

Contents

1 Introduction

Whenever the need arises to formulate an idea in a concise way, one finds that this is a painstaking process. This is even more the case when the need exists to express this idea within the boundaries of a restricted vocabulary, e.g. a set of terms. This however is the actual situation in many of the present-day information retrieval systems. As the query construction process is the most important process in the interaction with a retrieval system, it is necessary to support the searcher in any way we can. An on-line help system offering alternative phrasings and related terms can be quite useful during query construction. For instance, knowledge concerning the existence of more specific or less specific phrasings for the original query can be a reason for changing this original query.

As was mentioned in [1], the results of a search can only be as good as the description of the items being sought. Therefore we need adequate support during the query construction phase. In terms of Meron & Kuhns[2], we need to minimize the semantic noise in the information need representation. As a starting point for such query construction support, we take a look at formalisms for extending queries *after* they have been constructed. Extending queries with additional terms based on thesaurus information has been shown to increase the recall of an information retrieval system (see e.g. [3], [4]). For instance if we have a query on documents concerning 'Holland', the effect of extending the query to 'Holland' OR 'The Netherlands' will be that additional documents dealing with the subject 'The Netherlands' are retrieved. The problem with this approach is that the expanded query contains elements being irrelevant to the user's information need. The problem then is to select that part of the extended query covering the user's information need. The extension of queries after construction is also applied in [5]. This approach is based on semantic links, where semantic links are used to propagate Retrieval Status Values.

The problem with expanding a query *after* submission is that the user could develop a sense of having lost control. The retrieved documents only show a partial relevance to the submitted (i.e. unexpanded) query. We therefore propose an interactive process wherein the user constructs a query, in cooperation with the system. The system provides knowledge concerning semantic relations.

The state-of-the-art in information retrieval is based on hyper-media. There are a number of ways hyper-media based information retrieval can be modeled. We adopt a layered architecture (see e.g. [6]), and specifically a Two-level Hyper-media Architecture (see e.g. [7], [8]). There are many ways a representation of the information need can be constructed. In this paper we adopt a well-known approach called *Query by Navigation* (see e.g. [9], [10], [11]). In Query by Navigation a user is allowed to travel through a hyper-text presentation of an index. The aim of this search process is to find a set of document descriptions being the best description of the information need. The index is constructed through a characterization process where each document is represented by a hierarchy of broader and narrower phrases in the index.

This paper is partially based on our previous paper [12]. We will discuss the effects of adding new links to the characterization network. Adding a link is based on the relations between nodes in the network. However we do not restrict ourselves to nodes derived from characterization. We will show that adding new nodes representing for instance broader terms is also an attractive possibility. In Section 2 we will introduce the concept of document characterization. The process of Query by Navigation is explained in section 3. We show the interaction language the user may employ during the search. Also the aid the user can call on while searching is discussed. Relations between nouns the subject of Section 4. We explain our strategy for adding links in the Hyper-index based on the aforementioned relations between nouns. Augmenting a characterization network has a number of consequences, and these are discussed in Section 5.

2 Document characterization

A well-known approach to reduce the information retrieval problem stated earlier is to replace a document with its characterization (see e.g. [13]). This characterization is a terse summary of a document's contents. As a drawback we can see a loss of information because it is impossible

to contain all the information present in a document in the characterization. Also there is the problem that a document's characterization is less unique than that document. So if we have a characterization χ it is quite possible that a number of documents share this characterization.

2.1 Characterization Hierarchy

In Information Retrieval (IR), information objects are weakly characterized, i.e. different information objects can have the same characterization. Information Modelling (IM) on the other hand deals with strong characterization (or: unique identification). As an example, relational databases usually contain data about objects, s.t. all objects can be uniquely identified. Identification is a special topic of interest in conceptual data modelling and database design.

A basic technique for characterizing objects in IR uses keywords. We may for instance search for documents about *meditation* and *course*. For reasons of retrieval performance (e.g. recall, precision, search length), characterizations in IR become stronger. Examples of stronger characterizations such as index expressions and augmented index expressions will be considered in the next sections. This results in a Characterization Hierarchy (see also [14]).

In figure 1 we see an example Characterization Hierarchy. In the right part, six different representations to be used for characterization are given. In the left part the associated characterization *level* is given.

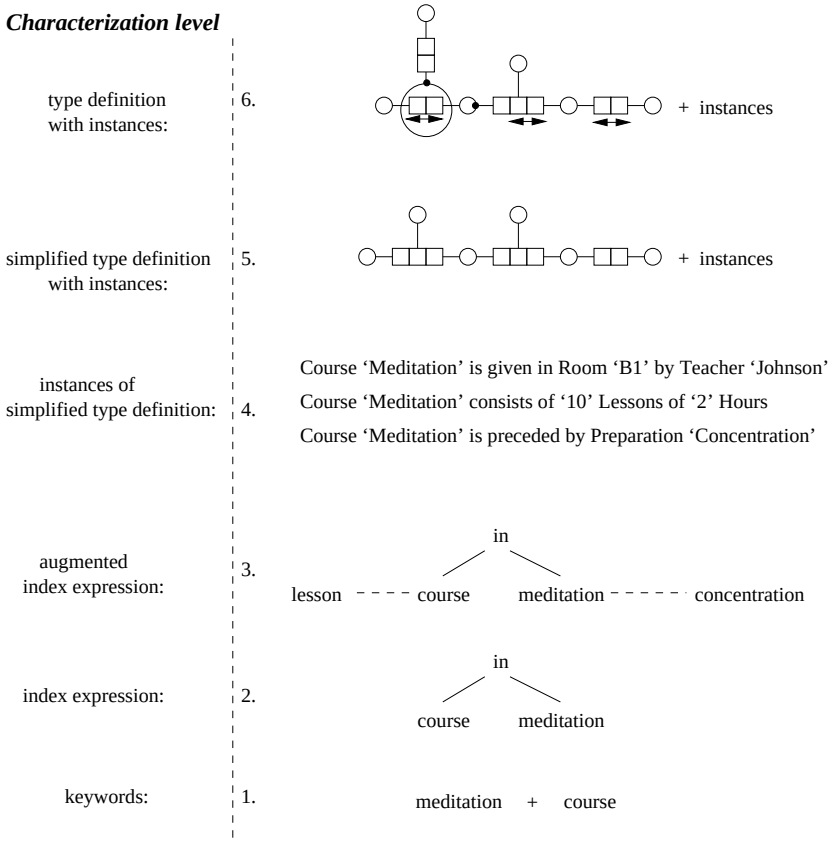

Figure 1: Example of Characterization Hierarchy

The hierarchy in figure 1 should be interpreted as follows. At the bottom level, a simple characterization is given based on keywords. The next level gives an index expression, also called characterization *network* (not to be confused with Characterization Hierarchy!). Index expressions

are augmented with additional information on the third level. Augmented index expressions are an excerpt of complete sentences (fourth level). These sentences are instances of type definitions known from conceptual data modelling (cf. [15], [16]). On the fifth level a simplified type definition is given (also called information structure), whereas the sixth level presents a full type definition (information structure with constraints).

