Annotation Search: The FAST Way

Nicola Ferro

University of Padua, Italy
ferro@dei.unipd.it

Abstract. This paper discusses how annotations can be exploited to develop information access and retrieval algorithms that take them into account. The paper proposes a general framework for developing such algorithms that specifically deals with the problem of accessing and retrieving topical information from annotations and annotated documents.

1 Introduction

Almost everybody is familiar with annotations and has his own intuitive idea about what they are, drawn from personal experience and the habit of dealing with some kind of annotation in every day life, which ranges from jottings for the shopping to taking notes during a lecture or even adding a commentary to a text. This intuitiveness makes annotations especially appealing for both researchers and users: the former propose annotations as an easy understandable way of performing user tasks, while the latter feel annotations to be a familiar tool for carrying out their own tasks. To cite a few examples, if we deal with the Semantic Web, annotations are considered as metadata [20,22]; in collaborative applications annotations are seen as a discourse [14] and might be considered even like e-mails [15]; in the field of digital libraries annotations are treated as an additional content [1,4]; when we talk about scientific databases, annotations represent both provenance information about the managed data and a way of curating the database itself [10]; in the case of data mining and data visualization, annotations are seen as a means for recording the history of user explorations in visualization environments [18]; finally, in social networks and collaborative tagging, annotations are tags or keywords on different kinds of digital content, e.g. photos or bookmarks [16]. It can be noted as these different notions of annotation partially overlap or share commonalities. For a thorough review of the various viewpoints about annotations, please refer to [3,7].

The *Flexible Annotation Service Tool (FAST)* covers many of the uses and applications of annotations discussed above, since it is able to represent and manage annotations which range from metadata to full content; its flexible and modular architecture makes it suitable for annotating general Web resources as well as digital objects managed by different digital library systems; the annotations themselves can be complex multimedia compound objects, with a varying degree of visibility which ranges from private to shared and public annotations and different access rights. The FAST annotation service has proven its flexibility and adaptability to different applicative contexts in many different ways. It has

M. Agosti et al. (Eds.): ECDL 2009, LNCS 5714, pp. 15–26, 2009.

been integrated into the DelosDLMS [2], the prototype of next generation digital library system developed by DELOS[1], the European network of excellence on digital libraries. It has been used as architectural framework for the DiLAS project [1]. Finally, a completely new and re-engineerd version of it has been recently integrated into The European Library (TEL)[2] development portal.

In this paper, we will focus our attention on this improved version of the FAST annotation service and its completely new query processing and retrieval engine, which allows for searching and retrieving annotations and documents according to both structured and unstructured queries.

The paper is organised as follows: Section 2 presents an intuitive introduction to the annotation model supported by the FAST annotation service; Section 3 discusses our search and retrieval framework; Section 4 describes the query language supported by FAST; Section 4.1 provides an example of use of the proposed search and retrieval framework; lastly, Section 5 draws some conclusions.

2 Annotation Model

FAST adopts and implements the formal model for annotations proposed in [7] which has also been utilised in the reference model for digital libraries[3] developed by DELOS [12] and now carried on by DL.org[4], the European coordination action on digital library interoperability, best practices and modelling foundations. In the following, we discuss the proposed model in an intuitive way and by means of examples; the reader interested in its formal foundations can refer to [7].

Annotations are compound multimedia objects constituted by different *signs of annotation* which materialize the annotation itself. For example, we can have *textual signs*, which contain the textual content of the annotation, *image signs*, if the annotation is made up of images, and so on. In turn, each sign is characterized by one or more *meanings of annotation* which specify the semantics of the sign. For example, we can have a sign whose meaning corresponds to the title field in the *Dublin Core (DC)*[5] metadata schema, in the case of a metadata annotation, or we can have a sign carrying a question of the author about a document whose meaning may be "question" or similar. Moreover, an annotation is uniquely identified by a handle, which usually takes the form of a pair (namespace, identifier), where the namespace provides logical grouping of the identifiers, has a scope which defines its visibility, and can be shared with different groups of users.

