Annotations as Context for Searching Documents

Maristella Agosti and Nicola Ferro

Department of Information Engineering – University of Padua,
Via Gradenigo, 6/B – 35131 Padova (PD) – Italy
{maristella.agosti, nicola.ferro}@unipd.it

Abstract. This paper discusses how to exploit annotations as a useful context in order to search and retrieve relevant documents for a user query. This paper provides a formal framework which can be useful in facing this problem and shows how this framework can be employed, by using techniques which come from the hypertext information retrieval and data fusion fields.

1 Introduction

Digital Library Management Systems (DLMSs) are currently in a state of evolution: today they are simply places where information resources can be stored and made available, whereas for tomorrow they will become an integrated part of the way the user works. For example, instead of simply downloading a paper and then working on a printed version, a user will be able to work directly with the paper by means of the tools provided by the DLMS and share their work with colleagues. This way, the user's intellectual work and the information resources provided by the DLMS can be merged together in order to constitute a single working context. Thus, the DLMS is no longer perceived as something external to the intellectual production process nor as a mere consulting tool, but as an intrinsic and active part of the intellectual production process, as pointed out in [1].

Annotations are effective means in order to enable the paradigm of interaction between users and DLMSs envisioned above, since they are very well-established practice and widely used. Annotations are not only a way of explaining and enriching an information resource with personal observations, but also a means of transmitting and sharing ideas in order to improve collaborative work practices. Furthermore, annotations allow users to naturally merge and link personal contents with the information resources provided by the DLMS in order to create a common context that unifies all of these contents.

In fact, annotations allow the creation of new relationships among existing contents, by means of links that connect annotations together and with existing content. In this sense we can consider that existing content and annotations constitute a hypertext, according to the definition of hypertext provided in [2]. This hypertext can be exploited not only for providing alternative navigation and browsing capabilities, but can also offer advanced search functionalities. Furthermore, [3] considers annotations as a natural way of creating and increasing

F. Crestani and I. Ruthven (Eds.): CoLIS 2005, LNCS 3507, pp. 155–170, 2005.

hypertexts that connect information resources in a DLMS by actively engaging users. Finally, the hypertext existing between information resources and annotations enables different annotation configurations, that are *threads of annotations*, i.e. an annotation made in response to another annotation, and *sets of annotation*, i.e. a bundle of annotations on the same passage of text [4, 5].

Thus, annotations introduce a new content layer aimed at elucidating the meaning of underlying documents, so that annotations can make hidden facets of the annotated documents in a more explicit way. In conclusion, we can consider that annotations constitute a special kind of context, that we call *annotative context*, for the documents of a DLMS, because they provide additional content which is related to the annotated documents. This viewpoint about annotations covers a wide range of annotations, ranging from personal jottings in the margin of a page to scholarly comments made by an expert in order to explain a passage of a text. Thus, these different kinds of annotations involve different scopes for the annotation itself and, consequently, different kinds of annotative context. If we deal with a personal jotting, the recipient of the annotation is usually the author himself and so this kind of annotation involves a *private annotative context*; on the other hand, the recipients of a scholarly annotation are usually people who are not necessarily related to the author of the annotation, which thus involves a *public annotative context*; finally, a team of people can work together on a shared topic and can exchange annotations related to the topic in question: thus, in this case we have a *collaborative annotative context*.

In this paper, we aim at exploiting the annotative context in order to use annotations as an effective means for searching and retrieving the documents managed by a DLMS. The presentation is structured as follows: Section 2 introduces an overview of our approach; Section 3 describes our reference architecture; Section 4 presents our framework, which enables the annotations to be effectively employed to search for the documents, and describes an example of data fusion strategy applied to the framework; finally, Section 5 draws some conclusions and gives us an outlook for the future.

2 Search Strategy Overview

Despite all of the research in modelling annotations and providing annotation–enabled systems, there is much less study regarding the usage of annotations for retrieving documents. Golovchinsky et al. [6] compare queries based on annotations with relevance feedback, and considers annotation–based queries as an automatic technique for query construction, since queries are automatically generated from annotated text, e.g. from highlighted text. Frommholz et. al [7] consider annotations – specifically annotations threads – as an extension of the document they belong to, creating a discourse context, in which not only the annotation itself but also its position in the discourse and its type, are exploited for searching and retrieving documents; this approach is revised and extended upon in [8] to probabilistic datalog.