2.2 Index expressions

In this section we consider the level of index expressions. Index expressions (see e.g. [17]) are pruned sentences taken from a document in order to index this document. We prune sentences because stop words are to be removed. A stop word is a word occurring so frequently throughout the collection of documents that it has little or no discriminative power. Therefore we remove all stop words from the title. The general syntax for index expressions is based on the sets of connectors $\mathcal{C}$, terms $\mathcal{T}$ and adjectives $\mathcal{A}$. Basically, connectors correspond to prepositions, whereas terms correspond to nouns. The syntax is:

$$\begin{aligned}
\mathsf{Expr} &\rightarrow \epsilon \mid \mathsf{Nexpr} \\
\mathsf{Nexpr} &\rightarrow \mathsf{Unit}\ \{\ \mathsf{Connector}\ \mathsf{Unit}\ \}^* \\
\mathsf{Unit} &\rightarrow \mathsf{Term} \mid \mathsf{Adjective}\ \mathsf{Unit} \\
\mathsf{Connector} &\rightarrow c \in \mathcal{C} \\
\mathsf{Term} &\rightarrow t \in \mathcal{T} \\
\mathsf{Adjective} &\rightarrow a \in \mathcal{A}
\end{aligned}$$

The general form of an index expression I is $I = u_0 \bigotimes_{i=1}^{k} c_i I_i$. In this expression u_0 is called the leading unit, whereas c_i, $1 \leq i \leq k$ are connectors joining subexpressions I_i to t_0. This denotation also shows the relative position of a subexpression I_i with respect to a subexpression I_j. If $i < j$ then I_i appears earlier in sentence I than I_j.

Given two index expressions I and J we can say that J is a subexpression of I if J is contained in I. We then write $J \circledS I$. A subexpression J of I is called a direct subexpression if it can be obtained from I by removing one leaf from I. The set of terms $\mathcal{T}$ is defined as the set of index expressions having only the empty index expression ϵ as a subexpression. We can create a document characterization by taking the power index expression set $\mathcal{P}(I)$, being the set of all subexpressions that can be formed from the index expression I. This power index expression set is then turned into a partial order by creating a graph $L = \langle \mathcal{P}, \mathcal{E} \rangle$ where $\mathcal{E}$ is the set of arcs defined by: $\alpha \rightarrow \beta \in \mathcal{E}$ if α is a direct subexpression of β.

2.3 Characterizations in general

In this section we abstract from any particular characterization level, At the basis of a characterization are so-called descriptors. A descriptor is an element of some descriptor language which describes the contents of a document. The characterization of a document d is expressed as $\chi(d)$. It consists of a set of descriptors $\mathcal{D}$ and a set of edges $\mathcal{E}$ between these descriptors. As a result of the set of edges we have a partial ordering of the set of descriptors. The graph which has been constructed in this way is a Directed Acyclic Graph. Some descriptors have the special property that they are the most narrow characterization of some document. Such a descriptor is called a root descriptor.

Definition 2.1

For any document D, $\mathsf{Root}(D)$ is defined as the minimal element of the descriptors which describe the content of D. $\mathsf{Root}(D) = d \in \chi(D) : \forall_{e \in \chi(D), e \neq d}[d \rightarrow e \in \mathcal{E}^*]$ *where $\mathcal{E}^*$ is the transitive closure of $\mathcal{E}$.* □

The characterization relation is total, i.e. each document has a characterization. For each document the set of descriptors is non-empty. The set of edges may be empty. Demanding that a

characterization is non-empty is crucial, since a query is constructed from such characterizations. Should a document have an empty characterization it can never be retrieved. In order to guarantee a non-empty characterization the universal descriptor ε is introduced. This is a descriptor which will always describe a document. The universal descriptor is dependent on the set of documents. For instance if $\mathcal{O}$ is a set of slides, then the descriptor slide could be the universal descriptor.

[C1] $\forall_{D \in \mathcal{O}} [\varepsilon \in \chi(D)]$

The set of documents and the characterization network are strongly linked in the sense that relations between documents are preserved on the characterization level. So the fact that a document is for instance a chapter of another document is also expressed in the characterization network. The question remains precisely which elements of the characterization network are involved in this inherited relation. Clearly we have to limit ourselves to descriptors for d and e. If documents d and e are related, then not all descriptors of the characterization hierarchies for d and e are related. Only those descriptors of the hierarchies which most uniquely describe d and e are related.

[C2] For any two documents D and E we have that $D \ R \ E \Rightarrow \mathsf{Root}(D) \ R \ \mathsf{Root}(E)$

2.4 Support

Each descriptor indexes a certain subset of the set of documents. This set is called a descriptor's *support*, denoted as $\mathsf{Support}(d)$ if d is a descriptor.

Definition 2.2

$$\mathsf{Support}(d) = \{ D \in \mathcal{O} \mid d \in \chi(D) \}$$ □

As the support relation is the inverse of the characterization relation, it is clear that the edges are related to the support as follows:

Lemma 2.1 For any two descriptors d and e, if $d \rightarrow e \in \mathcal{E}$ then $\mathsf{Support}(d) \supset \mathsf{Support}(e)$

If we look at all the refinements of a descriptor d (i.e. all descriptors being more specific than d), then the following property holds:

Lemma 2.2 $\bigcup_{d \rightarrow e} \mathsf{Support}(e) \subseteq \mathsf{Support}(d)$

An important conclusion is that there can be no descriptors not referring to any document. This requirement is important, because query results are derived from the support of the descriptors in the query. If d is a descriptor with an empty support, and the query $q = \{d\}$ is submitted for evaluation, then *no* retrieval result can be determined.

Lemma 2.3 For any descriptor d we have that $\mathsf{Support}(d)$ is not empty.

The number of documents indexed by a descriptor can vary greatly. Some descriptors will be used sparingly, while others have become a virtual stop word with regard to the particular document collection. In fact, there is at least one descriptor indexing *all* documents:

Lemma 2.4 The support of the universal descriptor is the set of documents $\mathcal{O}$.

This descriptor with maximal support therefore has the characteristic that if we are in the descriptor ε, recall will be maximal. Precision on the other hand will be very low, but not minimal.

Proof:

Since the universal descriptor is a sub-characterization of *all* descriptors we have that $\mathsf{Support}(d) \subset \mathsf{Support}(\varepsilon)$ for all descriptors d. As the characterization relation is total, the only set that fulfills this requirement is the set of documents $\mathcal{O}$. □

Next we consider *refinement* steps. A descriptor d is said to refine another descriptor e if d offers a more narrow description than e. Every document carrying d in its characterization also carries e in its characterization. The concept of refinement is defined as follows:

Definition 2.3
We say that e refines d if:

1. *an edge from d to e is present: $d \rightarrow e \in \mathcal{E}^*$*
2. *the support of d is a proper subset of the support of e:* $\mathsf{Support}(d) \supset \mathsf{Support}(e)$

□

Clearly, no descriptor may refine itself. This is a direct effect of the irreflexivity of the characterization edges.

