Information Parallax

F.A. Grootjen, Th. P. van der Weide
Th.P.vanderWeide@cs.ru.nl
Radboud University Nijmegen,
Faculty of Science, Mathematics and Computing Science,
P.O. Box 9010,
6500 GL Nijmegen,
The Netherlands

To effectively use and exchange information among AI systems, a formal specification of the representation of their shared domain of discourse - called an ontology - is indispensable. In this paper we introduce a special kind of knowledge representation based on a dual view on the universe of discourse and show how it can be used in human activities such as searching, in-depth exploration and browsing.

After a formal definition of dualistic ontologies we exemplify this definition with three different (well known) kinds of ontologies, based on the vector model, on formal concept analysis and on fuzzy logic respectively. The vector model leads to concepts derived by latent semantic indexing using the singular value decomposition. Both the set model as the fuzzy set model lead to Formal Concept Analysis, in which the fuzzy set model is equipped with a parameter that controls the fine-graining of the resulting concepts. We discuss the relation between the resulting systems of concepts.

Finally, we demonstrate the use of this theory by introducing the dual search engine. We show how this search engine can be employed to support the human activities addressed above.

Keywords: formal concepts, latent semantics, dual search engine, ontology, knowledge discovery

INTRODUCTION

Sharing information is a real challenge in situations where we can not rely on some common underlying body of conceptualization and representation. Ontologies are crucial for enabling knowledge-level interoperation between agents, since meaningful interaction among them can only occur when they share some common interpretation of the vocabulary used in their communication (Farquhar, 1996).

The often cited article of Berners-Lee (2001) on the Semantic Web and its semantic foundation called 'Ontology' inspired many researchers in different fields to contribute to this topic, almost making it a `revamped cross-disciplinary buzzword' (Spynes, 2004). In this paper we will limit ourselves to Sowa's view on ontologies (Sowa, 2004; Sowa, 1984):

> The subject of ontology is the study of the categories of things that exist or may exist in some domain. The product of such a study, called an ontology, is a catalog of the types of things that are assumed to exist in a domain of interest *D* from the perspective of a person who uses a language *L* for the purpose of talking about *D*.

The construction of ontologies can be done either manually, automatically or in some hybrid supervised way. Manual construction is difficult and time consuming, but yields verified and rich ontologies. The effort constructing such an

ontology may be reduced by using a tool (Noy, 2001; Farquhar, 1996}, or by reusing other ontologies (Gruber, 1992).

Automatically (or fully unsupervised) constructed ontologies will be less detailed and may contain errors but are labor free and may even be evolutionary (adapting to the changing situation). Furthermore, in some application areas small errors and mismatches in the ontology do not lead to dramatic effects. An example of such an area is Information Retrieval.

This paper will present a formal, constructive and usable approach to unsupervised generation of concepts (or categories). Just like parallax is used to measure distance between celestial bodies, the dual perspective view on the universe of discourse creates an extra dimension which is visualized by concepts. This is referred to as *Information Parallax*.

In this paper we will study ontologies from the Information Retrieval point of view, but the theory can be applied to other fields as well. In the classic retrieval situation there is a searcher with an information need, and a system that is (hopefully) able to supply the information the searcher is looking for.

In order to solve the retrieval problem the system somehow has to determine the relevance of each information item. Of course, without feedback from the searcher the system is only able to do an 'educated guess' about the relevancy. Judging the relevancy of each information item can be simplified by using an ontology, especially if this ontology corresponds with the searcher's view on the universe of discourse.

A searcher can express an information need in several ways: the searcher may select or present a relevant information object (also referred to as a document), or formulate the information need by a combination of search terms (a query). In practice, searchers find it difficult to provide a proper query formulation, but have no problems in recognizing a document as being relevant. According to Taylor (1968), the following levels of information need may be distinguished:

1. The visceral need: the searcher experiences unconsciously something is missing. We assume the searcher at this stage is capable to recognize (at least) some characteristics of what could satisfy this need.
2. The conscious need: the searcher is aware of this need, and can judge the relevance of documents. At this stage the searcher may start to actively search for ways to satisfy the need.
3. The formalized need: the searcher has some either implicit or explicit formulation of the need. In case of an implicit formulation, a searcher can judge the relevance of description of the need. This assumption is the motivation for the mechanism of Query by Navigation.
4. The compromised need: a compromise of the best product composition from the actual assortment.

It will become apparent that dualistic ontologies can support the searcher on all these levels of information need.

From an abstract point of view, the information retrieval problem may be seen as a semantics transformation problem. We assume a searcher to have some mental model of the world. It is within this model that a searcher is aware of a knowledge gap. The searcher will try to find information objects that help the searcher to fill this gap. In order to facilitate finding information objects, a typical solution is to construct a catalogue which offers the searcher the opportunity to have a more directed avenue for search.

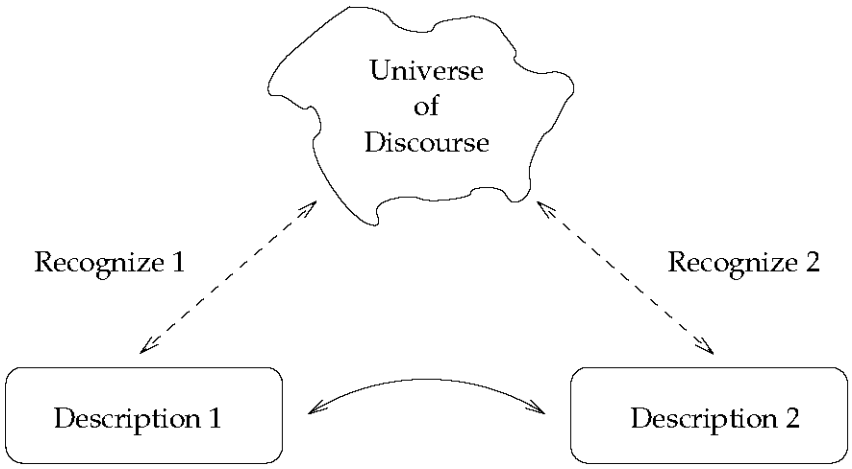

Figure 1. Different models of the real world

Traditionally (see Salton, 1983), Information Retrieval systems try to relate a set of *descriptors* (or terms) with a set of *information objects* (or documents). Since single terms have only limited descriptive power, IR systems allow individual terms to be combined into bigger semantic units. These units as referred to as *intensional (meaning) objects*. How terms are combined to form intensional objects depends on the actual IR system. Likewise, a combination of documents will be called an *extensional (meaning) object*. We will call an IR system *dual* if it can transform intensional objects into extensional objects, and vice versa. In general, we will use the term *dualistic system* for such a system. To demonstrate the look and feel of a realistic system, we present **DUALITY** (see figure 2 and section 4).

An example of an intensional object is a query while an example of an extensional object is the outcome of search. We will use the terms *query* and *search result* as alternative terms for intensional and extensional objects respectively. Being a combination of terms, intensional objects can be used to capture the meaning of a document, while extensional objects (being a combination of documents) can be used to capture the meaning of a term. As such, a dualistic system may be seen as a mutual semantics assigning system.

Figure 2. Initial query

The principle of alternating between the intensional and extensional view of an information need has been employed to some extent in previous work. In the probabilistic model of information retrieval (see Führ, 1989), a searcher is offered a sample set of documents. This extensional object is inspected by the searcher, and the relevant documents are marked. From this marking the retrieval system derives an intensional view on the information need. The system uses this intensional object to derive (using statistical techniques) a new extensional object, and offers this to the searcher for evaluation, etc. The probabilistic model may be seen as a one-sided dualistic system.
In Hearst (1996) a user interface for information retrieval based on the scatter-gather technique has been introduced. The system makes a limited classification of the collection, and presents this classification to the searcher. The searcher then selects the appropriate classes, after which the procedure is repeated on sub-collection consisting of the selected classes. In this procedure, the extensional objects subsets of the document collection, and the intensional objects consist of the summarizations of those classes.

In this paper, we focus on dualistic systems. In section 2 we show how different views on the real world can be combined to recognize concepts as semantic fixed points. We provide a formal definition and discuss some properties. In section 3.1 we focus on the interpretation of concepts in the context of the vector model, and find a relation with the latent semantic indexing approach (see Deerwester, 1990). This approach is based on the singular value decomposition, usually applied when noise removal is an issue. In section 3.2 we study concepts in the set model, and find the relation with formal concept analysis (see Ganter, 1996). This approach is very fine-grained, and can be used to find a needle in the haystack. In section 3.3 the fuzzy set model and fuzzy logic are the basis the formal concept approach is generalized, to cover some degree of uncertainty. This degree is a parameter steering the trade-off between granularity and (computational) complexity. In section 4 we apply this general theory by introducing the dual search engine **DUALITY**, and show the validity of the approach taken in this paper. Finally, in section 5 we present some conclusions.