Annotations can be linked to digital objects with two main types of links. (1) *annotate link*: an annotation annotates a digital object, which can be either a document or another annotation. The "annotate link" is intended to allow an annotation only to annotate one or more parts of a given digital object. Therefore, this kind of link lets the annotation express *intra-digital object relationships*,

[1] `http://www.delos.info/`

[2] `http://www.theeuropeanlibrary.org/`

[3] `http://www.delos.info/ReferenceModel/`

[4] `http://www.dlorg.eu/`

[5] `http://www.dublincore.org/`

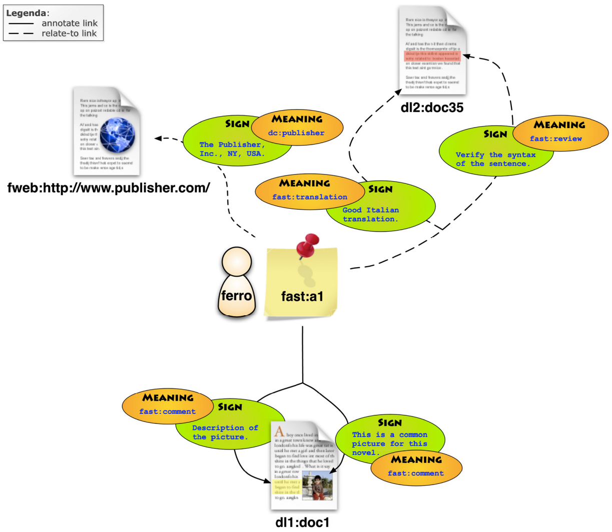
Legenda:
annotate link
relate-to link
dl2:doc35
fweb:http://www.publisher.com/
Sign
The Publisher, Inc., NY, USA.
Meaning
dc:publisher
Sign
Verify the syntax of the sentence.
Meaning
fast:review
Meaning
fast:translation
Sign
Good Italian translation.
ferro
fast:a1
Meaning
fast:comment
Sign
Description of the picture.
Sign
This is a common picture for this novel.
Meaning
fast:comment
dl1:doc1

Fig. 1. Example of annotation according to the FAST annotation model

meaning that the annotation creates a relationship between the different parts of the annotated digital object. (2) *relate-to link*: an annotation relates to a digital object, which can be either a document or another annotation. The "relate-to link" is intended to allow an annotation to relate only to one or more parts of other digital objects, but not the annotated one. Therefore, this kind of link lets the annotation express *inter-digital object* relationships, meaning that the annotation creates a relationship between the annotated digital object and the other digital objects related to it.

Figure 1 shows an example of annotation which summarizes the discussion so far. The annotation, whose identifier is `a1` and which belongs to namespace `fast`, is authored by the user `ferro`[6]. It annotates a document containing a novel, whose identifier is `doc1` and which belongs to the namespace `dl1` of a digital library which manages it. The annotation relates to another document containing a translation of the novel, whose identifier is `doc35` and which belongs to the namespace `dl2` of a digital library different from the one which manages `doc1`; in addition, it also relates to the Web page of the publisher of the novel, whose identifier is `http://www.publisher.com/` and which belongs to the namespace `fweb`, used for indicating Web resources.

In particular, `a1` annotates two distinct parts of `doc1`: it annotates an image by using a textual sign whose content is "`This is a common picture for this novel`" and whose meaning is to be a `comment` in the `fast` namespace; it also annotates a sentence by using another textual sign whose content is "`Description of the picture`" and whose meaning is to be a `comment` in the `fast` namespace. In this

[6] Also the users of the system are identified by using a pair (identifier, namespace) but for space reasons the namespace for the user `ferro` is not reported in the figure and the following discussion.

way, the annotate link is capable of expressing intra-digital object relationships, as discussed above, since it relates different parts of the same annotated digital object.

In addition, `a1` relates the document `doc1` to its Italian translation by linking to the whole document `doc35` with a textual sign whose content is “`Good Italian translation`” and whose meaning is to be a `translation` in the `fast` namespace; it also relates to a specific sentence of the translation with a textual sign which asks to “`Verify the syntax of the sentence`” and whose meaning is to be a `review` in the `fast` namespace. Finally, `a1` also relates the document to the Web page of the publisher of the novel with a textual sign whose content is “`The Publisher, Inc., NY, USA`” and whose meaning is to be the `publisher` field of the DC metadata schema. In this way, the relate-to link is able to express inter-digital object relationships since, in this example, it relates to a novel with both its Italian translation and its publisher Web page.