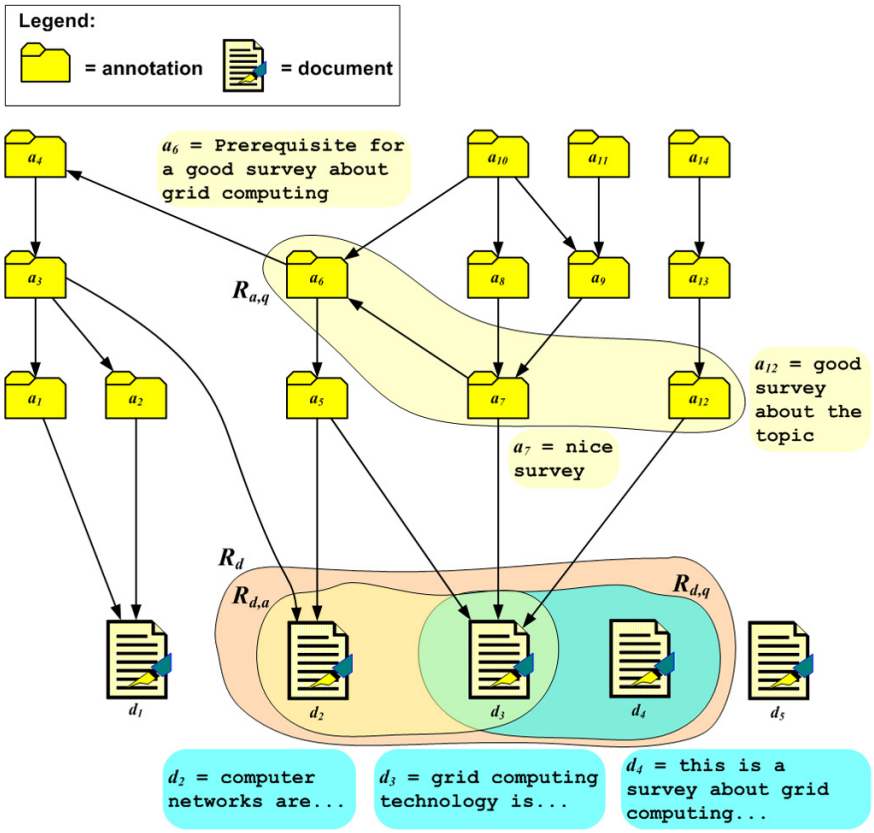

Fig. 1. Example of the document–annotation hypertext used for search purposes

We need to develop a search strategy which is able to effectively take into account the multiple sources of evidence which come from both documents and annotations. In fact, the combining of these multiple sources of evidence can be exploited in order to improve the performances of an information management system. Our aim is to retrieve more documents that are relevant and to have them ranked in a way which is better than a system that does not makes use of annotations.

We will now introduce our search strategy by means of illustrating an example. It is important to note that this is not an exhaustive example, however it will help the reader to familiarize themselves with our search strategy. Figure 1 shows a possible hypertext which could exist among documents and annotations, and which we have called document–annotation hypertext. Suppose that we have the following query: $q =$ "`good survey grid computing`".

Firstly, we can start by searching the set of documents for this query. Let us suppose that we obtain the first result set $R_{d,q} = \{d_4, d_3\}$ ($R_{d,q}$ stands for: Result Documents by Query) where, intuitively, d_4 is ranked higher than d_3 because three query terms out of four are contained in d_4 while d_3 contains only

two terms out of four. However, none of these two documents explains anything about how good the survey is and d_3 does not specify whether the document is a survey or not. Moreover, d_2 is not retrieved because it is concerned with computer networks in general and not with grid computing in particular.

Secondly, we can also search the set of annotations for this query. Suppose that we obtain the second result set $R_{a,q} = \{a_6, a_{12}, a_7\}$ ($R_{a,q}$ stands for: Result Annotations by Query) where, intuitively, a_6 has the highest rank because it contains all of the query terms; a_{12} is ranked lower than a_6 because it contains only two query terms; finally, a_7 has the lowest rank because it contains only one query term. It is worth noting that neither a_7 nor a_{12} explains what the topic of the survey is about, even if they provide additional information about the document they annotate; in a certain sense, it is the symmetric problem with respect to d_3 and d_4, that do not specify that much about the "survey side" of the query. At this point, we have two distinct sources of evidence on hand – the one which comes from the document set and the one which comes from the annotation set – and therefore we should exploit both of them in order to better satisfy the user's information need. Thus, we can exploit them with a twofold aim: firstly, to add new relevant documents to the result set and, secondly, to re-rank the documents in the result set. With this in mind, we can note that:

- the annotations thread $a_6 \rightarrow a_5 \rightarrow d_2$ allows us to connect annotation a_6 to document d_2, suggesting that also document d_2 should be included in the result set. However, d_2 should not be ranked very high because, intuitively, it does not contain any query term and we deduce that it could be related to a survey about grid computing by means of an annotation that is two steps away from d_2;
- the annotations set a_7 and a_{12} regarding document d_3 allows us to understand that d_3 is a survey about grid computing, which is probably a good one. Therefore, we could consider ranking it higher.

Thus, we can identify a third result set $R_{d,a} = \{d_3, d_2\}$ ($R_{d,a}$ stands for: Result Documents by Annotation) where d_3 is ranked higher than d_2 for the reasons explained above. Note that we identified $R_{d,a}$ by means of $R_{a,q}$, that is we found the documents contained in $R_{d,a}$ using the annotations contained in $R_{a,q}$ and the document–annotation hypertext permitted us to pass from annotations ($R_{a,q}$) to documents ($R_{d,q}$).

We can conclude this line of reasoning with the final result set $R_d = \{d_3, d_4, d_2\}$ (R_d stands for: Result Documents). Intuitively, d_3 has the highest rank because it is strongly supported by its own evidence and the evidence provided by the annotations a_7 and a_{12}; in fact, $d_3 \in R_{d,q} \cap R_{d,a}$, as depicted in Figure 1. d_4 keeps its former rank, which is now lower than the rank given to d_3, due to the fact that it is not supported by any further evidence except its own; indeed, $d_4 \in R_{d,q} \setminus R_{d,a}$, as depicted in Figure 1. Finally, we add d_2 which has the lowest rank, due to the fact that it is supported only by the annotation a_6 which, as mentioned above, is not so close to d_2; indeed, $d_2 \in R_{d,a} \setminus R_{d,q}$, as depicted in Figure 1.

In conclusion, annotations provide us with an additional context which can be exploited with the ultimate goal of retrieving more documents that are relevant and better ranked. Furthermore, the document–annotation hypertext is the basic infrastructure which enables us to combine the sources of evidence which derive from documents and annotations. Thus, we face this research problem in the context of *data fusion* [9], because we need to combine the source of evidence which comes from annotations with the one which comes from documents. Moreover, also *Hypertext Information Retrieval (HIR)* techniques [10] are suitable in order to support the search strategy described above, because we need to deal with an hypertext in order to combine the different sources of evidence.