DUALISTIC SYSTEMS

A system that can transform different views on some area of interest is called a *dualistic system*. For convenience, we will refer to these two views as the intensional and the extensional view. In this section we show that under weak assumptions this connection can be used to introduce a formal notion of concepts. These concepts are the base for an ontology.

Consider a dualistic system as described in the previous section. Let **I** be its set of intensional objects and **E** its set of extensional objects. We assume an equivalence relation $\equiv_i$ for comparing intensional objects expressing their similarity, and its counterpart $\equiv_e$ on extensional objects (we will leave out the indices when no confusion is likely to occur). The motivation to introduce similarity relations is to be able to handle for example equivalences that originate from syntactic variety in queries.

The Model

As intensional and extensional objects provide a different perspective on some area of interest, they will be semantically related. This is modeled by assuming that intensional and extensional objects have assigned a meaning in terms of each other. The function *match*: $\mathbf{I} \rightarrow \mathbf{E}$ interprets intensional objects in terms of extensional objects, the function *index*: $\mathbf{E} \rightarrow \mathbf{I}$ does it the opposite way (see figure 3). The assignment of meaning should be closed under similarity:

DS 1. *Similar queries yield a similar query result:*
$q_1 \equiv_i q_2 \Rightarrow match(q_1) \equiv_e match(q_2)$

DS 2. *Similar collections have a similar description:*
$d_1 \equiv_e d_2 \Rightarrow index(d_1) \equiv_i index(d_2)$

These rules express the intuition that (1) the matching of internal meaning is not dependent on its surface structure representation and (2) indexing of external meaning handles representation variety consistently. These requirements are referred to as the *similarity closure assumptions*. The resulting dualistic system is denoted as

$$< < \mathbf{I}, \equiv_i >, < \mathbf{E}, \equiv_e >, match, index >$$

We do not make any special assumptions on the relation between the functions *match* and *index* governing their interaction.

Remark: The basis <***O***, ***A***, ~> for Formal Concept Analysis is formal context, consisting of a set ***O*** of objects, a set ***A*** of attributes, and a relation ~ over ***A*** × ***O***. If for example, we interpret extensonal objects as subsets of a document collection **D**, and the functions *match* and *index* are antigonous in the sense that for all documents *d* and intensional objects *q* we have: $d \in match\ (q) \Leftrightarrow \exists_e [d \in e \wedge index\ (e) = q]$. The context relation then is defined as d ~ q ≡ $d \in match\ (q)$. Then <**D**, **I**, ~> is a formal context. Note that our approach does not require structure of the intensional and extensional objects, and also not a connection between the functions *match* and *index*.

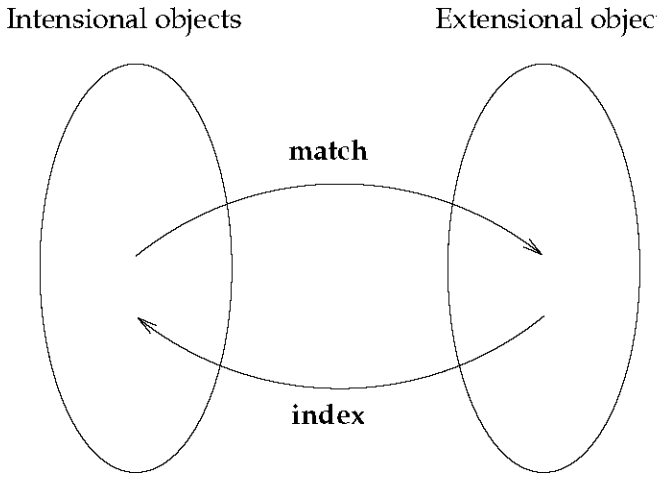

Figure 3. A dualistic system

Proto-concepts

In general, the meaning assigning functions *match* and *index* are not presumed to be inverse to each other. As a consequence, mutual sharing of meaning is a special property. As a first step, we wonder what objects are invariant under *mirroring*, i.e. the subsequent application of *index* and *match* in either order. We introduce proto-concepts as objects that have a similar mirror (reflection):

Qpc = {q| $index(match(q)) \equiv_i q$} (*query proto-concepts*)
Dpc = {d| $match(index(d)) \equiv_e d$} (*search result proto-concepts*)

If intensional object q has a similar mirror, then this meaning *match (q)*, assigned by the dualistic system, will also be a proto-concept. For example, if an intensional object q has a similar reflection from the dualistic system, then the meaning represented by q is also available in the extensional view. In other words, the functions *match* and *index* can be restricted and seen as mappings between these sets **Qpc** and **Dpc** of proto-concepts:

Lemma 1

1. $q \in$ **Qpc** $\Rightarrow$ $match(q) \in$ **Dpc**
2. $d \in$ **Dpc** $\Rightarrow$ $index(d) \in$ **Qpc**

So, if a query is similar to its reflection, then the (extensional) meaning is also similar to its reflections. For search results this is formulated analogously.

Proof:

We will only show the first case. Let $q \in$ **Qpc**, then $index(match(q)) \equiv_i q$, and according to the similarity closure assumption DS1, we conclude: match($index(match(q))) \equiv_e match(q)$, and thus match(q) $\in$ **Dpc**.

Furthermore, as a direct consequence of the similarity closure assumptions, these restricted functions respect similarity:

Lemma 2

1. $q \in$ **Qpc** $\wedge$ $q \equiv_i q'$ $\Rightarrow$ $q' \in$ **Qpc**
2. $d \in$ **Dpc** $\wedge$ $d \equiv_e d'$ $\Rightarrow$ $d' \in$ **Dpc**

Proof:

We show the first case. Let q be similar to its reflection, and q' be an alternative for q, then from the similarity closure assumption DS2 we conclude $index(match(q)) \equiv_i index(match(q'))$. Using the transitivity of similarity we conclude $index(match(q')) \equiv_i index(match(q)) \equiv_i q \equiv_i q'$, and thus $q' \in \mathbf{Qpc}$.

Proto-concepts thus are stable under the variation covered by similarity relations.

Abstracting from Variation

The similarity relations on intensional and extensional objects may be seen as a mechanism dealing with the variation that is offered by the underlying description mechanism. In this subsection we abstract from these variations.

It is easily verified that the restriction of the relation $\equiv_i$ to **Qpc** still is an equivalence relation. Let $\mathbf{Qc} = \mathbf{Qpc}\backslash\equiv_i$ be the corresponding set of equivalence classes. The equivalence class containing q is denoted as $[q]_i$. The class $[q]_i$ of may be seen as the deep structure of q. The same holds for the restriction of $\equiv_e$ to **Dpc**. The set **Dc** is introduced analogously, $[d]_e$ will denote equivalence class of $d \in \mathbf{Dpc}$.

The functions m: $\mathbf{Qc} \rightarrow \mathbf{Dc}$ and i: $\mathbf{Dc} \rightarrow \mathbf{Qc}$ are the generalizations of the restricted versions of *match* and *index* over equivalence classes. Let $qc \in \mathbf{Dc}$ be some equivalence class from $\mathbf{Qpc}\backslash\equiv_i$ then $m\ (qc)$ is obtained by taking any q from class qc, and taking the equivalence class containing $match\ (q)$. As a result of lemma 1 we have $match\ (q) \in \mathbf{Dpc}$. As a consequence of lemma 2 the resulting class does not depend on the actual q taken from qc. The function i is introduced analogously:

$$m(qc) = [match(q)]_e \qquad \text{for } q \in qc$$
$$i(dc) = [index(d)]_i \qquad \text{for } d \in dc$$

This brings us to a main result of this paper:

Theorem 1
The functions m and i are inverse functions.

Proof:

1. Assume $qc \in \mathbf{Qc}$, and let $q \in qc$. As $q \in \mathbf{Qpc}$, we conclude $index(match(q)) \equiv_i q$, and thus $qc = [index(match(q))]_i$. Consequently, $i(m(qc)) = qc$.
2. Assume $dc \in \mathbf{Dc}$, and let $q \in qc$. As $d \in \mathbf{Dpc}$, we conclude $match(index(d)) \equiv_e d$, and thus $dc = [match(index(d))]_e$. Consequently, $m(i(dc)) = dc$.

Concepts

As we are looking in a dualistic system for sharing of meaning, we concentrate on combinations of intensional and extensional objects. Symmetry in mutual meaning assignment for such combinations is a central issue in text and data mining environments. Such combinations are referred to as concepts.

Definition 1

A pair (qc,dc) is called a concept if: $m(qc) = dc \wedge i(dc) = qc$.