This example illustrates how annotations can range from metadata to content and they can also mix metadata and content, as in the case of annotation `a1`. The example also shows another important feature of the annotations: they can take a part of a hypertext, as observed by [9,19,23], since they enable the creation of new relationships among existing digital objects, by means of links that connect annotations together with existing digital objects. The hypertext between annotations and annotated objects can be exploited for providing alternative navigation and browsing capabilities. In addition, it can span and cross the boundaries of the single digital library, if users need to interact with the information resources managed by diverse information management systems. In the example above, the annotation `a1` links the document `doc1` which belongs to the digital library `dl1` to both the document `doc35` which belongs to the digital library `dl2` and the Web page `http://www.publisher.com/`. This latter possibility is quite innovative, because it offers the means for interconnecting various digital libraries in a personalized and meaningful way for the end-user and this is a major challenge for the next generation digital libraries [21] and their interoperability [8].

3 The FAST Search Framework

The problem of information access and retrieval by exploiting annotations comprises two different issues: the first concerns the search and retrieval of the annotations themselves; the second regards the search and retrieval of annotated documents. The first case requires the design and development of algorithms able to express complex queries which take into account both the different features of the annotations and the context to which annotations belong. The second case calls for algorithms able to estimate the relevance of annotated documents with respect to a user information need on the basis of the annotations on them.

Previous research has addressed this topic from different viewpoints. [13] deals with the problem of full-text retrieval by providing proximity search operators able to match queries “where some of the words might be in the main text and others in an annotation". [17] intends annotations as a kind of automatic query expansion by adding text from highlighted passages in a document. Both these examples consider annotations as instrumental to document retrieval. [11] proposes a "social validation" approach where the degree of global confirmation or refutation of a debate carried out by annotations is evaluated; this example mainly focuses on the information access for annotations themselves. Lastly, [15] proposes an object-oriented, probabilistic logical model which takes into account annotation polarity, as in the previous case, in order to search and retrieve both documents and annotations; this latter example takes into consideration both aspects discussed above.

We propose a framework which allows users to express both simple free-text queries and complex semi-structured queries in order to retrieve either annotations or documents. The objective of our work is to design a framework able to present relevant results (documents or annotations) based on their topicality without considering, at the same time, their polarity, as done in [11] and [15]. Both positive and negative arguments can be considered as relevant to the topic entailed by a user information need and we would like to avoid "hard wiring" the reasoning about polarity in the retrieval process. For example, some retrieval tasks, such as ad-hoc search or bibliographic search, may need to be carried out without taking into consideration polarity of possible annotations on relevant documents or catalogue records. Moreover, for many retrieval tasks, users have acquired an intuitive mental model of what is going on and this helps them to easily interpret the retrieved results and reformulate the query. The introduction of reasoning about polarity requires a different mental model for the users and this might impair the search and retrieval process if the users do not easily understand how documents and annotations are ranked. We therefore prefer to treat polarity as an advanced option, that the user may apply on demand, and leave it as a possible extension to the present search and retrieval framework.

When we design search functionalities that allow users to retrieve annotations and documents according to their information needs, we have to take into consideration two key points:

- *annotations are constituted by both structured and unstructured content*: information such as the author or the media type of the annotation is part of the structured content of the annotation, while the comments and remarks contained in the body of the annotation are unstructured content;
- *documents and annotations constitute a hypertext*: annotations enable the creation of new relationships among existing documents, by means of links that connect annotations together and with existing documents.

The presence of both structured and unstructured content within an annotation calls for different types of search functionalities, since structured content can be dealt with exact match searches while unstructured content can be dealt with best match searches. These two different types of searches may need to be merged together in a query if, for example, the user wants to retrieve annotations by a given author about a given topic; this could be expressed by a boolean AND query which specifies both the author and the content of the annotations to be searched. Nevertheless, boolean clauses are best suited for dealing with exact match searches and they need to be somewhat extended to also deal with best match searches. Therefore, we need to envision a search strategy able to express complex conditions that involve both exact and best match searches. This is discussed in section 3.1 where the proposed approach stems from our previous work [6] which points out the need for both structured and unstructured search with annotations.

The hypertext that connects documents to annotations calls for a search strategy that takes it into consideration and allows us to modify the score of annotations and/or documents according to the paths in the hypertext. For example, we could consider that an annotation, retrieved in response to a user query, is more relevant if it is part of a thread where other annotations have also been retrieved in response to the same query rather than if it is part of a thread where it is the only annotation that matches the query. This is discussed in section 3.2 and revises and improves our previous work [5], where we proposed an initial algorithm for taking into account the hypertextual nature of annotations.