2.5 Hyper-index construction

In this section we discuss how characterizations can be implemented as a hyper-text, thus yielding a so-called hyper-index. A main feature of hyper-text is the link connecting elements of the hyper-text on the basis of some relation between the connected elements. Following Frei & Stieger (see [5]), a link is a tuple

$$\langle \lambda_T, \lambda_s, \lambda_e \rangle$$

where

$$\begin{aligned} \lambda_T \in \mathcal{T} &= \text{the type of the link} \\ \lambda_s \in \mathcal{D} &= \text{the origin of the link} \\ \lambda_e \in \mathcal{D} &= \text{the destination of the link} \end{aligned}$$

After characterization there are only two types, based on a descriptor's support: refinement and enlargement. These are each other's inverses. Following a refinement link means that we arrive at a descriptor characterizing a subset of the support of the source of the link. When an enlargement link is followed the destination of that link will characterize a superset of the support of the source. A third type of link is the 'see-also' type. This type is not based on the support of source and destination. In stead this relation is solely based on the descriptors themselves. There could be other components of a link as well. As a possible component we could have the time of creation, or alternatively the number of times the link has been traveled, the so-called activation count. We have chosen not to include such information because we do not have a need for these components here.

With the previously defined characterization network we can now define a hyper-index consisting of a set of descriptors. Transitions between these descriptors are possible via a set of links.

Definition 2.4
Given a characterization network $\langle \mathcal{D}, \mathcal{E} \rangle$, a hyper-index is defined as the tuple $\langle \mathcal{D}, \mathcal{L} \rangle$, where the set of links $\mathcal{L}$ is defined by

$$\mathcal{L} = \left\{ \langle \text{has-refinement}, d, e \rangle \;\middle|\; e \text{ refines } d \right\} \cup \left\{ \langle \text{has-enlargement}, e, d \rangle \;\middle|\; e \text{ refines } d \right\}$$

where d and e are descriptors. □

Suppose we have the set of documents given in Figure 2. Using our characterization algorithm will yield the hyper-index graph of Figure 3. From this figure we see that the hyper-index consists of two separate hierarchies: the representation for document D is separate from the representations of E and F. These latter two hierarchies are linked because they share the descriptor theft. If we examine the hyper-index constructed in the previous example, we can see that

Number	Title
D	Car burglary in Holland
E	Theft of bicycles in The Netherlands
F	Theft of automobiles in The Netherlands

Figure 2: Example set of documents

theft of automobiles in the netherlands refines theft, because Support(theft of automobiles in the netherlands)=$\{F\}$ and Support(theft)=$\{E, F\}$. Note that theft in the netherlands does not refine theft because they both have the support $\{E, F\}$. This example shows that even though a direct link from for instance theft of automobiles exists, it is also possible to reach theft via another path. However this is not a transitive link, because the indirect path involves a refinement link.

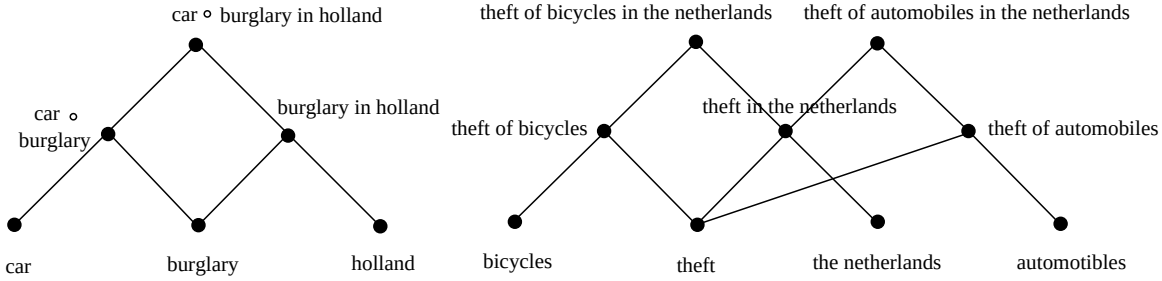

Figure 3: Example hyper-index

3 Query by Navigation

As a means of constructing a query for the retrieval system, Query by Navigation is used. With this formalism a user is allowed to meander through a hyper-text presentation of an index. Descriptors considered as representative of the Information Need can be marked as such. The sequence of search actions (called a search path) is analyzed and an assumption is made concerning the interests of the searcher. Based on this assumption the search process is supported by trying to guide the searcher to the index entry most likely to be the representation of the information need. This section builds on the formalism presented in [9].

The tool for disclosing a set of documents is a hyper-media-based information retrieval system. The frame of reference in this paper is a two-level hyper-media architecture ([18]). This describes how a hyper-media can be formed by creating two levels, the document level or hyper-base, and the index level or hyper-index.

3.1 Search Paths

In order to let the user construct a description of the information need we allow this user to navigate through the hyper-index. The sequence of search decisions taken during this navigation is called a search path. A search path p of k search actions can be written as $p : a_1; \ldots; a_k$, where the a_i's are search actions.

In every focus the user can execute a search action. The options are all the descriptors linked to the current focus φ. For the purposes of this paper we have chosen not to allow documents in the set of options. The set of options in a focus φ then is defined as:

Definition 3.1

$$\mathsf{Opt}(\varphi) = \left\{ d \in \mathcal{D} \;\middle|\; \langle \lambda_T, \varphi, d \rangle \in \mathcal{L} \right\} \qquad \square$$

When we take as an example the hyperindex of Figure 3, then the options in the focus car burglary is the set

$$\{\text{car}, \text{burglary}, \text{car burglary in holland}\}.$$

3.2 Search length

After a searcher has performed a number of search actions it is possible that a correct representation of the information need has been constructed. The number of actions needed to get this representation is called the search length. Novice searchers can be expected to have a relatively large search length. More experienced searchers most likely require far less actions to construct a representation of the information need. Note that experience comes in two varieties: experience with the process of navigation, and experience with the contents of the hyperindex in which the navigation is done (see e.g. [19]).

3.3 Search actions

In order to describe the behavior of the searcher in the hyperindex we consider four types of actions:

1. mark the focus or one of the options as being relevant to the information need (this style of searching is called *berry picking*; see e.g. [20]). This is denotated as $\star d$.
2. shift the focus to one of the options of the current focus. This is denotated as $\rightarrow d$.
3. discard an option as not worthy of further pursuit. The denotation for this action is $\neg d$.
4. affirming an option as highly interesting for further exploration. This is denotated as $+d$.

Before any search action has been taken (i.e. the search path consists of the empty descriptor ε) all descriptors are unmarked.

[M1] $p : \varepsilon \Rightarrow \forall_{d \in \mathcal{D}} [\neg \, \mathsf{Mark}(d)]$.

The only way in which a descriptor can become marked is by an explicit marking action occurring on search path p.

[M2] $\forall_{d \in \mathcal{D}} [\mathsf{Mark}(d) \Rightarrow \exists_{a \in p} [a = \star d]]$

Definition 3.2
Given a search path $p = a_1; \ldots; a_k$ we say that descriptor d has been visited when either one of the following events has occurred on p:

1. *there is a j such that $a_j = \rightarrow d$*
2. *there is a j such that $a_j = \star d$*
3. *there is a j such that $a_j = \neg d$*
4. *there is a j such that $a_j = +d$*

□

After the search process has been terminated the searcher could ask the retrieval system which set of documents satisfies the constructed query. The query to be submitted for evaluation consists of the marked descriptors:

$$q = \{ d \in \mathcal{D} \mid \mathsf{Mark}(d) \}$$

The most important characteristics of a search path are its origin, i.e. the descriptor where the search is started, and its destination, i.e. the descriptor where the decision was taken to stop searching because that descriptor offers the best description of the information need. The second most important feature of a search path is its length, i.e. the number of actions taken

along the path. When we examine these actions, a statement can be made concerning the level of contradiction. This will be high when many inconsistent actions have been performed. The level of contradiction will be low when no or only a few of such conflicting actions have been performed.