Let $\boldsymbol{C}$ be the set of concepts, then the following is a direct consequence of theorem 1:

Theorem 2

$\boldsymbol{C} = \{(qc,m(qc)) \mid qc \in \mathbf{Qc}\} = \{(i(dc),dc) \mid dc \in \mathbf{Qc}\}$

Concepts consist of an intensional and an extensional part. Concepts may be ordered by the knowledge they reflect, as represented both by their intention and extension. We will not further elaborate on this ordering of concepts, as such an ordering will become meaningful only if some further properties are assumed on the interaction between the functions *index* and *match*. The resulting set of concepts forms the ontology that is implicit for the dualistic system.

Descriptor Approximation

An interesting operator is the approximation of intensional or extensional objects. Let d be some extensional object. Then d is described by intensional object $index(d)$. It is possible, however, that no intensional object can produce this meaning, or: $\forall_q [match(q) \neq d]$. The question then is what descriptors are good approximations of the contents of this query result. An intensional object that produces extensional object d is called a root of d. We call an intensional object q an approximation of extensional object d if it is root of an intensional object similar to d. In other words, the materialization $match(q)$ of this descriptor has similar (intensional) meaning as d (see figure 4).

Definition 2

The set $Approx_e(d)$ of approximations of extensional object d is defined by:

$$Approx_e(d) = \{q \mid index(match(q)) \equiv_i index(d)\}$$

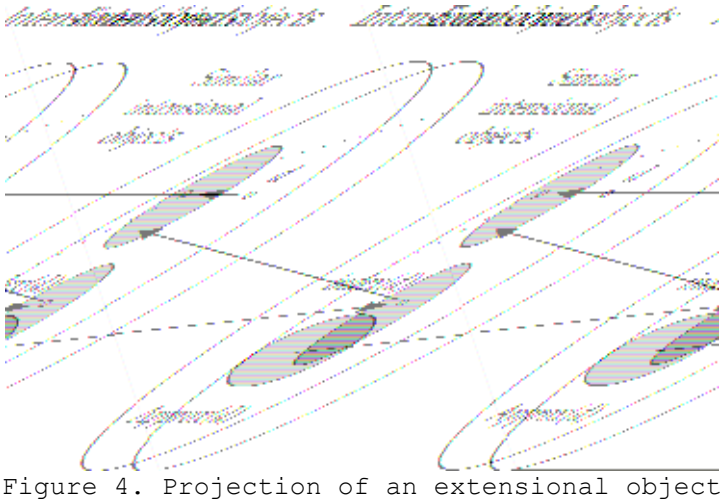

Figure 4. Projection of an extensional object

Analogously we can introduce the approximations of a descriptor q as those extensional objects d that have a mirror $match(index(d))$ similar to materialization of q (see figure 5):

Definition 3

The set $Approx_i(q)$ of approximations of extensional object d is defined by:

$$Approx_i(q) = \{d \mid match(index(d)) \equiv_e match(q)\}$$

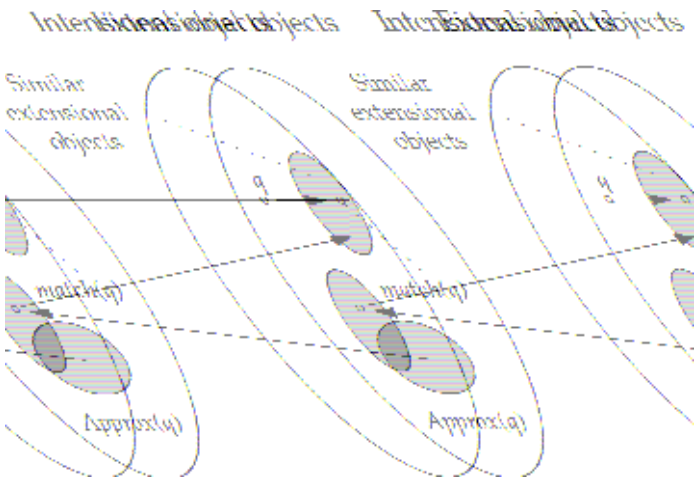

Figure 5. Projection of an intensional object

Approximations are an important feature in a dualistic system. If a searcher would offer a query result as a typical specimen of the information need, then approximations of this query result can be used as a starting point during the process of Query by Navigation (see section 4), supporting the searcher in finding a proper formulation of the information need.

Lemma 3
If query result d and query q are approximations of each other, then $([index(d)]_i,[match(q)]_e)$ is a concept.

Proof

Suppose query result d and query q are approximations of each other, or: $index(match(q)) \equiv_i index(d)$ and $match(index(d)) \equiv_e match(q)$. Then $match(q) \in$ **Dpc** and $index(d) \in$ **Qpc** are easily verified. The result then follows from theorem 2.

In section 4 we will demonstrate how a search engine may employ the nature of dualistic systems. Besides the dual view, this system will benefit from descriptor approximation.

Note that our approach does not cover lower and upper approximations, as for example introduced in rough set theory. The reason is that we do *not* assume a (partial) ordering relation on intensional and extentional objects.

SPECIAL REALIZATIONS

In this section we will discuss in three different realizations of dualistic systems: the vector model, the set model and the fuzzy model. These realizations provide different ways in which the dualistic system can derive its concepts according to the rules of the general model from this section. In the vector model, focus is on finding a minimal set of concepts spanning the conceptual space available in a document collection. With each concept a value is associated that describes the relevancy of that concept in the collection. This provides the opportunity to eliminate concepts that are a consequence of

semantic noise. The set model results in a much more refined look, trying to give a complete view on the concepts in the collection, providing an ontology that describes the nature of concepts in terms of generality. This conceptual view will usually be much larger than the conceptual view obtained by the vector model. However, in cases likes looking for a needle in the haystack, the searcher actually may be looking for rare information that would be interpreted as noise in the vector model approach. The fuzzy model provides the opportunity to balance between granularity and cost of computation.

The Vector Model

Assume a set **D** of documents and a set **T** of terms, and an *aboutness* function A: **D** * **T** → [0,1]. This function *A* is usually represented as a matrix. The value $A_{d,t}$ describes the degree in which document *d* is about term *t*.

In the vector model intensional objects are linear combinations of terms, referred to as document vectors. On the other hand, extensional objects are seen as a linear combination of documents. As a consequence, both intensional and extensional objects are seen as vectors. The equivalence relations $\equiv_i$ and $\equiv_e$ are straightforward: two vectors are considered to be equivalent if they are a (positive) linear combination of each other:

$$x \equiv y \Leftrightarrow \exists_{\lambda>0}\ [x = \lambda y]$$

One might say that *x* and *y* cover the same topic, but only differ in degree of intensity, which is expressed by the scalar λ.

The functions *match* and its dual function *match* are defined as follows:

$$match(q) = Aq$$
$$index(d) = A^T d$$

These functions satisfy the similarity closure assumptions:

Lemma 4

1. $q_1 \equiv_i q_2 \Rightarrow match(q_1) \equiv_e match(q_2)$
2. $d_1 \equiv_e d_2 \Rightarrow index(d_1) \equiv_i index(d_2)$

Proof

1. Suppose $q_1 \equiv_i q_2$, then $q_1 = \lambda q_2$ for some $\lambda > 0$. Consequently: $match\ (q_1) = Aq_1 = \lambda\ Aq_2 = \lambda\ match(q_2)$ and thus $match(q_1) \equiv_e match(q_2)$.
2. Analogously.

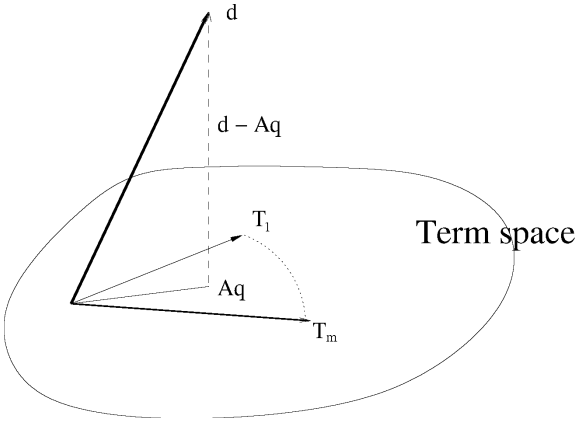

Figure 6. Projecting an extensional object onto term space

A value λ such that $Aq = \lambda d$ and $A^T d = \lambda q$ for non-zero vectors q and d, is called a singular value of matrix A. The vectors d and q are called left-singular and right-singular eigenvectors for singular value λ, respectively. Invariance under subsequent application of *match* and *index* leads to the eigenvectors of A^TA and AA^T respectively:

Lemma 5

1. **Qpc** = $\{q \mid \exists_{\lambda>0}\ [A^TAq = \lambda q]\}$
2. **Dpc** = $\{d \mid \exists_{\lambda>0}\ [AA^Td = \lambda d]\}$

Finding the eigenvalues and eigenvectors of A^TA for a given matrix A is called *Singular Value Decomposition* (SVD), also referred to as *Principal Component Analysis*. This approach, well known as Latent Semantic Indexing in IR research (Deerwester, 1990; Berry, 1995), is commonly used to sort out noise and relevant data. The idea behind this decomposition is that eigenvectors with relatively small eigenvalues can be eliminated (set to 0) without essentially disturbing the relevant data.