3.1 Annotation Extended Boolean Retrieval

The P-norm extended boolean model proposed by [24] is capable of dealing with and mixing both exact and best match queries, since it is an intermediate between the traditional boolean way of processing queries and the vector space processing model. Indeed, on the one hand, the P-norm model preserves the query structure inherent in the traditional boolean model by distinguishing among different boolean operators (and, or, not); on the other hand, it allows us to retrieve items that would not be retrieved by the traditional boolean model due to its strictness, and to rank them in decreasing order of query-document similarity. Moreover, the P-norm model is able to express queries that range from pure boolean queries to pure vector-space queries, thus offering great flexibility to the user.

Consider a set of terms $t_1, t_2, \ldots, t_n$ and let $\mathrm{sim}(a, t_i) \in [0, 1]$ be the similarity score of term t_i with respect to annotation a; $\mathrm{sim}(a, t_i) = 0$ if the term t_i is not present in the annotation a.

Let $p \geq 1$ be a real number indicating the degree of strictness of the boolean operator. A generalized **or**-query is expressed as $q_{\mathbf{or}(p)} = [t_1 \,\mathbf{or}^p\, t_2 \,\mathbf{or}^p \cdots \mathbf{or}^p\, t_n]$; a generalized **and**-query is expressed as $q_{\mathbf{and}(p)} = [t_1 \,\mathbf{and}^p\, t_2 \,\mathbf{and}^p \cdots \mathbf{and}^p\, t_n]$. The **extended boolean similarity scores** between an annotation and a query are defined as:

$$\mathrm{sim}_p^{\mathbf{or}}(a, q) = \left[\frac{\mathrm{sim}(a, t_1)^p + \mathrm{sim}(a, t_2)^p + \cdots + \mathrm{sim}(a, t_n)^p}{n}\right]^{\frac{1}{p}}$$

$$\mathrm{sim}_p^{\mathbf{and}}(a, q) = 1 - \left[\frac{(1 - \mathrm{sim}(a, t_1))^p + (1 - \mathrm{sim}(a, t_2))^p + \cdots + (1 - \mathrm{sim}(a, t_n))^p}{n}\right]^{\frac{1}{p}}$$

where t_i indicates a generic term of the query q. Note that for **not**-queries you have to substitute $1 - \mathrm{sim}(a, t_i)$ to $\mathrm{sim}(a, t_i)$ as term weight.

By varying the value of p between 1 and ∞, it is possible to obtain a query processing intermediate between a pure vector-processing model ($p = 1$) and a traditional boolean processing ($p = \infty$), as discussed in [24] to which the reader can refer for further details.

3.2 Annotation Hypertext-Driven Retrieval

Consider the document-annotation hypertext $H_{da} = (DO, E)$ where DO is a set of digital objects (either documents or annotations) and E is a set of edges indicating that an annotation is annotating a digital object, as introduced in [5,7].

The **hypertext similarity score** between an annotation and a query is defined as:

$$\mathrm{sim}_{\alpha}^{ht}(a, q) = \frac{1}{\alpha}\mathrm{sim}(a, q) + \frac{\alpha - 1}{\alpha} \cdot \frac{1}{|\mathrm{succ}(a)|} \sum_{a_k \in \mathrm{succ}(a)} \frac{\mathrm{sim}(a_k, q) + \mathrm{sim}_{\alpha}^{ht}(a_k, q)}{2}$$

where $\mathrm{sim}(a, q) \in [0, 1]$ is a generic similarity function between an annotation and a query, $\mathrm{succ}(a)$ is a function that returns the set of successors of an annotation a_j and α is a real number called the *annotation thread damping* factor. We consider that $\mathrm{sim}(a_j, q) = 0$ for those annotations that do not match the query.

$\mathrm{sim}_{\alpha}^{ht}(a, q)$ computes the weighted average between $\mathrm{sim}(a, q)$, the similarity score of an annotation with respect to a query, and the similarity scores which come from the thread to which the annotation belongs. In particular, the thread similarity scores are given by the average between the similarity scores of the successors of a and the hypertext similarity scores of the successors of a; in other words, the hypertext similarity score recursively averages the similarity scores of the annotations that belong to the same thread of the given annotation a. Furthermore, $\mathrm{sim}_{\alpha}^{ht}(a, q)$ penalizes similarity scores which come from lengthy paths, because for a path $P = a_0 \dots a_k$ the similarity score $\mathrm{sim}(a_k, q)$ of a_k is weighted $\frac{1}{2^k}$.