Retrieval of documents could be done by for instance retrieving the support of the final focus of the search path. This would discard any action taken on the search path. A more elaborate scheme would analyze these previous decisions on the search path. Ultimately, this leads to a ranked set of documents. Independent of the way in which we translate a search path into a set of relevant documents, having access to a descriptor's support is always needed.

3.4 Spreading marked descriptors

Although the set of marked descriptors is the most important reflection of a searcher's preference, the descriptors directly linked to these marked descriptors also are interesting. However these non-marked nodes are less important than the marked ones. In order to indicate that a node has been indirectly marked we introduce the set Mark^n where superscript n denotes the distance from the set of marked descriptors. This process of propagating a search path through the set of descriptors is called *spreading*. The set Mark is equal to the set Mark^0.

Definition 3.3

Given a set of marked descriptors Mark*, the propagation of the marked descriptors over distance* n *is defined as:*

$$\mathsf{Mark}^n = \left\{ d \in \mathcal{D} \;\middle|\; \neg \exists_i \left[d \in \mathsf{Mark}^i \right] \wedge \exists_{e \in \mathcal{D}} \left[\mathsf{Mark}^{n-1}(e) \wedge \langle \lambda_T, e, d \rangle \in \mathcal{L} \right] \right\}$$

□

Example 3.1

Consider the example hyper-index of Figure 4. If a search path is constructed where node a

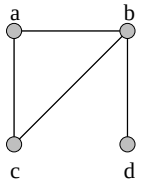

Figure 4: Example hyper-index

is marked, then:

$$\begin{aligned} \mathsf{Mark} &= \{a\} \\ \mathsf{Mark}^1 &= \{b, c\} \\ \mathsf{Mark}^2 &= \{d\} \end{aligned}$$

If on the other hand a search path has been constructed marking node b*, then:*

$$\begin{aligned} \mathsf{Mark} &= \{b\} \\ \mathsf{Mark}^1 &= \{a, c, d\} \end{aligned}$$

□

As any descriptor can be reached from any other descriptor, there is a k such that $\bigcup_{i=0}^{k} \mathsf{Mark}^i = \mathcal{D}$. This reflects the fact that all descriptors are related to each other. Some are related strongly,

others are related very distantly. If our wish is to reach all descriptors from a certain set of marked descriptors, it is important to know the maximum number of iterations to be performed. The following lemma gives the upper bound of this number.

Lemma 3.1 $k < |\mathcal{D}|$

Proof:

With a graph with n nodes ($n = |\mathcal{D}|$), it takes at most $n-1$ steps to reach any descriptor e from a given descriptor d. This worst case occurs for instance when $\mathcal{D}$ is a chain, i.e. when each d_i has as only two neighbors, viz. d_{i-1} and d_{i+1}, and $d = d_1$ and $e = d_n$.
Therefore $n-1$ is the highest value k has to assume such that all descriptors are element of some Mark^k. □

So it is possible to reach each descriptor from a set of marked descriptors. In practice however the retrieval system looks only a few links away from the search path. The motivation for this is that, given the evidence of the set Mark, the belief in the validity of Mark^i decreases as i increases.

Example 3.2

Consider again the example hyper-index of Figure 4. With the first search path of Example 4.1, $k = 2$ would suffice, while for the second search path $k = 1$ is enough to mark all descriptors. □

This definition of spreading disregards the type of the links. A more accurate kind of spreading should take this into account. Suppose we have a marked descriptor d with the following links:

$$\langle \text{refines}, d, e \rangle \text{ and } \langle \text{refines}, f, d \rangle$$

If we have to propagate d's marking we have to choose between marking either e or f.

After the searcher has traveled a search path p we would like to make a statement concerning the relevance of documents. Obviously, if none of the descriptors describing a document D have been visited on the search path, then D's relevance will be very low. On the other hand, if all the descriptors describing D have been marked then D's relevance will be very high. It would seem that the relevance of a document is proportional to the number of marked descriptors. Given a search path p, the relevance of a document d can be derived with the following definition:

Definition 3.4

Given a document D and a search path p, the relevance (or: the Retrieval Status Value) of D with respect to p is defined as:

$$\mathsf{rel}(D|p) = \frac{|\chi(D) \cap \mathsf{Mark}|}{|\chi(D)|}$$

□

This definition favors documents whose descriptors have all been marked. However, suppose we have a document e, none of whose descriptors have been marked. Yet, all of e's descriptors are in the immediate vicinity of the search path. One possible reason for this proximity is that these descriptors are strongly related to descriptors on the search path. While according to definition 3.4 e'd relevance is zero, arguably e has at least some relevance.

Definition 3.5

Given a document D and a search path p, the relevance of D with respect to p is given by:

$$\mathsf{rel}(D|p) = \sum_{i=0}^{k} w_i \frac{\left|\chi(D) \cap \mathsf{Mark}^i\right|}{|\chi(D)|}$$

□

In order to give preference to descriptors closest to the search path, w_i is proportional to the distance i between a descriptor and the search path, i.e. $w_i \sim \frac{1}{i+1}$.

Based on the Retrieval Status Values of documents it is possible to create a walk. This walk starts at the document with the highest RSV, and is followed by the next relevant document. Clearly it is necessary to define some threshold for the RSV. When a document's RSV rises above this threshold it is important enough to be incorporated into this walk. The threshold can be determined by setting an upper bound (say, α) on the probability of labeling a non relevant document as relevant with the relevance measure defined in Definition 3.5. Since most likely the documents in a walk are not connected in the Hyperbase, we need to create temporary links between these documents. After the walk has been finished these links are removed. The searcher must still be allowed to follow the links from a document on the walk.

3.5 Effect on retrieval results

Each action has an effect on the results if documents had to be retrieved. Here we show how these results are in terms of the support of the descriptors involved in a search action. The only actions considered are the focus shift, the marking and the rejection. So suppose we have descriptors t and u with $\mathsf{Support}(u)$ a subset of $\mathsf{Support}(t)$. Then we have the retrieval results as shown in Figure 5.

Action	Result
$t; \star u$	$\mathsf{Support}(u)$
$\star t; \neg u$	$\mathsf{Support}(t) - \mathsf{Support}(u)$
$\neg u; \star t$	$\mathsf{Support}(t) - \mathsf{Support}(u)$
$\star u; \neg t$	$\varnothing$

Figure 5: Boolean retrieval result of search action

The first search action narrows the scope of interest to subject u. Therefore only documents dealing with this subject should be retrieved. With the second search action, the user specifies that the narrower subject is not interesting, and thus documents dealing with subject u should not be retrieved. If a user goes to a broader subject, as shown in the third search action, we have a situation similar to the second search action. Finally, the fourth search action shows a very peculiar occurrence of both marking a subject u relevant, whereas the broader subject is judged not relevant.In terms of boolean retrieval, the search result would yield *no* documents.