The singular value decomposition of a square matrix A results in the following decomposition:

$$A_n = U \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} V^T$$

Where:

1. U is the matrix of left singular vectors, $UU^T = U^TU = I$.
2. Σ_r is a diagonal matrix containing of the roots of the eigenvalues, r is the rank of A^TA.
3. V is the matrix of right singular vectors, $VV^T = V^TV = I$.

The set *Approx(d)* of approximations of extensional object d is described by:

$$Approx(d) = \{q \mid A^TAq = A^Td\}$$

In terms of linear algebra, the vector q is the best solution of the equation $Aq \approx d$. Being the best solution means that (see figure 6) $d - Aq$ is orthogonal on the image space of A (the term space), i.e., $A^TA(d-Aq) = 0$. This optimal query q thus is the solution of the equation $A^TAq = A^Td$. As a consequence, the set *Approx (d)* consists of the projection from d onto term space.

The Set Model

Assume a set **D** of documents and a set **T** of terms, and assume a relation $\sim \subseteq$ **T** * **D**. We write $t \sim d$ to denote that term t describes document d. For example, $t \sim d \Leftrightarrow A_{d,t} > 0$. The tuple <**T**, **D**, ~> is called a *formal context* (see Ganter, 1996). It will be convenient to overload the similarity relation as follows:

$t \sim D \equiv \forall_{d \in D}\ [t \sim d]$
$Q \sim d \equiv \forall_{t \in Q}\ [t \sim d]$
$Q \sim D \equiv \forall_{t \in Q, d \in D}\ [t \sim d]$

While the vector model uses vectors as a grouping mechanism, the set model uses *sets* for this purpose. In the set model, intensional objects thus are sets of

terms, while extensional objects are sets of documents. Like before, intensional objects represent both queries and document meaning, while extensional objects represent the outcome of a search, or describe the meaning of a term. Similarity on intensional and extensional objects is introduced as set equivalence:

$$x \equiv y \Leftrightarrow x = y$$

The function *index* is introduced as the left-polar function:

$$index(D) = \{t \in \mathbf{T} \mid t \sim D\}$$

The function *match* corresponds to the right-polar function:

$$match(Q) = \{d \in \mathbf{D} \mid Q \sim d\}$$

Due to the simplicity of the similarity relation for both intensional and extensional objects, the similarity closure assumptions DS1 and DS2 are trivially satisfied. Notice that *index* and *match* form a *Galois connection*, a pair of reverse order functions between two partially ordered sets.

Note the special similarity relation implies that the set **Qpc** and **Qc** are isomorphic, as is the case with **Dpc** and **Dc**.

Before further focusing on the nature of concepts in this case, we summarize some properties that will be needed (for proofs see Grootjen, 2002). The polar functions introduce mutuality between documents and terms.

Lemma 6

1. $index(D) \sim D$
2. $Q \sim match(Q)$

Both polar functions are non-increasing functions as larger sets have more restrictions for sharing than smaller sets: the larger a set, the less the elements have in common.

Lemma 7

1. $D_1 \subseteq D_2 \Rightarrow index(D_1) \supseteq index(D_2)$
2. $Q_1 \subseteq Q_2 \Rightarrow match(Q_1) \supseteq match(Q_2)$

Mutual sharing of meaning between documents and attributes is a special case. First we provide a better characterization of this situation. In the next section, mutual sharing of meaning will be the basis for the introduction of concepts.

Lemma 8

$Q \sim D \Leftrightarrow D \subseteq match(Q) \Leftrightarrow Q \subseteq match(D)$

The polar functions can be decomposed in terms of elementary set operations. The following property shows how these operations distribute over the polar functions.

Lemma 9

1. $index(D_1 \cup D_2) = index(D_1) \cap index(D_2)$
2. $match(Q_1 \cup Q_2) = match(Q_1) \cap match(Q_2)$

Both document class and term class are extensions of their argument set:

Lemma 10

1. $D \subseteq match(index(D))$
2. $Q \subseteq index(match(Q))$

After these properties we return to the sets **Qpc** and **Dpc**. From each starting point, these sets are encountered after one step:

Lemma 11

1. $match(q) \in$ **Dpc**
2. $index(d) \in$ **Qpc**

Proof
We will only prove the first statement, the second is proven analogously.
From lemma 10:2 we conclude $Q \subseteq index(match(Q))$, and thus by lemma 7 we get: $match(Q) \supseteq match(index(match(Q)))$.
On the other hand, using lemma 10:1, substituting D by $match(A)$, we get: $match(Q) \subseteq match(index(match\ (Q)))$.
As a consequence: $match(index(match(Q))) = match(Q)$.

The set $Approx(d)$ of approximations of extensional object d has been introduced as:

$$Approx(d) = \{q |\ index(match(q)) = index(d)\}$$

Let $index(match(q)) = index(d)$, then also $match(index(match(Q))) = match(Q) = match(index(d))$. So $Approx(d)$ is the set containing the intensional object that approximates to the concept determined by intensional object $index(d)$.

The Fuzzy Set Model

In this section we consider a fuzzy model for information retrieval based on the construction of a fuzzy formal context. The basis for interpreting Information Retrieval in terms of many-valued logics is the introduction of a fuzzy implication. In Rijsbergen (1986) a non classical logic is proposed for information retrieval (see also Crestani, 1995). We will use $\rightarrow_f$ as a generic symbol for fuzzy implementation. Fuzzy implementation is seen as a function with signature [0,1] * [0,1] → [0,1]. $a \rightarrow_f b$ indicates how certain we are over the validity of the implication given how certain we are over its arguments (a and b respectively). Fuzzy logic provides a logics of vagueness (Hájek, 1996). Fuzzy logics may be based on a conjunction operator $t(x, y)$ and an implication operator $i(x, y)$. They form an adjoint couple if $z \leq i(x, y) \Leftrightarrow t(x, y) \leq y$. There are three main variants:

1. Łukasiewicz' logic
   $x \ \& \ y \ = max\ (0, x+y-1)$
   $x \rightarrow_Ł y = min\ (1, 1-x+y)$

2. Gödel's logic
   $x \wedge y \ = min\ (x, y)$
   $x \rightarrow_G y = (x \leq y \rightarrow 1; y)$

3. product logic

$x \odot y = xy$

$x \rightarrow_P y = (x \leq y \rightarrow 1;\ y/x)$

In these logic's, the constants *true* and *false* correspond to 1 and 0 respectively. As in the set model, we assume a set $\mathbf{D}$ of documents and a set $\mathbf{T}$ of terms. The aboutness relation is seen as a fuzzy relation, i.e., for each document d and term t the $A_{d,t}$ describes the degree in which document d is supposed to be about term t. This fuzzy relation may be identified with the aboutness matrix from the vector model.

In our fuzzy model for Information Retrieval, an intensional object is a fuzzy set over terms $\mathbf{T}$, while an extensional object is a fuzzy set of documents $\mathbf{D}$. Intensional and extensional objects are similar when they are equal. The similarity closure assumptions thus obviously are satisfied.

Indexing a set of documents can be seen as finding for each term the degree in which this term is implied by the (fuzzy) collection being indexed. The result of indexing is an intensional object, or a fuzzy set of terms. This may be expressed as:

$$index(D) = \lambda_{t \in \mathbf{T}} [\wedge_d [D(d) \rightarrow_f A_{d,t}]]$$

During matching it is determined to what degree documents are implied by the query. The result is an extensional object, or a fuzzy document set.

$$match(Q) = \lambda_{d \in \mathbf{D}} [\wedge_t [Q(t) \rightarrow_f A_{d,t}]]$$

Small certainty may originate from noise. A threshold ϑ is introduced for the recognition of noise. Scoring above this threshold means acceptance, otherwise the statement is believed to be invalid. In Elloumi (2004) this is effectuated by:

$$match_\vartheta(Q) = \lambda_{d \in \mathbf{D}} [\wedge_t [Q(t) \rightarrow_f A_{d,t} \geq \vartheta]]$$

where the outcome of the comparison operator is to be interpreted using the identities: *true* = 1 and *false* = 0. As a consequence, the result $match_\vartheta(Q)$ is a set of documents.