By varying the value of α between 0 and ∞, it is possible to obtain a query processing intermediate between a traditional information retrieval model ($\alpha = 1$), when $\mathrm{sim}_{1}^{ht}(a, q) = \mathrm{sim}(a, q)$ and only the similarity between the annotation and the query is taken into account, and a pure hypertext driven retrieval model ($\alpha = \infty$), when $\mathrm{sim}_{\infty}^{ht}(a, q) = \frac{1}{|\mathrm{succ}(a)|} \sum_{a_k \in \mathrm{succ}(a)} \frac{\mathrm{sim}(a_k, q) + \mathrm{sim}_{\infty}^{ht}(a_k, q)}{2}$ and only the thread to which the annotation belongs is taken into account.

Note that the above definition of the hypertext similarity score provides us with an additional degree of freedom in the choice of the actual function to be used for computing the similarity between a term and the annotation, which allows us to plug in different retrieval models for annotations besides the one based on extended boolean retrieval proposed in Section 3.1.

Finally, the hypertext-driven retrieval model allows us to compute a similarity score also for the documents that have been annotated, so that it is possible to search and retrieve documents in response to a user query by means of their annotations. The **similarity score by annotation** between the document and a query is defined as:

$$\mathrm{sim}_{\alpha}^{a}(d,q) = \frac{1}{|\mathrm{succ}(d)|} \sum_{a \in \mathrm{succ}(d)} \mathrm{sim}_{\alpha}^{ht}(a,q)$$

Basically, the similarity score by annotation of a document averages the hypertext similarity scores of the annotations that are annotating the document.

4 The FAST CQL Context Set

The search framework discussed in the previous section has been fully implemented in the running prototype of FAST by developing a query language based on the *Contextual Query Language (CQL)* syntax[7], version 1.2, which provides conformance to CQL up to level 2. The FAST Context Set supports all the search capabilities discussed in the previous section and provides users with a uniform query syntax.

Indexes. The FAST context set defines about 40 indexes and corresponds to the annotation and hypertext model discussed in Section 2, according to which annotations and documents can be searched. For example, `fast.annotation.text` matches annotations on the basis of their textual content and provides full text search capabilities in 14 different languages; `fast.document.byAnnotation` enables the search for documents on the basis of a complete CQL query on their annotations; `fast.annotation.language` indicates a search for annotation based on their language; `fast.annotation.author.email` indicates a search for annotation based on the email address of their author, and so on.

Relations and Relations Modifiers. The FAST context set does not introduce new relations but relies on those defined in the grammar of CQL. Nevertheless, it introduces the following relation modifier, called `thread`, which expressed the kind of hypertext similarity strategy to be applied when evaluating the relations, as discussed in Section 3.2.

The `thread` relation modifier can assume the following values: `noThread` ($\alpha = 1$), i.e. no hypertext structure has to be taken into account into the scoring but only the content of the digital object; `halfThread` ($\alpha = 2$), i.e. the hypertext structure influences the scoring as much as the content of the digital object; `almostThread` ($\alpha = 5$), i.e. the hypertext structure influences the scoring much more than the content of the digital object; `onlyThread` ($\alpha = \infty$), i.e. only the hypertext structure has to be taken into account for the scoring and not the content of the digital object.

Finally, the `*` and `?` wildcards are supported to mask zero or more than one characters, the former, and a single character, the latter.