3.6 Search aid

In retrospect, the latter decisions on the search path could be consistent with decisions taken earlier on the search path. Apparently, as a result of traveling a search path certain areas of the characterization network become less likely to be marked as relevant to the information need.

In order to support the process of information need determination the search path is analyzed. This analysis yields a probability function over the set of descriptors.

Definition 3.6

The target probability is a probability function $P : \mathcal{D} \to [0, 1]$. It expresses the probability of a descriptor becoming a berry, viz. that the descriptor will become the subject of a marking operation. □

Example 3.3

Consider the hyper-index of Figure 3. If a searcher moves from theft *to* theft in the netherlands*, then the target probability for* theft *will decrease.* □

Some descriptors will have a markedly higher target probability than other descriptors as a result of a search path. Others will have a very low target probability, thus stating that most

likely the searcher will not be interested in that descriptor. A very special set of descriptors is the search target set.

Definition 3.7

The search target $\mathcal{S} \subseteq \mathcal{D}$ is the set such that

1. $\forall_{d,e \in \mathcal{S}} [P(d) = P(e) \Rightarrow d = e]$

2. $\forall_{d \in \mathcal{S}, e \in \mathcal{D}} [P(d) \geq P(e)]$

□

The search target is a set of descriptors most likely to become a berry. As a means to support a user during concretization of the information need, with this search target in mind the options in each focus are ranked according to the shortest path to the search target. For n options, the function $\rho : \mathsf{Opt}(\varphi) \rightarrow \{1, \ldots, n\}$ assigns a ranking number to an option. The ranking resulting after an action a is performed is written as ρ_a.

Stating a hypothesis about the search target becomes more difficult as the level of contradiction increases. The ultimate goal of supporting the search process is to decrease the average search path length by trying to avoid actions which lead away from the search target.

Of course the situation might arise where there are a number of descriptors in different regions of the hyper-index forming a representation of the information need. In that case, after the searcher has been guided to one of these, he or she is then guided to the next one, et cetera.

4 Extending index expressions

The result of the characterization process is a network of descriptors, connected via links which indicate a refinement or an enlargement. This network will be augmented with semantic links, based on semantic relations such as synonyms, antonyms, and holonyms. In this section we will describe how the characterization network can be extended.

First we describe the tools which are required to find out whether two nouns are semantically related. Secondly we describe how the characterization of a document needs to be changed in order to uphold the mechanism of query by navigation. There are two distinct cases to be recognized, the first of which occurs when we are able to trace a relationship between two existing descriptors. The second occurs when we introduce a new descriptor as a result of detected relations.

4.1 Terminology

In order to avoid confusion with respect to the terminology introduced so far, an overview is presented here:

word: one or more characters representing a spoken word.

stem: the part of a word which is the essence of that word.

noun: a word that is the name of a subject.

descriptor: an element of a descriptor language .

term: a descriptor which consists of a single word.

index expression: a particular class of descriptor.

4.2 Basic tools

In this section we are not concerned with the way in which a lexicon has been implemented. We view a lexicon as a black box which has an input and an output. Based on our needs, we distinguish two cases. First, we must be able to determine which relation holds between two nouns τ_1 and τ_2. Since we assumed that only one relation holds between τ_1 and τ_2 we may safely speak

of a one-dimensional output. If $\mathcal{R}$ is the set of relations introduced in Section 5 and $\mathcal{N}$ the set of nouns, then

Definition 4.1
?relation : $\mathcal{N} \times \mathcal{N} \rightarrow \mathcal{R} \cup \{\text{null}\}$ □

If two nouns τ_1 and τ_2 are related as $\tau_1\ R\ \tau_2$ ($R \in \mathcal{R}$) then ?relation(τ_1, τ_2) returns R, else null.

In order to express the existence of a relation between two nouns we use the following definition:

Definition 4.2
?related : $\mathcal{N} \times \mathcal{N} \rightarrow \{\text{true}, \text{false}\}$.
?related$(n_1, n_2) =$ true *if there exists a relation* R *such that* ?relation$(n_1, n_2) = R$. □

Initially for all combinations of two terms $\neg$?related holds.

In order to keep track of terms which have been tested for the existence of a relation between them, the following relation is introduced.

Definition 4.3
For any two terms τ_1 *and* τ_2 *and a relation* R *the relation* ?tested *is defined as:*

$$\text{?tested} : \mathcal{T} \times \mathcal{T} \times \mathcal{R} \rightarrow \{\text{true,false}\}$$

□

Initially for all combinations of two terms there is no relation R such that ?tested holds.

4.3 Processing semantic relations

This section is devoted to explaining the actions which have to be performed after a semantic relation between two words has been detected. First we discuss the case where a relation exists between two existing descriptors. Secondly we show what has to be done when a relation exists between an existing descriptor, i.e. one which has been derived from characterization, and a word which has not been used in characterization but could be used for that purpose.
These two processes are performed until no new relations can be detected (i.e. for any $\tau_1, \tau_2 \in \mathcal{T}$ there exists a $R \in \mathcal{R}$ such that ?relation(τ_1, τ_2, R) holds), and no new descriptors can be created.

4.3.1 Relations between descriptors

The actions which need to be performed after a relation has been detected can be divided into three classes:

extend extending the characterization network, i.e. adding descriptors to a document's characterization, and defining new edges.

link defining links, i.e. adding new tuples $\langle \lambda_T, a, b \rangle$ to $\mathcal{L}$.

administer administrative tasks, i.e. keeping track of the descriptors have been tested.

In this section we formulate the effects of detecting a relation between two nouns.

Synonym

First we treat the synonym relation. In this situation there are two terms τ_1 and τ_2 in $\mathcal{T}$ which are synonyms. E.g. we might have the terms 'Ares' and 'Mars' which are synonyms in the context of deities. In most cases the support of τ_1 and τ_2 will *not* be the same. Especially in the case of title-based characterization Support(τ_1) and Support(τ_2) will have nearly always an empty intersection. The reason for this is that it is very uncommon for two synonyms to appear in on document's title. As τ_1 and τ_2 can be interchanged, the following actions need to be performed for the documents which are characterized by τ_1 and τ_2 (where χ is used as a shorthand for the characterization $\chi(d)$ of a document d):

Definition 4.4

If $?\text{relation}(\tau_1, \tau_2) = \text{synonym}$, *then* $\forall_{d \in \text{Support}(\tau_1) \cup \text{Support}(\tau_2)}$

- $\mathcal{D}(\chi) := \mathcal{D}(\chi) \cup \{\tau_1, \tau_2\}$
- $\mathcal{E}(\chi) := \mathcal{E}(\chi) \cup \{\varepsilon \to \tau_1, \varepsilon \to \tau_2\}$
- $\mathcal{L} := \mathcal{L} \cup \{\langle \text{has-synonym}, \tau_1, \tau_2 \rangle, \langle \text{has-synonym}, \tau_2, \tau_1 \rangle\}$
- $?\text{tested} := ?\text{tested} \cup \{\langle \tau_1, \tau_2, \text{synonym} \rangle, \langle \tau_2, \tau_1, \text{synonym} \rangle\}$

□

Antonym

Secondly let us consider what happens if we have a term τ_1 which has an antonym τ_2 which is also a term. For instance, suppose we have the term 'monotheism' which has the antonym 'polytheism'. Note that it is impossible that both τ_1 and τ_2 to appear in the characterization of the same document. Therefore, the support of τ_2 and τ_1 need not necessarily have an empty intersection. Note that the existence of an antonym relation between τ_1 and τ_2 does not allow us to draw any conclusions for the characterization network. For instance if document D is characterized by 'polytheism' we could say that D is *not* characterized by 'monotheism'. Or, if we presume the existence of negative descriptors, we could have the situation where '¬ monotheism' is a descriptor of D. Since this would imply that all descriptors have a negative counterpart we refrain from using such negative descriptors.