2.1 Search Strategy Issues

The search strategy introduced above presents some issues concerning how to use the document–annotation hypertext in order to identify the annotated documents, specifically regarding how to map $R_{a,q}$ to $R_{d,a}$.

In our previous example, we started from $R_{a,q} = \{a_6, a_{12}, a_7\}$ and we mapped it to $R_{d,a} = \{d_3, d_2\}$; this mapping is not the only possibility: we could also add d_1 to $R_{d,a}$, if we follow the path $a_6 \rightarrow a_4 \rightarrow a_3 \rightarrow a_1 \rightarrow d_1$.

The first issue is that the mapping between $R_{a,q}$ and $R_{d,a}$ is not univocally determined. The second issue concerns the cardinality of R_{d_a}: there is the risk, as shown above, that all the documents that have one or more annotations will be included in the $R_{d,a}$ set, through either a long or a short path. Worst case scenario, we could obtain $R_{d,a} = D$ or, in any case, $|R_{d,a}| \gg |R_{d,q}|$, even though we started with a few annotations retrieved for the query.

Thus, we should add some constraints to the document–annotation hypertext, so that the $R_{d,a}$ set can be unambiguously determined and its cardinality does not increase too much. We will discuss how to overcome these issues in Section 4.

3 Reference Architecture

As explained in the Section 1, annotations create an hypertext that allows users to merge their personal content with the information resources provided by diverse DLMSs: this hypertext can span and cross the boundaries of a single DLMS, if users need to interact with diverse DLMSs. The possibility of having a hypertext that spans the boundaries of different DLMSs is quite innovative because up to now DLMSs do not normally have a hypertext connecting information resources with each other and, if present, such a hypertext is usually confined within the boundaries of a single DLMS. In particular, annotations exploit the hypertext in order to provide users with a *distributed annotative context*, which connects the documents managed by different DLMSs.

We aim at designing and developing a system which is able to carry out the annotative context and the search strategy, previously discussed. We face this problem from an abstract point of view: we do not fully specify how each component of the system works but we describe and define how these components interact with each other. Thus, our architectural approach is based on *flexibility*,

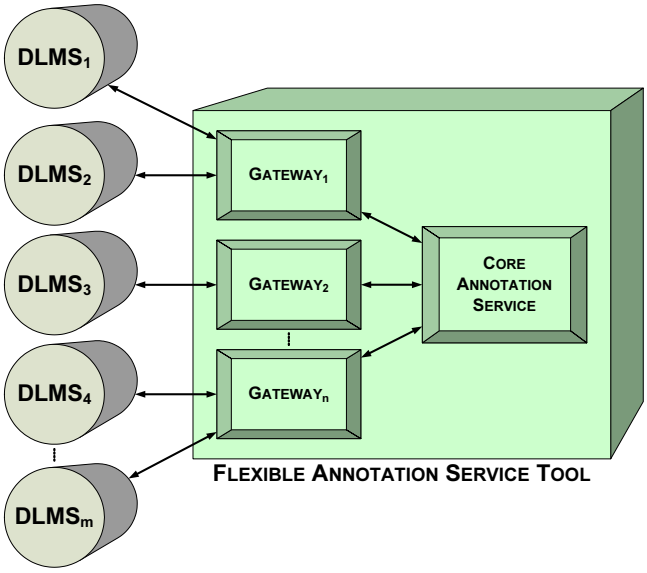

Fig. 2. Overview of the architecture of FAST with respect to different DLMSs

because we need to adopt an architecture which is flexible enough to support a wide range of different DLMSs; thus, we named our target system *Flexible Annotation Service Tool (FAST)*. Figure 2 shows the general architecture of the FAST system and its integration with different DLMSs: the *Core Annotation Service (CAS)* provides annotation management functionalities, and is able to interact with different gateways, that are specialised for integrating the CAS into different DLMSs. From the standpoint of a DLMS the FAST system acts like any other distributed service of the DLMS, even if it is actually made up of two distinct modules, the gateway and the CAS; on the other hand, the FAST system can be made available for another DLMS by creating a new gateway.

As a consequence of this architectural choice, the FAST system knows everything about annotations, however it cannot make any assumption regarding the information resources provided by the DLMS, being that it needs to cooperate with different DLMSs. This architectural choice influences the way in which our search strategy is carried out. Indeed, we aim at combining multiple sources of evidence which come from both documents and annotations. Since the source of evidence concerning the documents is completely managed by the DLMS, FAST has to query the DLMS in order to obtain it. Only after that FAST has acquired this information from the DLMS, it can be combined with the source of evidence which comes from annotations in order to create a list of result documents that better satisfies the user's information needs. In conclusion, we can now deal with a distributed search problem.