Using Gödel's logic, the index operator is further elaborated as follows:

$$\begin{aligned} index(D) &= \lambda_{t \in \mathbf{T}} [\wedge_d [D(d) \rightarrow_G A_{d,t}]] \\ &= \lambda_{t \in \mathbf{T}} [min_d\ (D(d) \leq A_{d,t} \rightarrow 1;\ A_{d,t})] \\ &= \text{(in case } D \text{ is a crisp set) } \lambda_{t \in \mathbf{T}} [min_{d \in \mathbf{D}}\ A_{d,t}] \end{aligned}$$

So it seems reasonable to restrict extensional objects to sets of documents. For the match operator we get:

$$\begin{aligned} match_\vartheta(Q) &= \lambda_{d \in \mathbf{D}} [\wedge_t [Q(t) \rightarrow_G A_{d,t} \geq \vartheta]] \\ &= \lambda_{d \in \mathbf{D}} [min_t\ (Q(t) \rightarrow_G A_{d,t} \geq \vartheta \rightarrow 1\ ;\ 0)] \\ &= \lambda_{d \in \mathbf{D}} [min_t\ (A_{d,t} \geq min\ (Q(t),\ \vartheta) \rightarrow 1\ ;\ 0)] \end{aligned}$$

Thus $match_\vartheta(Q)$ corresponds to the (crisp) set $\{d | A_{d,t} < Q(t) \Rightarrow A_{d,t} < \vartheta\}$. So documents should satisfy sufficient information on each term, except if the term is noise in that document. Note that a drawback of this approach is that documents with noisy terms only, will be retrieved.

In Elloumi (2004) a hybrid approach for matching is taken. The conjunction operator is defined as in Gödel's logic, while the implication is substantiated according to Łukasiewicz' logic. The matching operator then is elaborated as follows:

$$match_\vartheta(Q) = \lambda_{d \in \mathbf{D}} [\wedge_t [Q(t) \rightarrow_Ł A_{d,t} \geq \vartheta]]$$
$$= \lambda_{d \in \mathbf{D}} [min_t (min (1, 1 - Q(t) + A_{d,t}) \geq \vartheta)]$$

Thus in this case, $match_\vartheta (Q)$ corresponds to the (crisp) set

$$\{d \mid \forall_t [A_{d,t} < \vartheta \Rightarrow (Q(t) - A_{d,t}) \leq 1 - \vartheta]\}$$

So if a document would fail the requested supply of some term, then the term shortage for this document should be limited by $1 - \vartheta$.

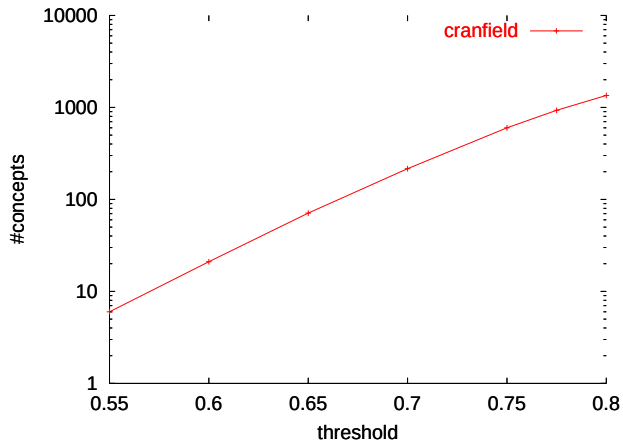

Figure 7. Granularity of concepts

First we note the special case $\vartheta = 1$. In that case the limit for term shortage is so strict that uniform term supply is requested:

$$match_1(Q) = \{d \mid \forall_t [A_{d,t} \geq Q(t)]\}$$

Lemma 12
For $\vartheta = 1$ the fuzzy set model is equivalent to the set model

Proof
For the formal concept we have: $t \sim d \Leftrightarrow A_{d,t} > 0$. If we assume $A_{d,t} \in \{0, 1\}$, then $A_{d,t} \geq Q(t)$ is equivalent with $A_{d,t} = 1$, and therefore with $t \sim d$.

For the case $\vartheta = 1$ all documents will pass the membership test to match the query. The resulting concept lattice thus will contain only 1 concept. The noise threshold may be used to take a position in between. For example, if small

variation is not likely to be a consequence of noise, then ϑ could be chosen near to 1. If a limited number of concepts is required, then a smaller value should be taken for the noise threshold. In figure 7 we see how for an example document collection the number of concepts depends on the noise threshold. Note that the figure suggests an almost linear dependency on a logarithmic scale.

The set *Approx(D)* of approximations of extensional object *D* has been introduced as:

$$Approx(D) = \{Q \mid index(match\ (Q)) = index(D)\}$$

Let *Q* be some query, then the associated query result reflects the degree in which the documents support the query. This query result is indexed as:

$$index(match(Q)) = \lambda_t\ [\wedge_d\ [\wedge_t\ [Q(t) \rightarrow_f A_{d,t}] \rightarrow_f A_{d,t}]\ ]$$

The condition *index(match(Q)) = index(D)* thus is formulated as:

$$\wedge_d\ [\wedge_t\ [Q(t) \rightarrow_f A_{d,t}] \rightarrow_f A_{d,t}] = \wedge_d\ [D(d) \rightarrow_f A_{d,t}]$$

This expression may be further simplified using the rules of the underlying logic.

THE DUAL SEARCH ENGINE

In this section we show how dualistic systems can be used in practice. As an example of the theory we describe the so called *dual search engine*. A simple prototype, called **DUALITY**, is discussed to show its behavior and to provide a flavor of its look and feel.

The general architecture of the dual search engine is presented in figure 8. The search engine internally uses the standard vector model for document representation. The documents are characterized by keywords. A more challenging test would be to use more elaborated characterizations like index expressions (Grootjen 2004), which have shown to perform better as a vehicle for query by navigation than keywords do.

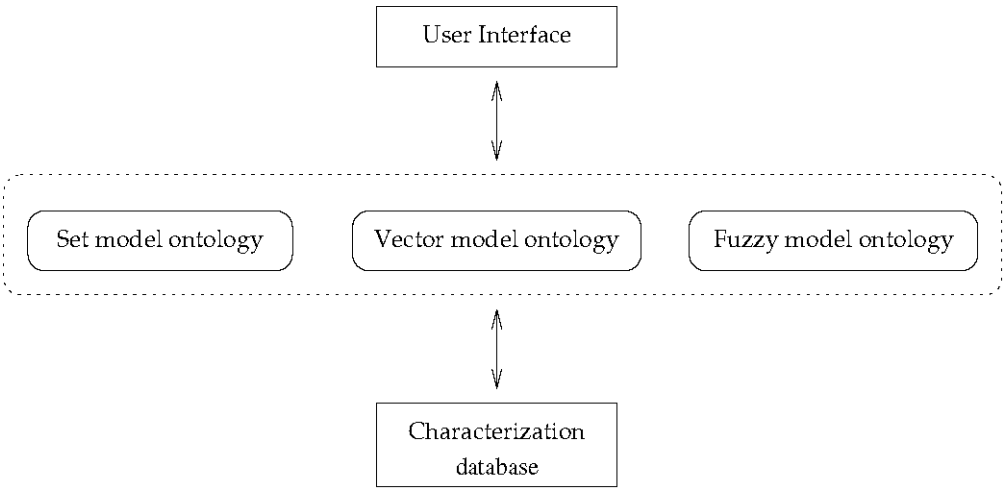

Figure 8. Dual search engine architecture

From the document representations **DUALITY** constructs the ontologies for the vector model approach (section 3.1), the set model approach (section 3.2) and the fuzzy model approach (section 3.3).

A typical state of the dual search engine during interaction with a searcher is an overview that displays both the intensional and extensional object that manifest the searchers current focus. The relation between these two objects is the consequence of the previous step of the searcher.

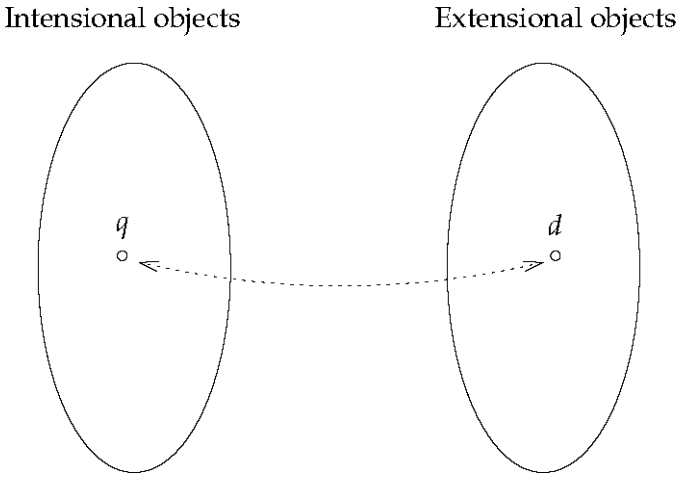

Figure 9. Interaction state of dual search engine

From the current state the engine is ready to process the following kinds of request made by the searcher:

1. shift focus,
2. refine focus,
3. shift conceptual view.