[7] `http://www.loc.gov/standards/sru/specs/cql.html`

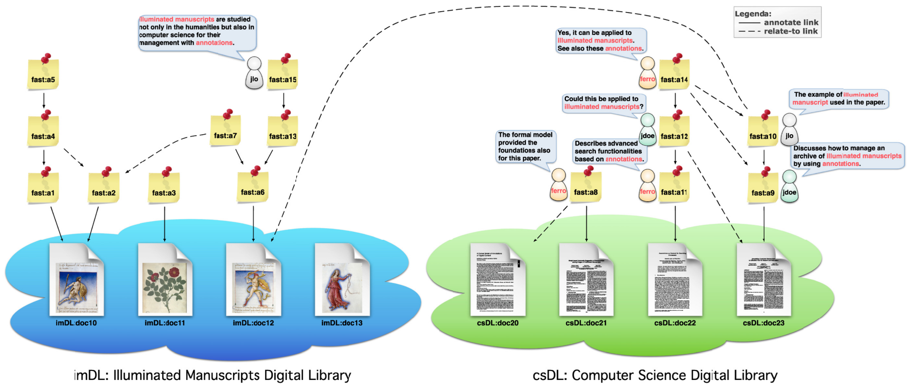
Illuminated manuscripts are studied not only in the humanities but also in computer science for their management with annotations.
jlo
fast:a5
fast:a15
fast:a4
fast:a7
fast:a13
fast:a1
fast:a2
fast:a3
fast:a6
imDL:doc10
imDL:doc11
imDL:doc12
imDL:doc13
imDL: Illuminated Manuscripts Digital Library
Legenda:
annotate link
relate-to link
Yes, it can be applied to illuminated manuscripts. See also these annotations.
ferro
fast:a14
Could this be applied to illuminated manuscripts?
jdoe
fast:a12
The example of illuminated manuscript used in the paper.
fast:a10
jlo
The formal model provided the foundations also for this paper.
Describes advanced search functionalities based on annotations.
Discusses how to manage an archive of illuminated manuscripts by using annotations.
ferro
fast:a8
ferro
fast:a11
fast:a9
jdoe
csDL:doc20
csDL:doc21
csDL:doc22
csDL:doc23
csDL: Computer Science Digital Library

Fig. 2. Example of annotation search

Boolean Operators and Boolean Modifiers. The FAST context set does not introduce new boolean operators, as these can only be defined by the CQL grammar. Nevertheless, it introduces the following boolean modifier, called `match`, which specifies the kind of matching to be applied when computing the boolean expression, according to the different extended boolean retrieval strategies discussed in Section 3.1.

The `match` relation modifier can assume the following values: `bestMatch` ($p = 1$), i.e. a best matching has to be performed; `looseMatch` ($p = 2$), i.e. a very approximate matching has to be performed; `fuzzyMatch` ($p = 5$), i.e. a fuzzy matching has to be performed; `exactMatch` ($p = \infty$), i.e. a strict boolean matching has to be performed.

4.1 Annotation Search Example

Consider the following example, shown in figure 2: the user is interested in finding annotations that deal with digital archives for illuminated manuscripts. In particular, the user is interested in the use of annotations for linking illuminated manuscripts and remembers that the author corresponding to the user `ferro` worked in this field. This information need could be expressed with the following query: $q = \texttt{illuminated manuscript}\,\mathbf{and}^p\;(\texttt{ferro}\,\mathbf{or}^p\;\texttt{annotation})$ which looks for annotations whose content is about $t_1 = \texttt{illuminated manuscripts}$ and either are authored by user $t_2 = \texttt{ferro}$ or also talk about $t_3 = \texttt{annotation}$s.

We can therefore compute the similarity between an annotation and the query as follows:

$$\mathrm{sim}_{\alpha}^{p}\left(a,q\right) = 1 - \left[\frac{\left(1 - \mathrm{sim}_{\alpha}^{ht}(a,t_1)\right)^p + \left(1 - \left(\frac{\mathrm{sim}_{\alpha}^{ht}(a,t_2)^p + \mathrm{sim}_{\alpha}^{ht}(a,t_3)^p}{2}\right)^{\frac{1}{p}}\right)^p}{2}\right]^{\frac{1}{p}}$$

where we used the hypertext similarity scores of section 3.2 to compute the similarity between an annotation and the different terms of the query and we used the extended boolean operators of section 3.1 to combine the results of the previous computations. Note that the above equation expresses a whole family of similarity scores, since it is parametric in both p and α.

Let us use the following weights: for the exact match clause, a weight 1 indicates that `ferro` is the author of the annotation, 0 that somebody else is the author; in the best match clause, the normalized term frequency (tf) / inverse document frequency (idf) weights are used: $w_{ij} = \frac{\mathrm{tf}_{ij} \cdot \log_2 \frac{N}{\mathrm{df}_i}}{\max_{h,k}(\mathrm{tf}_{hk}) \cdot \max_h \left(\log_2 \frac{N}{\mathrm{df}_h} \right)}$, where w_{ij} is the weight of term i in annotation j; tf_{ij} is the frequency of term i in annotation j (in our case it is always 1); df_i is the document frequency of term i, i.e. the number of annotations in which the term i is present (4 for the term `annotation` and 5 for the term `illuminated manuscript`); N is the total number of annotations, i.e. 15. We use $\mathrm{sim}\,(a_j, t_i) = w_{ij}$.