4 Search Strategy Framework

In order to carry out the introduced search strategy, we need to deal with two kinds of *Digital Objects (DOs)*, that are documents and annotations. Let D be the *set of documents* and $d \in D$ is a generic document; let A be the *set of annotations* and $a \in A$ is a generic annotation; let $DO = D \cup A$ be the set of digital objects and $do \in DO$ is a generic digital object, which can be either a document or an annotation. Finally, let Q be the *set of user queries* and $q \in Q$ is a generic query. The *Unified Modeling Language (UML)* [11, 12, 13] sequence diagram of Figure 3 summarizes our search strategy:

1. the user submits a query $q \in Q$ to FAST;
2. FAST forwards the query to the DLMS, which searches for documents to retrieve for the query q.
   We call $R_{d,q} \subseteq D$ the result set returned by the DLMS, $s_{d,q} \in [0,1]$ the similarity score of the document d with respect to the query q. According to our architecture, $R_{d,q}$ is completely defined and managed by the DLMS and FAST has no control over $R_{d,q}$. Thus, the DLMS has the function of providing $R_{d,q}$ and a similarity score $s_{d,q}$ for each document $d \in R_{d,q}$ to FAST;
3. FAST searches for annotations to retrieve for the query q.
   We call $R_{a,q} \subseteq A$ the result set returned by FAST, $s_{a,q} \in [0,1]$ the similarity score of the annotation a with respect to the query q.According to our architecture, $R_{a,q}$ is completely defined and managed by FAST;
4. FAST determines the documents associated to the annotations contained in $R_{a,q}$, by using a *mapping function* $\mathrm{M} : A \rightarrow D$, that associates an annotation $a \in A$ to a document $d \in D$.
   We call $R_{d,a} \subseteq D$ the set containing the documents associated to the annotations in $R_{a,q}$, i.e. $R_{d,a} = \mathrm{M}(R_{a,q})$; $s_{d,a} \in [0,1]$ is the similarity score of a document $d \in R_{d,a}$;
5. FAST combines the two sets $R_{d,q}$ and $R_{d,a}$ into one set $R_d = R_{d,q} \cup R_{d,a} \subseteq D$ in order to obtain only one list of retrieved documents. $s_d \in [0,1]$ is the similarity score of a document $d \in R_d$, obtained combining $s_{d,q}$ and $s_{d,a}$;
6. FAST returns the list of retrieved documents to the user.

We can point out some interesting characteristics of this search strategy. Firstly, in the fourth step FAST needs to employ both HIR and data fusion techniques: indeed, the different paths in the hypertext allow FAST to associate annotations to documents, which are necessary to determine $R_{d,a}$ from $R_{a,q}$; furthermore, FAST has to exploit also data fusion techniques in order to compute the similarity score $s_{d,a}$ of a document d from the similarity scores $s_{a,q}$ of the annotations linked to d. Secondly, in the fifth step we need to combine the similarity scores $s_{d,q}$ computed by the DLMS with the similarity scores $s_{d,a}$ computed by FAST, which is a data fusion problem. Finally, the sequence diagram of Figure 3 further highlights that we are dealing with a distributed search problem.

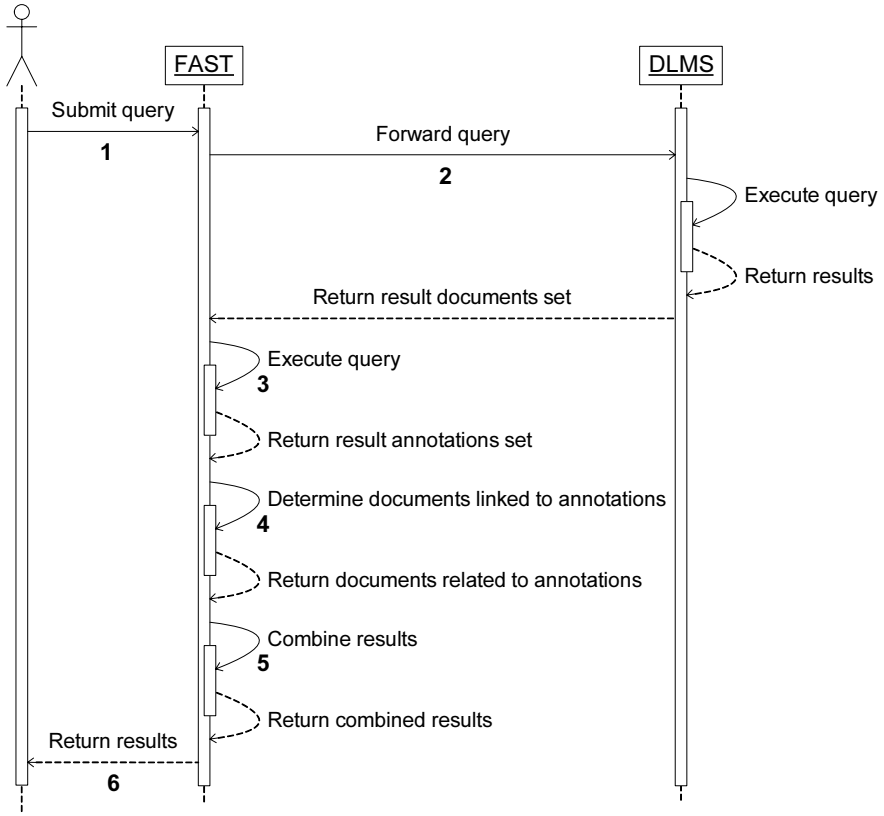

Fig. 3. Search strategy

Note that, as introduced in Section 3, we will face our problem from an abstract point of view. Thus, in the following sections we will not go into a lot of detail on how annotations and documents are indexed and searched, but instead we will assume that there is a component of the system designated with providing such functionalities.

In the next section, we will formally define the basic structure needed to perform our search strategy, which is the document–annotation hypertext; we will also point out some properties of the document–annotation hypertext relevant for our search strategy.

4.1 Document–Annotation Hypertext

Annotations can be linked to DOs with two main types of links, as pointed out in [5]:

– *annotate link*: an annotation annotates a DO, which can be a document or another annotation. The "annotate link" is intended only to allow an

annotation to annotate one or more parts of a given DO. Thus, this kind of link lets the annotation express *intra–DO relationships*, meaning that the annotation creates a relationship among the different parts of the annotated DO;
- *relate-to link*: an annotation relates to a DO, which can be a document or another annotation. The "relate-to link" is intended only to allow an annotation to relate to one or more parts of other DOs, but not the annotated one. Thus, this kind of link lets the annotation express *inter–DO relationships*, meaning that the annotation creates a relationship between the annotated DO and the other DOs that it is related to.