Definition 4.5

If $?\text{relation}(\tau_1, \tau_2) = \text{antonym}$, *then*

- $\mathcal{L} := \mathcal{L} \cup \{\langle \text{has-antonym}, \tau_1, \tau_2 \rangle, \langle \text{has-antonym}, \tau_2, \tau_1 \rangle\}$
- $?\text{tested} := ?\text{tested} \cup \{\langle \tau_1, \tau_2, \text{antonym} \rangle, \langle \tau_2, \tau_1, \text{antonym} \rangle\}$

□

If the searcher follows such a link, or more strongly yet also marks it, clearly a very important event occurs, because the searcher exhibits a drastic reversal of interest. Such a deviation from the previously shown search behavior means that most likely the search target has to be restated. The new search target will most likely be situated in a totally different region of the hyper-index.

Hypernym and hyponym

As a third possibility we have the case where a term τ_1 plays a role in a hypernym/hyponym hierarchy with a broader term τ_2. E.g. we could have τ_1 ='Ares' and τ_2 ='Greek God'. On the level of document characterization, any document characterized by τ_1 can also be characterized by the broader term τ_2. Note that the reverse can not always be said: a document characterized by τ_2 need not necessarily also be characterized by τ_1. The document dealing with 'means of transportation' could for instance discuss 'bicycles' and 'trains'. Therefore if τ_2 is a broader term of τ_1 then the following actions need to be performed ($\tau \to \tau'$ means that τ describes more documents than τ'):

Definition 4.6

If $?\text{relation}(\tau_1, \tau_2) = \text{hypernym}$, *then* $\forall_{d \in \text{Support}(\tau_1)}$

- $\mathcal{D}(\chi) := \mathcal{D}(\chi) \cup \{\tau_2\}$
- $\mathcal{E}(\chi) := \mathcal{E}(\chi) \cup \{\tau_2 \to \tau_1, \varepsilon \to \tau_2\}$
- $\mathcal{L} := \mathcal{L} \cup \{\langle \text{has-broader-term}, \tau_1, \tau_2 \rangle, \langle \text{has-narrower-term}, \tau_2, \tau_1 \rangle\}$
- $?\text{tested} := ?\text{tested} \cup \{\langle \tau_1, \tau_2, \text{hypernym} \rangle, \langle \tau_2, \tau_1, \text{hyponym} \rangle\}$

□

Meronym and holonym

Next we have the case where terms τ_1 and τ_2 are involved in a meronym/holonym hierarchy. An example for this case is 'mythology' (which is the holonym) and 'myth' (which is the meronym). This situation compares to the previous case. Again we have τ_2 which represents a broader subject than τ_1. A document characterized by τ_1 could also be characterized by τ_2. The reverse statement is again not necessarily always true.

Definition 4.7

If $?\mathsf{relation}(\tau_1, \tau_2) = \text{meronym}$, *then* $\forall_{d \in \mathsf{Support}(\tau_1)}$

- $\mathcal{D}(\chi) := \mathcal{D}(\chi) \cup \{\tau_2\}$
- $\mathcal{E}(\chi) := \mathcal{E}(\chi) \cup \{\tau_2 \to \tau_1, \varepsilon \to \tau_2\}$
- $\mathcal{L} := \mathcal{L} \cup \{\langle \text{has-part}, \tau_1, \tau_2 \rangle, \langle \text{is-part-of}, \tau_2, \tau_1 \rangle\}$
- $?\mathsf{tested} := ?\mathsf{tested} \cup \{\langle \tau_1, \tau_2, \text{meronym} \rangle, \langle \tau_2, \tau_1, \text{holonym} \rangle\}$

□

Stem

Finally, suppose we have a term τ which has a stem σ. An example could be 'periodical' which has the stem 'period'. If the stem of a term is not a meaningful word it can be expected that there is not yet a node for this stem. If this is the case a node has to be created. Creation of a node is the subject of the next section. If the stem is a word with a meaning, the following actions need to be performed:

Definition 4.8

If $?\mathsf{relation}(\tau, \sigma) = \text{stem}$, *then* $\forall_{d \in \mathsf{Support}(\tau)}$

- $\mathcal{D}(\chi) := \mathcal{D}(\chi) \cup \{\sigma\}$
- $\mathcal{E}(\chi) := \mathcal{E}(\chi) \cup \{\sigma \to \tau, \varepsilon \to \sigma\}$
- $\mathcal{L} := \mathcal{L} \cup \{\langle \text{has-stem}, \tau, \sigma \rangle, \langle \text{has-extension}, \sigma, \tau \rangle\}$
- $?\mathsf{tested} := ?\mathsf{tested} \cup \{\langle \tau, \sigma, \text{has-stem} \rangle, \langle \sigma, \tau, \text{has-extension} \rangle\}$

□

4.3.2 Introducing new descriptors

In this section we discuss what happens if semantic relations result in new descriptors. In this case we would have a term τ and a noun $n \in \mathcal{N} - \mathcal{T}$ such that $?\mathsf{relation}(\tau, n) \neq \text{null}$.

Synonym

Suppose that a searcher is interested in the subject 'Mars', and that although no node exists for this subject, there is a node for 'Ares'. If we assume that a searcher is unaware of the relationship between these subjects, then creation of a node for 'Mars' is definitely needed.
The only support which we can assign to n is the support of d. But then we would only create a copy of d, albeit under a different name. However we propose that adding a node for a synonym is justifiable if this new node can act as a 'bridge'. This means that there must also be another existing term which is related in some way to n.