The searcher may shift focus by either entering a new intensional or a new extensional object as base for further exploration.

Q1. After entering a query *q*, the engine evaluates the search result *match(q)*. This will produce the conventional list of documents, ordered by relevancy.

Q2. After entering a weighted set *d* of documents, the engine will produce a common description by evaluating *index(d)*. This will produce a list of terms, ordered by their weight.

Furthermore, the dual search engine makes it possible to refine the current focus by further elaboration on the results obtained. This feedback process is displayed in figure 10. Note that this process has a clear resemblance with the stratified architecture (see Bruza, 1992), as this architecture also has a separation in a hyperindex and a hyperbase. The contrast to the stratified architecture is that the primary focus of the stratified architecture is the support of Query by Navigation. The focus of the dual search engine is on exploiting the ability to switch between hyperbase and hyperindex.

R1. The result *r* of a *match*-operation may be used to create a new query *q* ***Op*** *index(r)*. This new query then is evaluated by the dual search engine, producing *match(q* ***Op*** *index(r))*.

R2. The result *r* of an *index*-operation may be used to create a new weighted set of documents *d* ***Op*** *match(r)*. This new set is evaluated by the dual search engine, producing *index(d* ***Op*** *match(r))*.

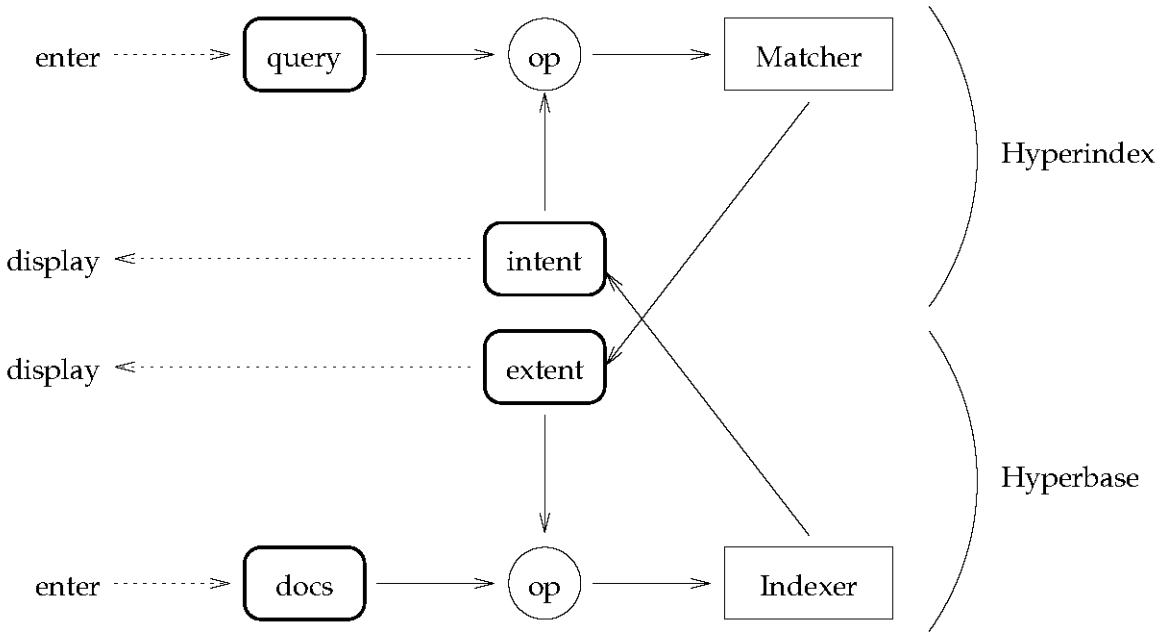

Figure 10. Feedback loop during searching

During a search quest, it may be profitable to change conceptual view. For example from a view directed towards the main semantic components (the vector model approach) to a more complete view (the set model approach). This is especially useful in combination with the approximation functions. The reason for approximation is that there will be no 1-1 correspondence between the concepts from the different conceptual views. In case we want to switch from underlying dualistic model, the following steps are useful:

P1. A query *q* is approximated by a set *Approx(q)* of extensional objects.

P2. An extensional object *d* is approximated by a set *Approx(d)* of queries.

Using these kind of requests, the searcher can employ the dual search engine in several ways.

Query by example A searcher may offer the dual search engine a document *d* that is very much alike the kind of documents wanted. The dual search engine determines these documents by evaluating match (*index({d})*).

The searcher may also use this for relevance feedback, by selecting a relevant subset from the initial query result.

Document contents Offering a document (or a set of documents) to the dual search engine may also be done in order to get an impression of its contents.

Coverage After entering a query *q*, the dual search engine evaluates *match({q})*. After inspecting some document *d*, the searcher might conclude a partial satisfaction of the information need (Bommel, 2005). This can be done by requesting the dual search engine to extract the characterization *index ({d})* from the original query *q*, leading to a new query for evaluation.

In subsection 4.1 we describe the elementary searching process, showing the operations of shifting and refining focus. In subsection 4.2 changing conceptual view is further discussed.

A Sample Session

We now demonstrate how a searcher may perform a search using the dual search engine **DUALITY**. The results are calculated by the **BRIGHT** system (see Grootjen, 2004), an generic tool for experimental Information Retrieval research. The underlying collection is the Cranfield Collection (see Cleverdon, 1967).

Query by Example

Suppose a searcher wants to know about problems associated with high speed aircraft. As an initial attempt the query 'high speed aircraft' is entered into the search engine (figure 2) which produces a classical ranked list of documents (see figure 11). Notice that the output of the search engine has two different panels, one called the *Intensional View* containing the (weighted) entered query keywords, and one called the *Extensional View* which shows the set of ranked documents.

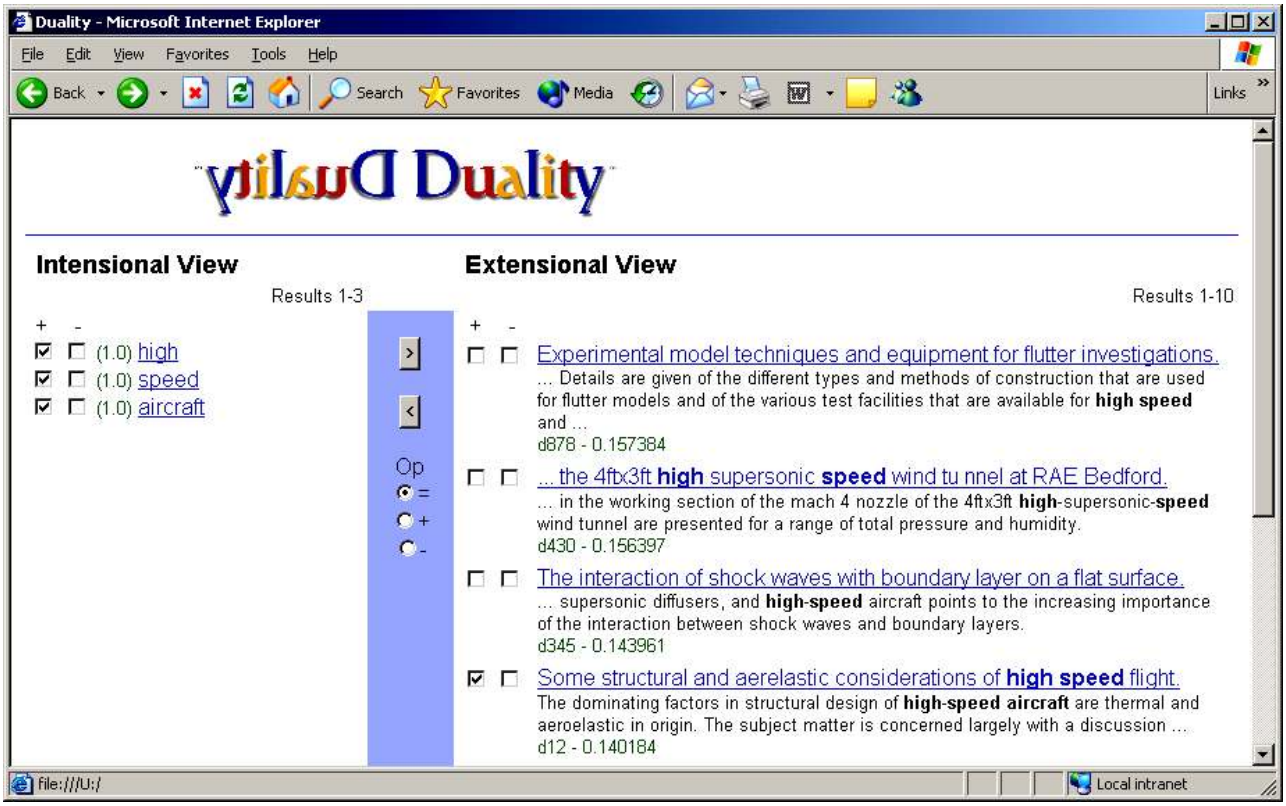

Figure 11. Ranked list

After inspecting the document titles and excerpts of the top 10 ranked documents, the searcher assesses the 4th document (d12) to be relevant, and selects the document's checkbox.