With respect to these two main types of link, we introduce the following constraint: an annotation must annotate one and only one DO, which can be either a document or another annotation, that is an annotation must have one and only one "annotate link". In other words, this constraint means that an annotation can be created only for the purpose of annotating a DO and not exclusively for relating to a DO. Moreover, an annotation can annotate one and only one DO, because the "annotate link" expresses intra–DO relationships and thus they cannot be mutual to multiple DOs which are different from the annotated one. Finally, this constraint does not prevent the annotation from relating to more than one DO, i.e. from having more than one "relate-to link". We can associate to these links a *set of allowed link types* $LT = \{\text{Annotate}, \text{RelateTo}\}$; an element $lt \in LT$ corresponds to one of the link types.

Definition 1. *The* ***document–annotation hypertext*** *is a labeled directed graph* $H_{da} = (DO, E_{da} \subseteq A \times DO)$ *where* DO *is the set of vertices and* E_{da} *is the set of edges. Let* $\mathrm{l}_{da} : E_{da} \to LT$ *be the labelling function. For each* $e = (a, do) \in E_{da}$ *there is a* $\mathrm{l}_{da}(e)$*-labeled edge from the annotation* a *to the generic digital object* do*. The following constraints must be satisfied:*

1. *each annotation* a *must annotate one and only one digital object*[1]*:*

$$\forall a \in A \; \exists! \, e = (a, do) \in E_{da} \mid \mathrm{l}_{da}(e) = \text{Annotate}$$

2. *the graph does not contain loops:*

$$\forall a \in A \nexists \, e = (a, do) \in E_{da} \mid a = do$$

3. *the graph does not contain cycles:*

$$\begin{aligned} &\nexists \, C = a_0 a_k a_{k-1} \cdots a_1 a_0 \mid \\ &\quad e_0 = (a_0, a_k), e_k = (a_k, a_{k-1}), \ldots, e_1 = (a_1, a_0) \in E_{da}, \\ &\quad \mathrm{l}_{da}(e_0) = \mathrm{l}_{da}(e_k) = \ldots = \mathrm{l}_{da}(e_1) = \text{Annotate} \end{aligned}$$

[1] $\exists!$ is the *unique existential quantifier*, and it is read "there exists a unique ... such that ...".

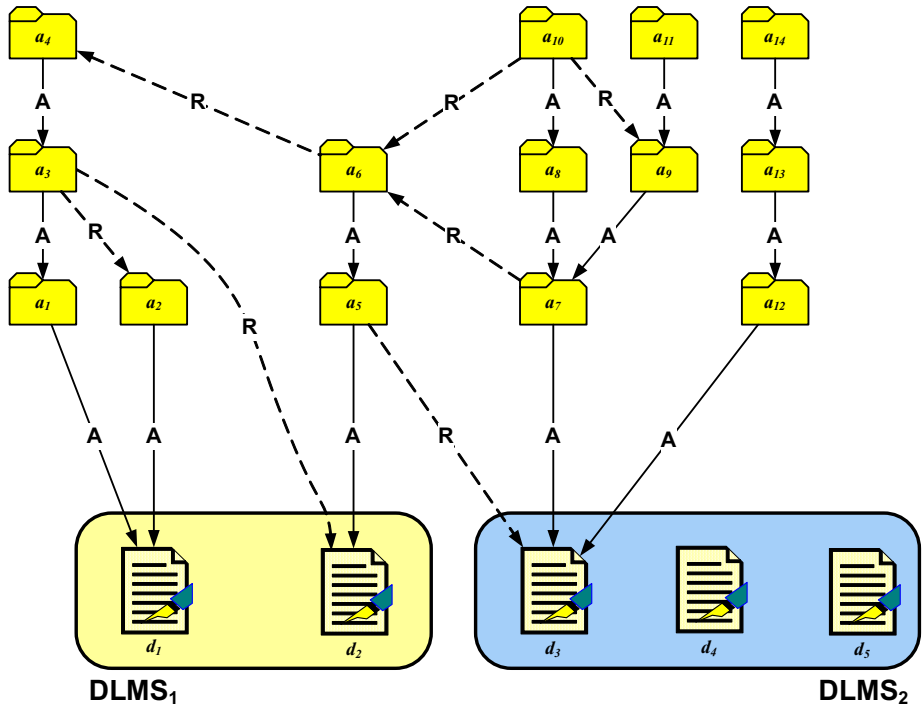

Fig. 4. Example of document–annotation hypertext H_{da}, corresponding at the hypertext shown in figure 1

Note that each $e \in E_{da}$ always starts from an annotation, while $e \in E_{da}$ that starts from a document does not exist. Each annotation is constrained to be incident with one and only one edge with link type "Annotate", thus formalizing the notion of link type mentioned above. The constraint related to loops prevent us from creating self-referencing annotations, which have no use for our purposes. Finally, annotations involve a temporal dimension, since each annotation has to annotate an already existing DO. Thus, the last constraint about cycles of annotations prevents us from creating cycles where the oldest annotation a_0 annotates the newest annotation a_k; note that this is not an issue for document vertices, since "Annotate" links can start only from annotations.