Definition 4.9

if $?\mathsf{relation}(\tau, n) = \text{synonym}$ *and* $\exists_{\tau'} [?\mathsf{relation}(\tau', n) \neq \text{null}]$
then $\mathcal{T} := \mathcal{T} \cup \{n\}$; $\forall_{d \in \mathsf{Support}(\tau)} [\mathcal{D} := \mathcal{D} \cup \{n\}]$

□

Antonym

Suppose a searcher has indicated a lack of interest in the subject of 'monotheism'. A reasonable assumption would be that this searcher could be interested in the subject of 'polytheism'. Problems arise when this node is not present in the hyper-index. If we have a descriptor τ with an antonym n, then the question is which documents can be characterized by n. This is a very awkward situation, which can be resolved in the following way. We start by adding n to the characterization of the documents which are not described by τ. This is potentially a very large set. However this set can be further narrowed down if τ is a refinement of some other term τ'. This is therefore a suitable strategy for adding antonyms:

Definition 4.10

if $?\mathsf{relation}(\tau, n) = \text{antonym}$ *and* $\exists_{\tau'}\,[\tau \text{ refines } \tau']$
then $\mathcal{T} := \mathcal{T} \cup \{n\};\ \forall_{d \in \mathsf{Support}(\tau') - \mathsf{Support}(\tau)}\,[\mathcal{D} := \mathcal{D} \cup \{n\}]$

□

Hyponym and hypernym

When a searcher is constructing a query concerning the subject of *computer science*, he or she might also be interested in the narrower term *information retrieval*. However there might not actually be a document characterized as such. If we would like to add such a descriptor to the network, then we would be hard put to say which documents are characterized by it. The only statement that we can make is that such documents must be drawn from the set of documents characterized by a broader term. Without further information it is therefore difficult to add nodes for narrower terms. Broader terms, however look more promising.
Suppose we have a term τ_1 for which there is some noun n such that n **broader_than** t. A very interesting situation occurs when there are terms τ_2 for which also the relation n **broader_than** τ_2 holds. Imagine the situation where a searcher is in descriptor 'Ares'. In this case it would be very interesting to know that there are other descriptors which along with 'Ares' form a (partial) breakdown of the subject represented by 'Deity'. This could be a reason for the searcher to rephrase the query by substituting 'Ares' with 'Deity'.
A document which is characterized by τ can therefore also be characterized by n.

Definition 4.11

if $?\mathsf{relation}(\tau, n) = \text{hypernym}$ *and* $\exists_{\tau' \neq \tau}\,[?\mathsf{relation}(\tau', n) \neq \text{null}]$
then $\mathcal{T} := \mathcal{T} \cup \{n\};\ \forall_{\tau \in \mathcal{T}}\,[\forall_{R \in \mathcal{R}}\,[\neg\, ?\mathsf{relation}(\tau, n, R)]]$

□

Holonym and meronym

As the holonym and meronym have the same properties as the hyponym and hypernym relations, the same train of thought as used with these relations applies here. Again we add a holonym node to the set of descriptors if some of its meronyms already occur as descriptors.

Definition 4.12

if $?\mathsf{relation}(\tau, n) = \text{meronym}$ *and* $\exists_{\tau' \neq \tau}\,[?\mathsf{relation}(\tau', n) \neq \text{null}]$
then $\mathcal{T} := \mathcal{T} \cup \{n\};\ \forall_{\tau \in \mathcal{T}}\,[\forall_{R \in \mathcal{R}}\,[\neg\, ?\mathsf{relation}(\tau, n, R)]]$

□

Stem

Suppose we have a term τ with a stem σ. A document which is characterized by τ can thus also be characterized by σ. In a way, σ is a broader term than τ. If $\mathsf{Stem}(\tau) = \sigma$ and $\tau \neq \sigma$, then the following actions need to be performed:

Definition 4.13

$\forall_{d \in \mathsf{Support}(\tau)}$

- $\mathcal{D}(\chi) := \mathcal{D}(\chi) \cup \{\sigma\}$
- $\mathcal{E}(\chi) := \mathcal{E}(\chi) \cup \{\sigma \to \tau, \varepsilon \to \sigma\}$
- $\mathcal{L} := \mathcal{L} \cup \{\langle \text{has-stem}, \tau, \sigma \rangle, \langle \text{has-extension}, \sigma, \tau \rangle\}$
- ?tested := ?tested $\cup \{\langle \tau, \sigma, \text{has-stem} \rangle, \langle \sigma, \tau, \text{has-extension} \rangle\}$

□

5 Consequences

5.1 Going beyond terms

We have only discussed relations between terms. However, the terms are but a subset of the set of descriptors. We would also like to make statements concerning relations between descriptors. Specifically we would like to make statements concerning those descriptors which are maximal descriptors. In other words we would like to be able to translate relations between descriptors into relations between documents. Suppose we have a descriptor d which consists of a number of terms $t_1, \ldots, t_k$ and a descriptor e which consists of a number of terms $u_1, \ldots, u_l$. If none of the t_i's has a relation with any of the u_j's, then obviously no reason exists to derive a relation between d and e. On the other end of the scale, if all t_i's are related with one of the u_k's then there is definitely a reason to derive a relation between d and e. However, making a statement as to how much of the terms need to be related before we can say that d and e are related is beyond the scope of this paper. As an example of this case, suppose we have d which contains the term holland, while descriptor e contains the term the netherlands. Although d and e are likely to be relevant to each other, the exact value of this relevance depends on the other terms used by d and e. In order to indicate to the searcher that a semantic relation between two descriptors d and e has been derived, the type see-also is introduced.

Definition 5.1

If two descriptors d and e have been derived as semantically related, then the hyperindex links may be extended as follows:

$$\mathcal{L} := \mathcal{L} \cup \{\langle \text{see-also}, d, e \rangle, \langle \text{see-also}, e, d \rangle\}$$

□

If d and e are descriptors which can not be further extended (i.e. there is no descriptor f such that f refines d), and d and e are related based on their respective terms, then we introduce document links. These bi-directional links run from elements of $\mathsf{Support}(d)$ to elements of $\mathsf{Support}(e)$ in the following fashion:

Definition 5.2

If two maximal descriptors d and e have been found to be semantically related, then the document links Λ may be extended as follows:

$$\begin{aligned} \Lambda \quad := \quad & \Lambda \\ & \cup \{\langle \text{see-also}, D, E \rangle \mid D \in \mathsf{Support}(d) \wedge E \in \mathsf{Support}(e)\} \\ & \cup \{\langle \text{see-also}, E, D \rangle \mid D \in \mathsf{Support}(d) \wedge E \in \mathsf{Support}(e)\} \end{aligned}$$

□

5.2 Inherited relations

If we have added a descriptor n, and n is related to other nodes $\tau_1, \ldots, \tau_k$ then what happens to the inherited relations in which τ_i plays a role? For instance, we might have two descriptors low counties and western europe. Given these two descriptors we might decide to add a descriptor the netherlands. Now, if western europe is linked to descriptor european union, then we have to remove this link, and in stead add a new link from the netherlands to european union. If a descriptor d participates in an inherited relation, then there is a document D such that d is the maximal element of $\chi(D)$. If we add a descriptor to the set of descriptors which describe document D, then the situation might arise where the maximal element of $\chi(D)$ is no longer d. So in that case we have to remove the inherited link between d and say e and create an inherited link between the new maximal element d' and e.
Note that adding a broader term, or a holonym, does not change the maximal element of a hierarchy. Only in the case of a narrower term or a synonym does a possibility of a new maximal element occur.

5.3 Support

In the characterization network an edge from descriptor d to descriptor e meant that the documents indexed by e are also indexed by d. In the previous section the characterization results were augmented with semantic relations. From definitions 6.4 through 6.12 it is clear that the following lemma holds:

Lemma 5.1 If a link $\langle R, d, e \rangle$ exists in the hyperindex and $R \neq$ antonym, then either $\mathsf{Support}(d) \subseteq \mathsf{Support}(e)$ or $\mathsf{Support}(e) \subseteq \mathsf{Support}(d)$ holds.

