Annotation-based Document Retrieval with Four-Valued Probabilistic Datalog

Ingo Frommholz
frommholz@ipsi.fhg.de

Ulrich Thiel
thiel@ipsi.fhg.de

Thomas Kamps
kamps@ipsi.fhg.de

Fraunhofer IPSI
Integrated Publication and Information Systems Institute
Dolivostr. 15
D-64293 Darmstadt, Germany

ABSTRACT

The COLLATE system (collaboratory for annotation, indexing and retrieval of digitized historical archive material) provides film researchers with a collaborative environment in which historic documents about European films can be analysed, interpreted and discussed, using nested annotations and discourse structure relations among them. Annotations are metadata, and annotation threads form a hypertext containing positive and negative links, constituting a certain kind of context exploitable for document retrieval. In this paper, we discuss a solution for using annotations for information retrieval. To exploit annotation threads which consist of nested annotations and typed links between them, an annotation-based retrieval approach should have to cope with negative and contradictory statements. The nested annotation retrieval approach (NARA) is an approach addressing these issues. Based on this, we present NARAlog, an implementation using four-valued probabilistic datalog (FVPD), able to perform an in-depth analysis of annotation threads and to deal with contradictory statements.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval Models*

General Terms

Algorithms, Design

Keywords

Annotations, four-valued probabilistic logics, datalog, context-based retrieval, formal model

WIRD'04, the first Workshop on the Integration of Information Retrieval and Databases (WIRD'04), Sheffield, United Kingdom

1. INTRODUCTION

Annotations are a means of supporting several tasks performed in Digital Libraries. They can assist the creation of new documents when new ideas and thoughts are discussed by means of annotations. Through annotations, users can interpret the document material at hand. Annotations support the effective use of documents by providing additional information which can make the content more intellectually accessible. Shared and public annotations can be used as building blocks for collaboration.

In this paper, we discuss how annotations can be exploited for information retrieval. Annotations made on documents can be seen as a special kind of *document context* containing additional information about the document. Using annotations for information retrieval means taking this context into account. From a syntactic point of view, annotations are a special kind of metadata [3]. They can appear in a simple form as direct comments on documents to more advanced forms like nested annotations with typed links connected to certain parts of documents. They can be textual, referential or graphical [2] and form a certain kind of hypertext (the so-called *annotation thread*) together with the document they are referring to. Annotations can contain content about content as well as additional content. They implicitly contain certain dialogue acts which might be made explicit by defining an appropriate annotation model. Nested annotations together with specific link types can be applied to model scientific discussions [6]. These discussions can contain additional content as well as possibly contradictory statements about the content of documents or annotations. Methods exploiting the annotation context for document retrieval should be able to perform an in-depth analysis of annotation threads in order to cope with the phenomena described above.

In the remainder of the paper, we will first briefly introduce the annotation model developed within the COLLATE system, which incorporates many features which can be found in recent annotation systems, such as nested annotations and typed links. We will then introduce our idea of a nested annotation retrieval approach (NARA) which, as the name implies, is capable of dealing with nested annotations as well as negative and contradictory statements. After that,

an example implementation of NARA based on four-valued probabilistic datalog (FVPD) [10] is introduced. Related work will be discussed and conclusions will be given.

2. THE COLLATE ANNOTATION MODEL

The COLLATE[1] system focuses on historic film documentation, dealing with documents about films of the 20s and 30s of the last century. Such documents can be, for example, censorship decisions, newspaper articles, etc. They are digitised and stored in the system repository. COLLATE supports the work between film scientists in different locations by establishing a collaboration cycle [9]: users can react to other user's contribution, and so the cycle continues. Users have the option of manually assigning keywords to the digitised documents as well as cataloguing them according to a pre-defined schema. One of the central concepts of COLLATE is to support document interpretation by enabling scientific discussion about documents through annotation threads.