Figure 4 shows an example of document–annotation hypertext, which corresponds to the hypertext show in Figure 1, where the "Annotate links" are represented with a continuous line labeled "A", while the "RelateTo links" are represented with a dotted line labeled "R". Figure 4 also points out another important feature of the document–annotation hypertext: it can span and cross the boundaries of the single DLMS, as discussed in Section 3. The DLMS_1 manages d_1 and d_2, while the DLMS_2 manages d_3, d_4, and d_5. There are annotations that act as a bridge between two DLMSs: for example, a_5 annotates d_2, which

is managed by DLMS_1, and refers to d_3, which is managed by DLMS_2. This is a quite an innovative characteristic of the document–annotation hypertext. This characteristic further highlights the distributed nature of our search strategy, which is not only distributed between the DLMS and FAST, but it may also involve more DLMSs.

The following proposition will show that each annotation a belongs to a unique tree rooted in a document d.

Proposition 1. *Let $H'_{da} = (DO', E'_{da})$ be the subgraph of H_{da}, such that:*

- $E'_{da} = \{e \in E_{da} \mid \mathrm{l}_{da}(e) = \text{Annotate}\}$
- $DO' = \{do \in DO \mid \exists\, e' \in E'_{da}, e' = (a, do)\}$

H'_{da} is the subgraph whose edges are of kind Annotate *and whose vertices are incident with at least one of such edges. Let $H''_{da} = (DO'', E''_{da})$ be the underlying graph of H'_{da}, that is the undirected version of H'_{da}.*

The following properties hold: H''_{da} is a forest[2] *and every tree in H''_{da} contains a unique document vertex d.*

Proof. Ab absurdo: if H''_{da} was not a forest, then it would be a cyclic graph. The only way of obtaining a cycle in H''_{da} is that in H_{da}:

$$\exists\, a \in A,\, \exists\, e_1 = (a, do_1)\,, e_2 = (a, do_2) \in E_{da}, do_1 \neq do_2 \mid \\ \mathrm{l}_{da}(e_1) = \mathrm{l}_{da}(e_2) = \text{Annotate}$$

i.e. an annotation exists in H_{da} from which two *Annotate* edges start from, but this contradicts the definition 1 given for the graph H_{da} and thus, H''_{da} is a forest.

Since H''_{da} is a forest, its components are trees. Ab absurdo suppose that there is a tree T whose vertices are only annotations. A tree T with n vertices has $n-1$ edges but, for the item number 1 of definition 1 each annotation a must be incident with one and only one Annotate edge, then for n annotations there are n edges in H''_{da}; so T can not be a tree. Therefore, every tree in H''_{da} contains, at least, a document vertex d. Suppose now that there is a tree T which contains two document vertices d_1 and d_2, $d_1 \neq d_2$. Being that for every two vertices in a tree there is a unique path connecting them, in the path $P = d_1 a_1 \ldots a_i \ldots a_k d_2$ there must be an annotation a_i from which in H_{da} two edges of kind Annotate start, since by definition of H_{da} the are no edges of the type $e = (d_m, d_n) \in E_{da}$. But the annotation a_i contradicts the definition of H_{da} and thus, there is a unique document vertex d in T. □

Proposition 1 assures us that for each document there is a unique tree T_d that can be rooted in d. Remembering that in a tree any two given vertices are linked by a unique path, for each annotation $a \in R_{a,q}$ we can determine the unique path to the root d of the tree to which the annotation belongs. In this way we can figure out the mapping function M between $R_{a,q}$ and $R_{d,a}$. Finally, we are

[2] A forest is an acyclic graph. A forest is a graph whose components are trees [14].

sure that each annotation $a \in A$ belongs to a tree T_d in H''_{da}, since by definition of H_{da} each annotation must be an incident with one and only one edge e with $\mathrm{l}_{da}(e) = \text{Annotate}$ and thus each annotation $a \in A$ also belongs to H''_{da}.

Note that if we had not removed the "RelateTo link" edges from the graph H''_{da}, it could have contained cycles; consider Figure 4: for example, a cycle would be $C = a_7 a_6 a_{10} a_8 a_7$, because in H''_{da} we do not consider the direction of the edges.

Finally it is worth noting that the document-annotation hypertext of definition 1 lets the mapping function M and the set $R_{d,a}$ overcome the issues described in Section 2.1: firstly, $R_{d,a}$ is unambiguously identified, since proposition 2 ensures us that each annotation $a \in A$ belongs to a unique tree rooted in a document $d \in D$; secondly, the cardinality of $R_{d,a}$ is not too high, since each annotation is connected to only one document and so $|R_{d,a}| \leq |R_{a,q}|$.

Our search strategy consists of several steps: we assume that we have already determined both $R_{d,q}$ and $R_{a,q}$ (respectively, the second and third step of the search strategy), by using the proper information retrieval techniques for indexing and retrieving both documents and annotations; for the fourth and fifth steps it is necessary to define proper algorithms, which are discussed in the following sections.

4.2 Search Strategy Step 4: Hypertext-Driven Data Fusion

We call *hypertext-driven data fusion* the fourth step of our search strategy, because it needs to exploit the document–annotation hypertext in order to compute the similarity scores $s_{d,a}$ for the documents in $R_{d,a}$, that are the documents determined by using annotations, by combining the similarity scores $s_{a,q}$ of the annotations linked to them.

Proposition 1 ensures us that each annotation belongs to a tree rooted in a document. Thus, we can carry out the mapping function M between $R_{a,q}$ and $R_{d,a}$ by simply associating each annotation $a \in R_{a,q}$ to the document d at the root of the tree the annotation belongs to. In this way, $R_{d,a}$ can be unambiguously determined starting from $R_{a,q}$.