For example, if we set $p = 2$ (`looseMatch` boolean modifier) and $\alpha = 2$ (`halfThread` relation modifier), which correspond to the following query according to the FAST context set and the CQL syntax:

```
(fast.annotation.text =/thread=halfThread "illuminated manuscript")
 and/match=looseMatch
(
  (fast.annotation.author.identifier =/thread=halfThread ferro)
   or/match=looseMatch
  (fast.annotation.text =/thread=halfThread annotation)
)
```

we obtain the following ranking: `fast:a12` with score 0.52; `fast:a9` with score 0.50; `fast:a11` with score 0.48; `fast:a14` with score 0.46; `fast:a15` with score 0.38; `fast:a10` with score 0.18; `fast:a18` with score 0.16;

As an additional example, if we set $p = 2$ (`looseMatch` boolean modifier) and $\alpha = \infty$ (`onlyThread` relation modifier), we obtain the following ranking: `fast:a12` with score 0.46; `fast:a11` with score 0.41; `fast:a9` with score 0.18; `fast:a8`, `fast:a10`, `fast:a14`, and `fast:a15` with score 0.

As you can note the `onlyThread` relation modifier can produce some unexpected results since, for example, the annotation `fast:a14`, which is the only one that matches all the query clauses, get a similarity score of 0 being a leaf in the tree. On the other hand, the `halfThread` relation modifier produces a ranking where all the annotations are present but relevance is still moved from annotations that are leaves of the tree to inner annotations. This is still the case of `fast:a14` which is not the top ranked annotation while `fast:a12` gets ranked before it, being part of a thread. Moreover, lengthy paths are penalized: `fast:a11` is in the same thread of `fast:a14` and `fast:a12` but it does not benefit from the score obtained from `fast:a14` as much as `fast:a12`, due the lengthy path, and gets ranked after `fast:a12`.

The examples above show how flexible and powerful are the querying capabilities available for the user in order to search and retrieve annotations and how much they actually impact and modify the ranking and result list returned to the user, providing him with different views on the managed annotations.

5 Conclusions

We have discussed the issues related to the design and development of retrieval algorithms capable of taking into account annotations and their peculiar features. In particular, we have introduced a general framework which allows us to develop such retrieval algorithms based on a combination of extended boolean retrieval operators and hypertext driven information retrieval.

The proposed framework is quite general since it can express queries which range from pure boolean queries to unstructured queries as well as formulate queries which range from considering only the content of the annotations to considering only the structure of the hypertext between documents and annotations. Moreover, the proposed framework allows us to seamlessly plug different retrieval models into it.

Acknowledgements

The work reported has been partially supported by the TELplus Targeted Project for digital libraries, as part of the *eContentplus* Program of the European Commission (Contract ECP-2006-DILI-510003).

References

1. Agosti, M., Albrechtsen, H., Ferro, N., Frommholz, I., Hansen, P., Orio, N., Panizzi, E., Pejtersen, A.M., Thiel, U.: DiLAS: a Digital Library Annotation Service. In: Proc. Int. Workshop on Annotation for Collaboration – Methods, Tools, and Practices (IWAC 2005), pp. 91–101. CNRS - Programme société de l'information (2005)
2. Agosti, M., Berretti, S., Brettlecker, G., Del Bimbo, A., Ferro, N., Fuhr, N., Keim, D.A., Klas, C.-P., Lidy, T., Milano, D., Norrie, M.C., Ranaldi, P., Rauber, A., Schek, H.-J., Schreck, T., Schuldt, H., Signer, B., Springmann, M.: Delos-DLMS - the integrated DELOS digital library management system. In: Thanos, C., Borri, F., Candela, L. (eds.) Digital Libraries: Research and Development. LNCS, vol. 4877, pp. 36–45. Springer, Heidelberg (2007)
3. Agosti, M., Bonfiglio-Dosio, G., Ferro, N.: A Historical and Contemporary Study on Annotations to Derive Key Features for Systems Design. International Journal on Digital Libraries (IJDL) 8(1), 1–19 (2007)
4. Agosti, M., Ferro, N.: Annotations: Enriching a digital library. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 88–100. Springer, Heidelberg (2003)
5. Agosti, M., Ferro, N.: Annotations as context for searching documents. In: Crestani, F., Ruthven, I. (eds.) CoLIS 2005. LNCS, vol. 3507, pp. 155–170. Springer, Heidelberg (2005)
6. Agosti, M., Ferro, N.: Search strategies for finding annotations and annotated documents: The FAST service. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 270–281. Springer, Heidelberg (2006)
7. Agosti, M., Ferro, N.: A Formal Model of Annotations of Digital Content. ACM Transactions on Information Systems (TOIS) 26(1), 3:1–3:57 (2008)