Annotation threads consist of the annotated document (or a part of it) as root and nested annotations connected to the root. The links between the nodes of an annotation thread (documents and textual annotations) are typed with so-called *discourse structure relations*. In COLLATE, we defined the following relations: *elaboration* (giving additional information), *analogy* (describing similarities), *difference* (describing contrasts), *cause* (stating a cause for specific circumstances), *background information* (e.g., information about the background of an author), *interpretation* (of statements), *support argument* and *counterargument* (support or attack other arguments). Figure 1 shows an example of two discourse structure relations. The incorporation of these relations is discussed in more detail in [6]. Modeling annotation threads in this way gives us explicit information about the pragmatics of statements (through link types); this is important for the definition of suitable retrieval methods, as we will see later. An annotation thread in our case is a directed acyclic graph and forms a hypertext according to the definition in [1].

3. NESTED ANNOTATION RETRIEVAL APPROACH (NARA)

Since annotations contain valuable additional information about the document they are referring to, we will now briefly discuss how such information can be used for information retrieval. Due to their strong connection to documents, annotations can be seen as a kind of metadata. On the other hand, annotation threads build a rather complex hypertext [3], so advanced methods have to be applied.

Consider the example of an annotation thread as it is shown in Figure 1. The first annotation a_1 contains an interpretation of the content of d. A film scientist states that there might be other reasons for censoring the film d is talking about than described in d. a_1 extends the content of d, making it potentially interesting for users seeking for political censorship. But if we also take a_2 into account, we find a counterargument to a_1; another film scientist disagrees with the first one's opinion, so we have two contradictory statements. Ignoring a_2 or the link type between a_1 and a_2 would

[1] `http://www.collate.de/`

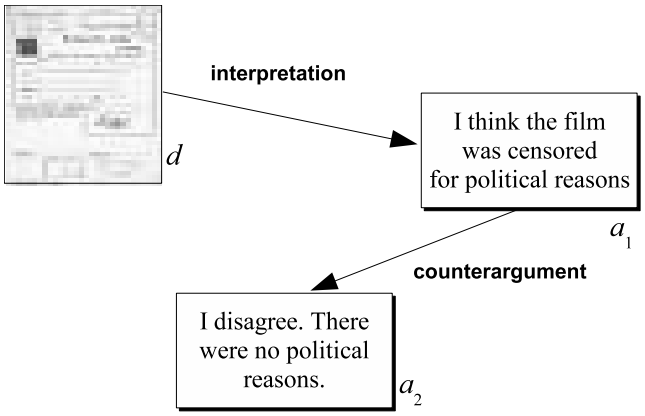

Figure 1: An example annotation thread

not be appropriate; whereas a_1 potentially raises the relevance of d when looking for political censorship, a_2 would lower it again. An in-depth analysis of the annotation thread has to be performed to cope with this situation accordingly, in essence with contradictory statements and nested annotations. This is addressed by our nested annotation retrieval approach (NARA).

Before discussing NARA, we provide a definition of an annotation thread.

Definition 1. Let A be the set of annotations and D be the set of documents. An *annotation thread* $\mathcal{A} = (N, E)$ is a directed acyclic graph with only one single root node $d \in D$. $N \subset D \cup A$ is the set of nodes in $\mathcal{A}$ representing documents and annotations, respectively; it is $n \in A$ if n is not the root node of $\mathcal{A}$. Let L be the set of link types; $E \subset N \times N \times L$ is the set of edges in $\mathcal{A}$ labeled with a link type $l \in L$.

In our framework, a query q consists of a set of query terms given by the user. We now outline our nested annotation retrieval approach which is a two-phase approach performing an in-depth analysis of annotation threads w.r.t. the query, using the content of annotations and the graph structure of the annotation threads.

1. *Initial content-based retrieval*: In this phase, the content-based retrieval status values (RSV) of each node in the annotation thread w.r.t. the query q are calculated. This is done using a retrieval function

$$r_{content} : N^D \times Q^D \longrightarrow \mathbb{R}$$

which maps a node description $n^D \in N^D$ of a document or annotation and a query description $q^D \in Q^D$ onto a real number. Depending on the underlying retrieval and indexing model, nodes and queries can be described as, e.g., vectors consisting of the weights of the terms contained in the document, annotation, or query, respectively. In the logic-based approach presented later, documents, annotations and queries are described using probabilistic facts.