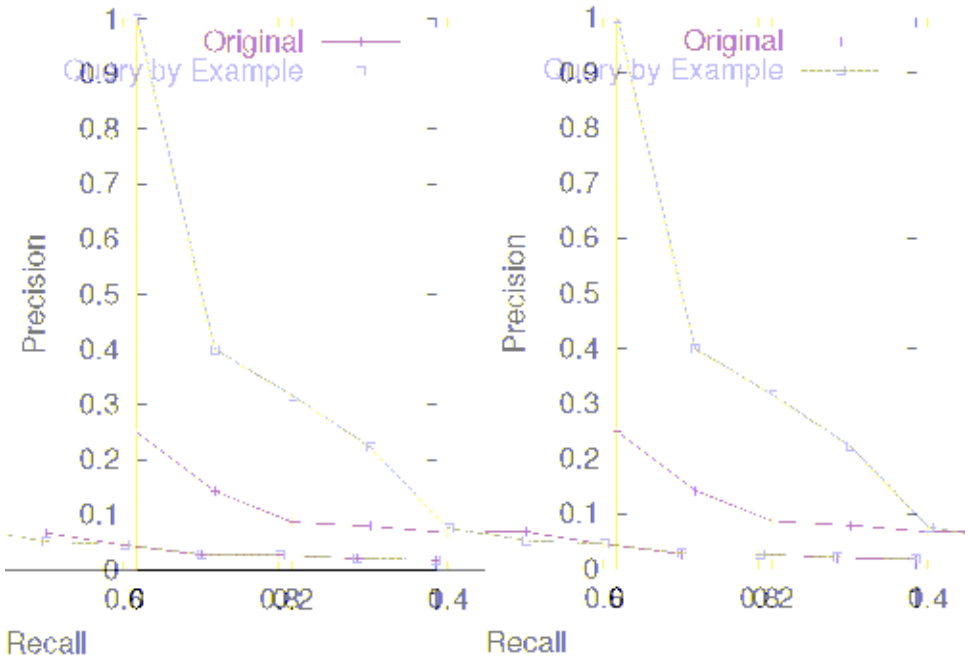

Figure 12. Retrieval performance

Since the selected document covers the desired topic area, the user decides to use 'Query by Example'. This is done in two steps: first the corresponding Intensional object is created by pressing the index button (the button marked with the symbol **>**). This will update the Intensional View panel and shows a new list of weighted terms. The second step is to update the Extensional View panel using these new terms. This is done by pressing the match button (the button marked with the symbol **<**). The result, depicted in figure 13, shows the new list of documents. Since this query is part of the Cranfield Collection, and therefore accompanied by relevance judgments, we can calculate the performance of the retrieval result. Not surprisingly, the performance improved drastically (see figure 12). Note that, in contrast to the example, more than one document can be selected when performing Query by Example.

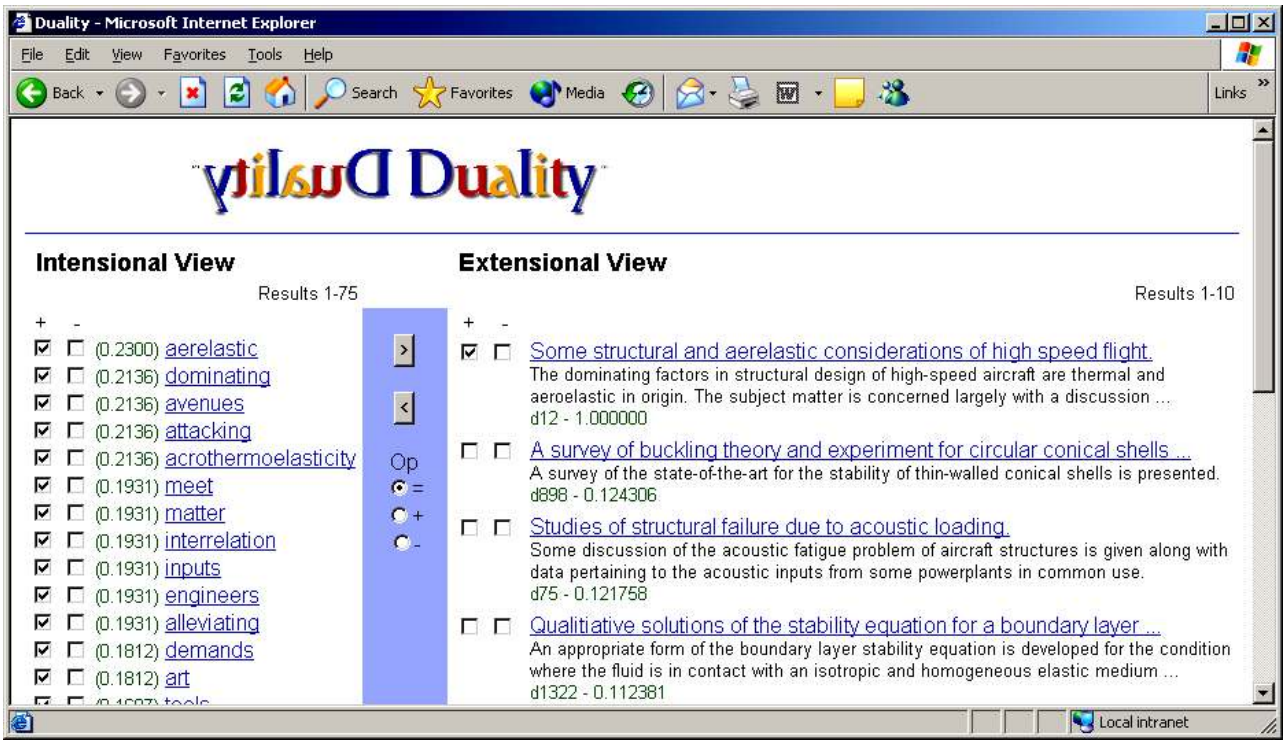

Figure 13. Query by example

Coverage

In the previous example a document selection is used to create a new set of terminals, which is directly used as input to a subsequent match call; the original query terminals are *replaced* by the new ones. **DUALITY** offers the possibility to do more than that: in some cases we don't want the old terminal set to be replaced (**Op =**). So two simple operators are implemented allowing the user to add (**Op +**) or subtract (**Op -**) the index or match result to the current set. Another possibility is to modify the resulting terminal list before invoking the index function: the searcher might add or penalize terminals.

These operators can be used to disambiguate the original query: for example a query about operating systems returns pages about Linux which we want to ignore. Or when our information need is already partly covered, and we are looking for additional (new, residual) information.

Note that after selecting a relevant document, and invoking index with Op + followed by a match is equivalent to applying the standard Rocchio technique for relevance feedback (see for example Baeza-Yates (1999).

Crossing The Boundary

We will conclude this example section by showing the benefits of combining two dualistic ontologies: we will show how the vector space model and the fuzzy model can be combined to yield one powerful search tool.

Suppose a searcher as an information need described as query 173 of the Cranfield collection:

References on Lyapunov's method on the stability of linear differential equations with periodic coefficients.

From the relevance judgment we know that there are only 3 relevant documents in the collection. Assume that during (vector space based) browsing the searcher finds the relevant document d532. Instead of using Query by Example or Relevance feedback, the searcher decides to switch to the fuzzy concept model (In our example, the fuzzy concept lattice is generated with threshold 0.775 and contains 993 concepts}.

Figure 14. Switching to fuzzy concept model

Using the approximation function **DUALITY** presents the fuzzy concept containing d532 with its fuzzy terminal set. One of the extra features of the conceptual model is that the concepts are *ordered*. This enables the user to navigate to related concepts (Query by Navigation). As shown in figure 14 the searcher can beam down to the bottom concept of the lattice or beam up to 3 different superconcepts. After inspecting the terminals presented by the concepts, the searcher decides to beam up to the concept containing both d532 and d367 (which

happen to be both relevant). The process of beaming up increases the extension, and reduces the intention. This is clear when we look at the result (figure 15): the list of terminals is shorter since it covers two documents. In this new fuzzy concept the searcher can beam up to the top concept of the lattice, beam down back to the concept containing only d532, or beam down to the concept of document d367. If the searcher switches back to the vector space model, doing a Query by Example of the two found documents, he will get a ranked document list as extension, with the three relevant documents ranked 1, 2 and 3.