8. Agosti, M., Ferro, N.: Annotations: a way to interoperability in DL. In: Christensen-Dalsgaard, B., Castelli, D., Ammitzbøll Jurik, B., Lippincott, J. (eds.) ECDL 2008. LNCS, vol. 5173, pp. 291–295. Springer, Heidelberg (2008)
9. Agosti, M., Ferro, N., Frommholz, I., Thiel, U.: Annotations in digital libraries and collaboratories – facets, models and usage. In: Heery, R., Lyon, L. (eds.) ECDL 2004. LNCS, vol. 3232, pp. 244–255. Springer, Heidelberg (2004)
10. Buneman, P., Khanna, S., Tan, W.-C.: On Propagation of Deletions and Annotations Through Views. In: Proc. 21st ACM Symposium on Principles of Database Systems (PODS 2002), pp. 150–158. ACM Press, New York (2002)
11. Cabanac, G., Chevalier, M., Chrisment, C., Julien, C.: Collective Annotation: Perspectives for Information Retrieval Improvement. In: Proc. 8th Conference on Information Retrieval and its Applications (RIAO 2007) (2007), `http://riao.free.fr/papers/99.pdf`
12. Candela, L., et al.: The DELOS Digital Library Reference Model. Foundations for Digital Libraries (2007), `http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf`
13. Fraenkel, A.S., Klein, S.T.: Information Retrieval from Annotated Texts. Journal of the American Society for Information Science (JASIS) 50(10), 845–854 (1999)
14. Frommholz, I., Brocks, H., Thiel, U., Neuhold, E.J., 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.) ECDL 2003. LNCS, vol. 2769, pp. 434–445. Springer, Heidelberg (2003)
15. Frommholz, I., Fuhr, N.: Probabilistic, Object-oriented Logics for Annotation-based Retrieval in Digital Libraries. In: Proc. 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2006), pp. 55–64. ACM Press, New York (2006)
16. Golder, S., Huberman, B.A.: Usage Patterns of Collaborative Tagging Systems. Journal of Information Science 32(2), 198–208 (2006)
17. Golovchinsky, G., Price, M.N., Schilit, B.N.: From Reading to Retrieval: Freeform Ink Annotations as Queries. In: Proc. 22nd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999), pp. 19–25. ACM Press, New York (1999)
18. Groth, D.P., Streefkerk, K.: Provenance and Annotation for Visual Exploration Systems. IEEE Transactions on Visualization And Computer Graphics 12(6), 1500–1510 (2006)
19. Halasz, F.G., Moran, T.P., Trigg, R.H.: Notecards in a Nutshell. In: Proc. Conference on Human Factors in Computing Systems and Graphics Interface (CHI 1987), pp. 45–52. ACM Press, New York (1987)
20. Handschuh, S., Staab, S. (eds.): Annotation for the Semantic Web. IOS Press, Amsterdam (2003)
21. Ioannidis, Y., et al.: Digital library information-technology infrastructures. International Journal on Digital Libraries 5(4), 266–274 (2005)
22. Kahan, J., Koivunen, M.-R.: Annotea: an open RDF infrastructure for shared Web annotations. In: Proc. 10th Int. Conference on World Wide Web (WWW 2001), pp. 623–632. ACM Press, New York (2001)
23. Marshall, C.C.: Toward an Ecology of Hypertext Annotation. In: Proc. 9th ACM Conference on Hypertext and Hypermedia (HT 1998): links, objects, time and space-structure in hypermedia systems, pp. 40–49. ACM Press, New York (1998)
24. Salton, G., Fox, E.A., Wu, H.: Extended Boolean Information Retrieval. Communications of the ACM (CACM) 26(11), 1022–1036 (1983)