Having calculated the RSVs for each node, we gain information to which degree these nodes are relevant w.r.t. the query q. This information is used for the analysis of the annotation thread, which is performed in the 2nd phase.

2. *Annotation-based re-weighting*: In this phase, the analysis of the annotation thread is performed in order to take the annotation context into account by biasing the initial retrieval status value $r_{content}(n^D, q^D)$ for each node d. The calculation of the context-based retrieval status value $r_{nara}(n, q)$ of a node n w.r.t. q can be described algorithmically as follows:

```
1: r_nara(n,q) = r_content(n^D, q^D)
2: if n is not a leaf in A then
3:    for all n' with (n,n',l) ∈ E do
4:       Calculate r_nara(n',q)
5:       r_nara(n,q) = f(r_nara(n,q), r_nara(n',q), l)
6:    end for
7: end if
```

The function f should stay undefined here. It biases the current weight $r_{nara}(n, q)$ using the context-based RSV of a successor node n' and the link type between n and n'. Implementations of NARA have to define f accordingly, especially in order to cope with inconsistent knowledge.

4. LOGIC-BASED IMPLEMENTATION OF NARA

In this section we will introduce NARAlog, which is an implementation of NARA using four-valued probabilistic datalog.

4.1 FVPD

Four-valued probabilistic datalog is an extension of probabilistic datalog. Similar to Prolog, its syntax consists of variables, constants, predicates and Horn clauses. Probabilities can be assigned to facts. Semantically, FVPD uses four different truth values (besides the classical ones, *true* and *false*) [10, 22]. These additional values are *unknown* and *inconsistent*. Basically, FVPD deals with an open world assumption (OWA). In contrast to a closed world assumption (CWA), the absence of an atom $p(a)$ does not imply $\neg p(a)$, but would assign the truth value *unknown* to $p(a)$ instead of *false*. In general, using four-valued logics and OWA, a model may contain a positive atom, a negative atom, neither of them or both. As an example, let the model $M = \{p(a), \neg p(a), p(b), \neg p(c)\}$ and the Herbrand base be $\{p(a), p(b), p(c), p(d)\}$. Then the following truth values would be assigned to the elements of the Herbrand base:

$p(a)$	*inconsistent*
$p(b)$	*true*
$p(c)$	*false*
$p(d)$	*unknown*

FVPD was originally developed for the retrieval of hypermedia documents and it integrates concepts coming from information retrieval (like uncertain inference) and deductive databases. In hypermedia documents, a similar situation might occur as we might have it with annotations: different nodes of a hypermedia document can contain contradictory statements. This can be handled with four-valued probabilistic logics, as the simple example in Figure 2 shows. Here, we see a hypermedia document d composed of two

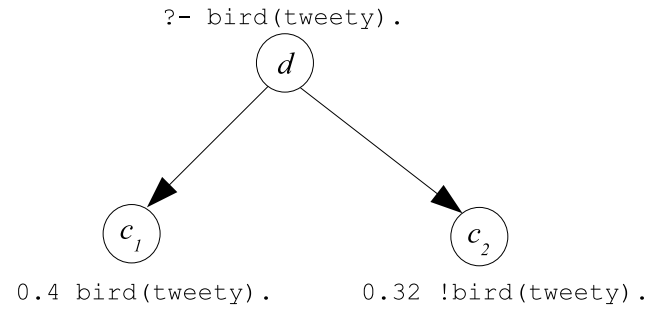

Figure 2: Hypermedia document with contradictory statements

chapters, c_1 and c_2. Each chapter constitutes its own context. Now consider the query `?- bird(tweety)`[2]. Both chapters c_1 and c_2 contain facts relevant to the query: c_1 provides evidence that `bird(tweety)` is true with the probability 0.4, and c_2 provides evidence that `bird(tweety)` is false with the probability 0.32. So we find contradictory information in both subparts of d (which is not contradictory in the subparts c_1 and c_2 themselves). To calculate the probability that `bird(tweety)` is true in the context of d, the positive evidence coming from c_1 (0.4) is combined with the non-negative evidence from c_2 $(1 - 0.32)$, resulting in $0.4 \cdot (1 - 0.32) = 0.272$. This process is called *knowledge augmentation* and it is discussed in more detail in [10, 22]. With HySpirit[3] there exists an implementation of FVPD [10].