5.4 Retrieval results

Although the characterization network has been augmented with semantic relations the same paths which could be constructed in the non-augmented network can still be constructed. Therefore the augmented network is at least as expressive as the non-augmented network.

Suppose descriptor d lies on search path S. After semantic links have been introduced, descriptors related to d lie closer to S than would be the case if no semantic links are present. As a consequence, they will contribute more to the relevance of the documents which they characterize. Thus it is to be expected that more documents will be considered relevant. The crux of the matter is of course whether the benefits (i.e. more documents which are truly relevant) outweigh the drawbacks (i.e. documents labeled relevant are not relevant at all).

One advantage of adding semantic relations is that most likely the spreading of the navigation results as described in Section 4.3 needs less iterations to achieve the same effect as when no semantic relations would be present. In fact, looking less far away from the set of marked descriptors is necessary in order to avoid a decrease in precision.

Precision and recall are an important way in which to measure the performance of a retrieval system. However, they do not take into account the effort needed by the searcher. In Section 3.2 we explained how the effort of finding a correct representation of the information need is reflected in the search length. Adding semantic links to the hyperindex should also result in a lower search length. In stead of painstakingly following the characterization network in search of related terms to the ones found thus far, related items are but a few clicks away due to the 'shortcuts' introduced by semantic links.

5.5 Cognitive load

Although the notion of adding links based on semantic properties is an interesting way to achieve greater insight in the structure of the hyperindex, there is certainly a price which the searcher has to pay. Because the average number of outgoing links has increased, the searcher most likely has to put more effort into making a selection. The relevance to the information need of the destination

of each outgoing link has to be determined, and simply because there are more outgoing links means that more descriptors have to be judged. For instance, we might have a descriptor d with say 3 links to synonyms. The cognitive burden can be reduced by creating an intermediate node called 'synonyms of d'. The links to the synonyms remain hidden from the searcher until the intermediate node is accessed.

It is to be expected that most searchers are interested in variations on descriptors which have been visited thus far. For instance, we might have a searcher whose train of thought goes like this:

I've just clicked this particular item. Wonder if there's anything else in here which looks a bit like it.

In this light the relations which are most likely to be of use to the searcher are the synonym relation and the stemming relation.

6 Conclusions and further research

In this paper we have shown how a characterization network can be augmented with information about semantic relations. The increased complexity of the network is offset by the increased means with which the user can express the information need. This increased expressive power shows in the fact that every query which could be constructed in the non-augmented network can also be constructed in the augmented network. Regarding the semantic relations which we have discussed, we have seen that the concept of antonyms is the most difficult to incorporate. The apparent reason for this is that it is difficult to add a descriptor to a document's characterization just because this descriptor has an antonym which *does* describe the document.

In this paper we have proposed to extend the characterization network and offer this extended network to the searcher. Another approach would be to let the characterization network remain intact, but only look for relations in which the focus plays a role. The only adjustment which would have to be made is that instead of looking for relations between all elements of $\mathcal{T}$ we would have to look for relations between elements of $\mathsf{Opt}(\varphi)$ if φ is the focus.

Only statical information has been added to the characterization network. Therefore, each user is presented the same network. An obvious source for refinement is to add individual user's links based on paths which have been traveled through the hyper-index. It can be argued that a link between the first descriptor and the last descriptor of a search path is justified. Adding such a link is only justified however if the search path satisfies certain conditions.

References

[1] R.H. Thompson and W.B. Croft. Support for browsing in an intelligent text retrieval system. *International Journal of Man-Machine Studies*, 30(2):639–668, 1989.

[2] M.E. Maron and J.L. Kuhns. On Relevance, Probabilistic Indexing and Information Retrieval. *Journal of the ACM*, 7:216–244, 1960.

[3] E.M. Voorhees. Query Expansion Using Lexical-Semantic Relations. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 1994.

[4] J. Kristensen. Expanding end-user's query statements for free text searching with a search-aid thesaurus. *Information Processing and Management*, 29(6):733–744, 1993.

[5] H.P. Frei and D. Stieger. The use of semantic links in hypertext information retrieval. *Information Processing and Management*, 31(1):1–13, 1995.

[6] D. Lucarella. A Model for Hypertext-Based Information Retrieval. In *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 81–94, Cambridge, United Kingdom, 1990. Cambridge University Press.

[7] M. Agosti, R. Colotti, and G. Gradenigo. A two-level hypertext retrieval model for legal data. In A. Bookstein, Y. Chiaramella, G. Salton, and V.V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 316–325, Chicago, Illinois, October 1991. ACM Press.

[8] P.D. Bruza and Th.P. van der Weide. Two Level Hypermedia - An Improved Architecture for Hypertext. In A.M. Tjoa and R. Wagner, editors, *Proceedings of the Data Base and Expert System Applications Conference (DEXA 90)*, pages 76–83, Vienna, Austria, 1990. Springer-Verlag.

[9] F.C. Berger and T.W.C. Huibers. A framework based on situation theory for searching in a thesaurus. *The New Review of Document and Text Management*, 1:253—276, 1995.

[10] F.C. Berger and P. van Bommel. Personalized Search Support for Networked Document Retrieval Using Link Inference. In A.M. Tjoa and R. Wagner, editors, *Proceedings of the Data Base and Expert System Applications Conference (DEXA 96)*, Vienna, Austria, 1996. Springer-Verlag.

[11] H.C. Arents and W.F.L. Bogaerts. Navigation without links and nodes without contents: intensional navigation in a third-order hypermedia system. *Hypermedia*, 5(3):187–204, June 1993.

[12] F.C. Berger and P. van Bommel. Augmenting a characterization network with semantical information. *Information Processing & Management*, 33(4):453–479, 1997. Reprinted with permission from Elsevier Science.

[13] G. Salton. *Automatic Text Processing–The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.

[14] P. van Bommel and Th.P. van der Weide. Multi Media Information Filtering on the WWW. In *Proceedings of the World Automation Congress*, Anchorage, Alaska, May 10-14 1998. TSI Press, NM, USA.

[15] P. van Bommel, A.H.M. ter Hofstede, and Th.P. van der Weide. Semantics and verification of object-role models. *Information Systems*, 16(5):471–495, October 1991.

[16] A.H.M. ter Hofstede and Th.P. van der Weide. Expressiveness in conceptual data modelling. *Data & Knowledge Engineering*, 10(1):65–100, February 1993.

[17] P.D. Bruza and Th.P. van der Weide. The Modelling and Retrieval of Documents using Index Expressions. *ACM SIGIR FORUM (Refereed Section)*, 25(2), 1991.

[18] P.D. Bruza. *Stratified Information Disclosure: A Synthesis between Information Retrieval and Hypermedia*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, 1993.

[19] K.L. Norman. Navigating the educational space with hypercourseware. *Hypermedia*, 6(1):35–60, January 1994.

[20] M.J. Bates. The design of browsing and berrypicking techniques for the on-line search interface. *Online Review*, 13(5):407–431, 1989.