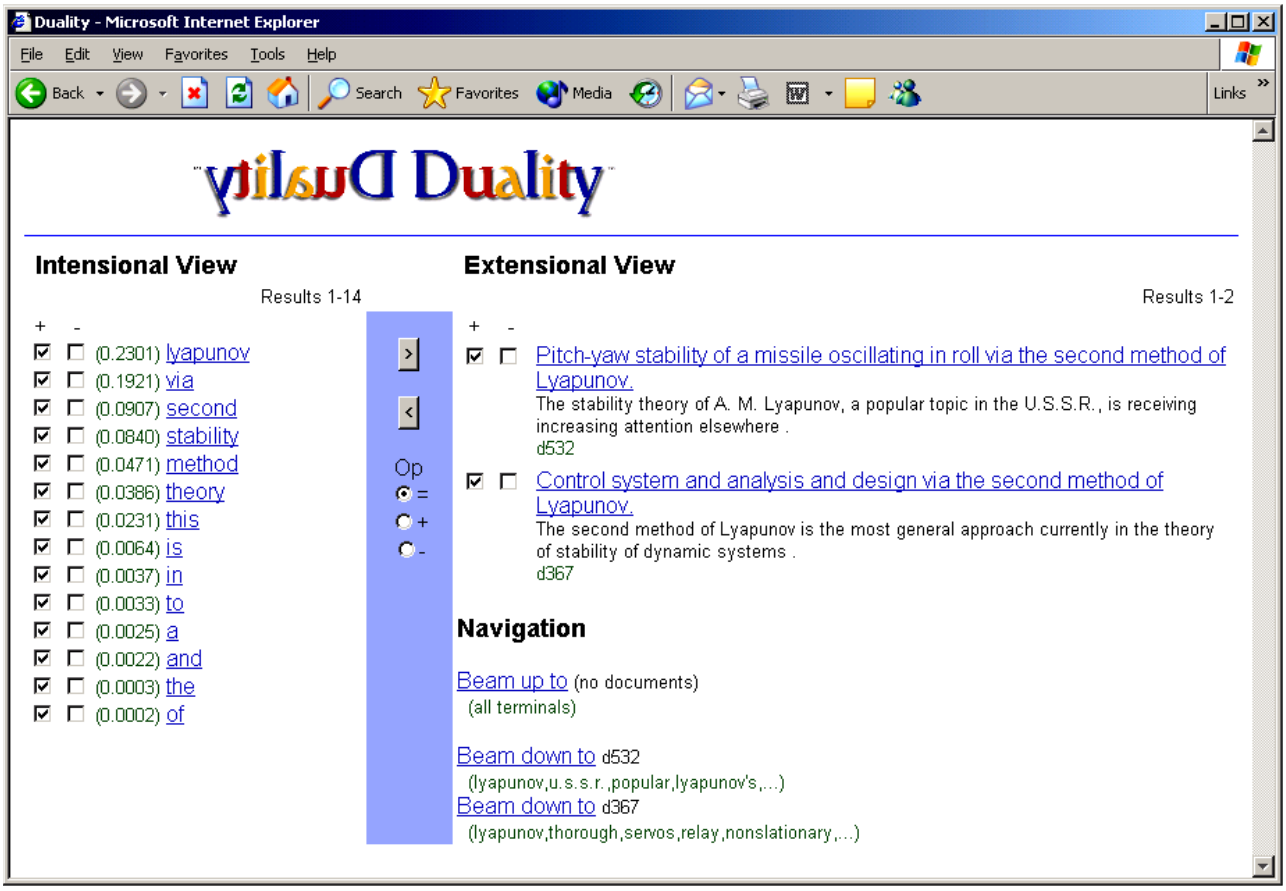

Figure 15. Beaming up

CONCLUSIONS

In this paper we have introduced the concept of dualistic ontologies, discussed properties of such ontologies, and related them to some well-known retrieval models. In order to demonstrate their usefulness, dual search engines have been introduced and illustrated by a sample session of its prototype **DUALITY**.

Further research may be directed towards further elaboration of the theoretical framework, and towards large scale experiments. The prototype **DUALITY** is based on the **BRIGHT** system (see Grootjen, 2004), which has successfully been applied in a large scale TREC experiment. The construction of a concept lattice can become a limiting factor, especially for the set model, as the number of concepts could possibly grow exponentially (in practice smaller upper bounds seem valid). The fuzzy model seems to be a good candidate to scale between completeness and feasibility in that case.

By using a reference set of documents, the dualistic approach can be used to compare different retrieval models based on their indexing and matching algorithm. We feel that this also may be useful in the context of ontology negotiation (see for example Bailin, 2001), where the need for explicit ontologies is crucial for the ontology negotiation process (ONP).

Another application could be in the context of user profiling and collaborative approaches. For example, a dualistic view on customers and purchased products would lead to a conceptual view on customers taste.

Further investigations might consider the question of how the combination of several dualistic ontologies may offer further opportunities for searchers to improve their retrieval effectiveness.

BIBLIOGRAPHY

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*, Addison-Wesley.

Bailin, S. & Truszkowski, W. (2001). Ontology negotiation between scientific archives. *Proceedings of the Thirteenth International Conference on Scientific and Statistical Database Management (SSDBM 2001)*. IEEE Press, July 2001.

Berners-Lee, T. & Hendler, L. & Lassila, O. (2001). The semantic web. *Scientific American,* May 2001, 35-43.

Berry, M.W. & Dumais, S.T. & O'Brien, G.W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4) 573-595.

Bommel, P. van & Weide, Th.P. van der. (2005) Measuring the incremental inforation value of documents. *Information Sciences*, (to appear).

Bruza, P.D. & Weide, Th.P. van der. (1992). Stratified hypermedia structures for information disclosure. *The Computer Journal*, 35(3) 208-220.

Cleverdon, C.W. (1967). The cranfield tests on index language devices. *Aslib Proceedings*, 173-194.

Crestani, F. & Rijsbergen, C.J. van. (1995). Information retrieval by logical imaging. *Journal of Documentation*, 51 3-17.

Deerwester, S.C. & Dumais, S.T. & Landauer, Th.K. & Furnas, G.W. & Harshman, R.A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*. 41(6) 391-407.

Elloumi, S. & Jaam, J. & Hasnah, A. & Jaoua, A. & Nafkha, I. (2004) A multi-level conceptual data reduction approach based in the lukasiewicz implication. *Information Sciences*, 163(4) 253-262, June 2004.

Farquhar, A., Fikes, R., Rice, J. (1996). The ontolingua server: A tool for collaborative ontology construction. Technical Report KSL 96-26, Stanford University, Knowledge Systems Laboratory, Stanford, 1996.

Führ, N. (1989) Models for retrieval with probabilistic indexing. Information Processing and Management, 25(1):55--72, 1989.

Ganter, B. & Wille, R. (1996). *Formale Begriffsanalyse, Mathematische Grundlagen*. Springer-Verlag Berline, 1996.

Grootjen, F.A. & Weide, Th.P. van der. (2002) Conceptual relevance feedback. *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, (NLPKE 2002), Tunis, October 2002.

Grootjen, F.A. & Weide, Th.P. van der. (2004) Effectiveness of index expressions. In *Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004)*}, Springer 165-175, Manchester, June 2004.

Grootjen, F.A. & Weide, Th.P. van der. (2004). The Bright side of information retrieval. Technical Report NIII, Radboud University of Nijmegen, 2004.

Gruber, T.R. (1992). Ontolingua: {A} mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, Stanford, 1992.

Hájek, P. & Godo, L. & Esteva, F. (1996). A complete many-valued logic with product-conjunction. *Archive for mathematical logic*, 35 191-208.

Hearst, M.A., Pedersen, J.O. (1996) Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval, 76-84, 1996.

Noy, N.F. & Sintek, M. & Decker, S. & Crubezy, M. & Fergerson, R.W. & Musen, R.W. (2001). Creating semantic web contents with protege-2000. *IEEE Intelligent Systems*}, 2(16), 60-71.

Rijsbergen, C.J. van. (1986) A non-classical logic for information retrieval. *The Computer Journal*, 29(6) 481-485.

Salton, G. & McGill, M.J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill New York.

Sowa, J.F. (1984) *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, Massachusetts.

Sowa, J.F. (2004) *Guided Tour of Ontology*. http://www.jfsowa.com/ontology/

Spyns, P. & De Bo, J. (2004) Ontologies: a revamped cross-disciplinary buzzword or a truly promising interdisciplinary research topic? Technical Report STAR-2004-20, Vrije Universiteit Brussel.

Taylor, R.S. (1968). Question-negotiation and information seeking in libraries. *College and Research Libraries 1968* 178-194.