Before we can compute $s_{d,a}$ for each document $d \in R_{d,a}$, we need to introduce the notion of *compound similarity score*. To this end, consider the graph $H''_{da} = (DO'', E'')$, a tree T_d rooted in a document $d \in DO''$ and a subtree T_a of T_d rooted in an annotation a. Let $s^c_{a,q}$ be the *compound similarity score* between an annotation $a \in DO''$ and a query $q \in Q$, defined as follows:

$$s^c_{a,q} = \begin{cases} \alpha s_{a,q} & \text{if } a \text{ is a leaf} \\ \alpha s_{a,q} + \frac{(1-\alpha)}{|\mathrm{succ}(a)|} \sum\limits_{a_k \in \mathrm{succ}(a)} s^c_{a_k,q} & \text{if } a \text{ is not a leaf} \end{cases} \tag{1}$$

where $\mathrm{succ}(v_j)$ is a function that returns the set of successors of a vertex v_j and $\alpha \in [0, 1]$ is a parameter. In the following we assume that $s_{a,q}$ is zero for annotations that do not belong to $R_{a,q}$.

$s^c_{a,q}$ recursively computes the weighted average between the similarity score $s_{a,q}$ of an annotation a and the average of the compound similarity scores of its

successors. Furthermore $s^c_{a,q}$ penalizes scores which come from lengthy paths, because for a path $P = a_0 \dots a_k$ the similarity score $s_{a_k,q}$ of a_k is weighted $\alpha(1-\alpha)^k$. Thus $s^c_{a,q}$ satisfies the requirement, expressed in Section 2, that the similarity scores should not be influenced by annotations that are too far apart from the document. Remember that $s_{a,q}$ is not null only for those annotations that belong to $R_{a,q}$; thus annotations, that belong to a path but not to $R_{a,q}$, do not contribute to $s^c_{a,q}$, even if they are taken into account during the averaging by the $|\text{succ}(a)|$ term, thus further penalizing long paths. Equation (1) resembles the `CombANZ` strategy of [15], proposing a recursive version of this strategy, even if `CombANZ` averages only on non-zero similarity scores. In this sense we entitled this section graph-driven data fusion strategy. Example of functions similar to $s^c_{a,q}(a,q)$ can be found in [7, 8, 16], but [7, 8] exploit a probabilistic framework and chooses the path with the maximum probability of the relevance of a document, while [16] does not average the similarity scores and has an iterative approach to the problem.

At this point, for each document $d \in R_{d,a}$ FAST needs to compute its similarity score $s_{d,a}$. If we consider the graph H''_{da}, and for each document $d \in R_{d,a}$ we identify the tree T_d rooted in d, then the similarity score $s_{d,a}$ is given by:

$$s_{d,a} = \frac{1}{|\text{succ}(d)|} \sum_{a \in \text{succ}(d)} s^c_{a,q} \tag{2}$$

where $\text{succ}(v_j)$ is a function that returns the set of successors of a vertex v_j. $s_{d,a}$ simply averages the compound similarity score of the annotations belonging to the tree rooted in d.

4.3 Search Strategy Step 4: Traditional Data Fusion

We call *traditional data fusion* the fifth step of our search strategy, because in this step we compute a similarity score s_d for a document by combining the evidence which comes from $R_{d,q}$ and $R_{d,a}$, as in a usual data fusion problem. With this in mind, we can apply the `CombMNZ` strategy, proposed by [15], as follows:

$$s_d = \begin{cases} 2\,(s_{d,q} + s_{d,a}) & \text{if } d \in R_{d,q} \cap R_{d,a} \\ s_{d,q} & \text{if } d \in R_{d,q} \cap \overline{R_{d,a}} \\ s_{d,a} & \text{if } d \in \overline{R_{d,q}} \cap R_{d,a} \end{cases} \tag{3}$$

If the similarity score $s_{d,q}$ is not normalized, before applying equation (3), we can normalize it according to the expression proposed by [17]:

$$\bar{s}_{d,q} = \frac{s_{d,q} - \min_{d \in R_{d,q}} s_{d,q}}{\max_{d \in R_{d,q}} s_{d,q} - \min_{d \in R_{d,q}} s_{d,q}} \tag{4}$$

4.4 Example of the Search Strategy

Consider the example discussed in Section 2 and shown in Figure 1. Suppose that: $R_{d,q} = \{d_4, d_3\}$ with $s_{d_3,q} = 0.40$, and $s_{d_4,q} = 0.85$; $R_{a,q} = \{a_6, a_{12}, a_7\}$ with $s_{a_6,q} = 0.90$, $s_{a_{12},q} = 0.25$, and $s_{a_7,q} = 0.10$.

In order to carry out the fourth step of our search strategy, i.e. the hypertext-driven data fusion strategy, we start mapping $R_{a,q} = \{a_6, a_{12}, a_7\}$ into $R_{d,a} = \{d_2, d_3\}$. Then, we choose $\alpha = 0.50$, as an example, and, by applying equations (1) and (2), we obtain:

$$s_{d_2,a} = s^c_{a_5,q} = \alpha s_{a_5,q} + (1-\alpha) s^c_{a_6,q} = \alpha(1-\alpha) s_{a_6,q} = 0.23$$
$$s_{d_3,a} = \frac{1}{2}\left(s^c_{a_7,q} + s^c_{a_{12},q}\right) = \frac{\alpha}{2}\left(s_{a_7,q} + s_{a_{12},q}\right) = 0.09$$

In order to carry out the fifth step of our search strategy, i.e. the traditional data fusion strategy, we apply equation (3), obtaining $R_d = \{d_3, d_4, d_2\}$ with:

$$s_{d_2} = s_{d_2,a} = 0.23$$
$$s_{d_3} = 2\left(s_{d_3,q} + s_{d_3,a}\right) = 0.98$$
$$s_{d_4} = s_{d_4,q} = 0.85$$

In conclusion, equations (1), (2), and (3) fit well with the search strategy discussed in Section 2. Indeed, the initial ranking provided the DLMS was d_4, d_3, while the final ranking is d_3, d_4, d_2. Thus, we re-ranked the documents, giving a better rank to d_3 which benefits from the evidence of both documents and annotations, and we also added the new document d_2 to the result list, without ranking it too high, since it has been only added on the basis of the annotations which it is linked to.

5 Conclusions and Future Work

We presented a framework in which annotations can be exploited as a useful context in order to retrieve documents relevant for a user's query. Then, we showed how this framework can be effectively employed for developing search strategies, that adopt techniques which come from the HIR and data fusion fields.

Future research work will be concerned with the application of the proposed search strategy to a real application in order to assess the performances of the proposed search strategy. An obstacle to the evaluation of these kinds of systems is the lack of an experimental test collection with annotations, that would allow us to test and quantitatively compare different search strategies.

Acknowledgements

Sincere thanks are also due to Luca Pretto for the time he spent in discussing the aspects related to document-annotation hypertext.

The work reported in this paper has been conducted in the context of a joint program between the Italian National Research Council (CNR) and the Ministry of Education (MIUR), under the law 449/97-99. The work is also partially supported by the DELOS Network of Excellence on Digital Libraries, as part of the

Information Society Technologies (IST) Program of the European Commission (Contract G038-507618).

References

1. Agosti, M., Ferro, N.: An Information Service Architecture for Annotations. In Agosti, M., Schek, H.J., Türker, C., eds.: Digital Library Architectures: Peer-to-Peer, Grid, and Service-Orientation, Pre-proceedings of the 6th Thematic Workshop of the EU Network of Excellence DELOS, Edizioni Libreria Progetto, Padova, Italy (2004) 115–126
2. Agosti, M.: An Overview of Hypertext. In Agosti, M., Smeaton, A., eds.: Information Retrieval and Hypertext. Kluwer Academic Publishers, Norwell (MA), USA (1996) 27–47
3. Marshall, C.C.: Toward an Ecology of Hypertext Annotation. In Akscyn, R., ed.: Proc. 9th ACM Conference on Hypertext and Hypermedia (HT 1998): links, objects, time and space-structure in hypermedia systems, ACM Press, New York, USA (1998) 40–49
4. Agosti, M., Ferro, N.: Annotations: Enriching a Digital Library. In Koch, T., Sølvberg, I.T., eds.: Proc. 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003), Lecture Notes in Computer Science (LNCS) 2769, Springer, Heidelberg, Germany (2003) 88–100
5. Agosti, M., Ferro, N., Frommholz, I., Thiel, U.: Annotations in Digital Libraries and Collaboratories – Facets, Models and Usage. In Heery, R., Lyon, L., eds.: Proc. 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2004), Lecture Notes in Computer Science (LNCS) 3232, Springer, Heidelberg, Germany (2004) 244–255
6. Golovchinsky, G., Price, M.N., Schilit, B.N.: From Reading to Retrieval: Freeform Ink Annotations as Queries. In Gey, F., Hearst, M., Tong, R., eds.: Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999), ACM Press, New York, USA (1999) 19–25
7. Frommholz, I., Brocks, H., Thiel, U., Neuhold, E., Iannone, L., Semeraro, G., Berardi, M., Ceci, M.: Document-Centered Collaboration for Scholars in the Humanities – The COLLATE System. In Koch, T., Sølvberg, I.T., eds.: Proc. 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003), Lecture Notes in Computer Science (LNCS) 2769, Springer, Heidelberg, Germany (2003) 434–445
8. Frommholz, I., Thiel, U., Kamps, T.: Annotation-based Document Retrieval with Four-Valued Probabilistic Datalog. In Baeza-Yates, R., Maarek, Y., Roelleke, T., de Vries, A.P., eds.: Proc. 3rd XML and Information Retrieval Workshop and the 1st Workshop on the Integration of Information Retrieval and Databases (WIRD2004), `http://homepages.cwi.nl/~arjen/wird04/wird04-proceedings.pdf` [last visited 2004, November 22] (2004) 31–38
9. Croft, W.B.: Combining Approaches to Information Retrieval. In Croft, W.B., ed.: Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval. Kluwer Academic Publishers, Norwell (MA), USA (2000) 1–36
10. Agosti, M., Smeaton, A., eds.: Information Retrieval and Hypertext, Kluwer Academic Publishers, Norwell (MA), USA (1996)

11. OMG: OMG Unified Modeling Language Specification – March 2003, Version 1.5, formal/03-03-01. http://www.omg.org/technology/documents/formal/uml.htm [last visited 2004, November 22] (2003)
12. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading (MA), USA (1999)
13. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading (MA), USA (1999)
14. Diestel, R.: Graph Theory. Springer-Verlag, New York, USA (2000)
15. Fox, E.A., Shaw, J.: Combination of Multiple Searches. In Harman, D.K., ed.: Proc. 2nd Text REtrieval Conference (TREC 2), NIST Special Publication 500–215 (1994) 243–252
16. Savoy, J.: Citation Schemes in Hypertext Information Retrieval. In Agosti, M., Smeaton, A., eds.: Information Retrieval and Hypertext. Kluwer Academic Publishers, Norwell (MA), USA (1996) 99–120
17. Lee, J.H.: Analyses of Multiple Evidence Combination. In Belkin, N.J., Narasimhalu, A.D., Willett, P., Hersh, W., Can, F., Voorhees, E., eds.: Proc. 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1997), ACM Press, New York, USA (1997) 267–276