4.2 NARAlog

The process of knowledge augmentation dealing with contradictory statements coming from subparts of hypermedia documents is similar to the process of dealing with such knowledge in an annotation thread. In hypermedia documents, we consider subparts (like c_1 and c_2 in the example described above) in order to calculate the final probability for a fact to be true in a higher context (like d). With NARA as described in Section 3, we consider direct annotations in the re-weighting phase. This makes FVPD an interesting framework for a logic-based implementation of NARA, which we call NARAlog.

NARAlog is based on the view of information retrieval as uncertain inference, as proposed in [23]. The retrieval function thus estimates the probability $P(d \rightarrow q)$ that a document d implies a query q. We state that an open world assumption is more suitable for our approach than a closed world assumption since we have to deal with positive and negative evidence in annotations which should be handled independently.

[2]Queries are formulated as goal clauses in FVPD
[3]`http://qmir.dcs.qmul.ac.uk/hyspirit.html`

NARAlog underlies some assumptions on the annotation collection. We assume that we have an annotation thread similar to the one used in COLLATE, and that the link types can be categorised in positive and negative ones w.r.t. their effect on the retrieval weight of the link source (see the discussion below). Link types are explicitly given or derived automatically with some uncertainty (methods determining link types are not an issue here). Annotations are atomic in that their pragmatics are consistent with the according link type, e.g. an annotation being a counterargument would not contain an additional interpretation. We assume that in this case users would create two corresponding annotations.

4.2.1 Components

Several components are needed in NARAlog, which are content-based retrieval weights of both documents and annotations, document and link types, positive and negative links and access probabilities. These components will be discussed now.

4.2.1.1 Content-based Indexing and Retrieval

In NARA, the results of the first phase are content-based retrieval weights determined by a retrieval function $r_{content}$, which is based on document and query descriptions usually derived by an indexing process. NARAlog indexes documents and annotations as probabilistic facts, using the predicates `term` and `termspace`. `term` is a binary predicate describing term weights w.r.t. documents. As an example, the probabilistic facts

```
0.2 term(d1, "political").
0.3 term(d1, "censorship").
```

would mean that the document `d1` contains the terms "political" and "censorship" and their weights are 0.2 and 0.3, respectively. These weights might be, e.g., normalised term frequencies. The facts

```
0.3 termspace("political").
0.5 termspace("censorship").
```

provide another weight to a term which is independent of the documents but unique for the whole termspace. This value might be, e.g., the inverse document frequency.

Query terms are described as probabilistic facts in a similar way. Suppose the query is "political reasons", we would create the following facts:

```
qterm("political").
qterm("reasons").
```

(if no probabilistic weight is given, 1 is assumed).

Based on these facts, a content-based retrieval function implementing the initial NARA phase can be described as the following rule:

```
r_content(N) :- qterm(T) & termspace(T) &
                term(N,T).
```

Because of the query term "political", this rule would yield $1 \cdot 0.3 \cdot 0.2 = 0.06$ for `d1`.

4.2.1.2 Document and Link Types

Nodes in an annotation thread have to be classified whether they are documents or annotations. This can be done using the facts

```
document(d1).
annotation(a1).
annotation(a2).
```

which mean that `d1` is a document and `a1` and `a2` are annotations.

Furthermore, we have to model links of a certain type. This can be done, e.g., by creating appropriate binary predicates representing links, as can be seen in the following example.

```
interpretation(d1,a1).
counterargument(a1,a2).
```

These facts represent the links as they are found in Figure 1. In a system like COLLATE where the relation types are explicitly given, we can assign the probability 1 to each link predicate if the corresponding link appears between two nodes. However, other systems might not offer explicit link types, but have to automatically derive them from, e.g., the content of the link's source or destination node. This means such link types are estimated with a degree of uncertainty which can manifest itself in probabilistic weights less than 1.

4.2.1.3 Positive and Negative Links

Link types should be categorised w.r.t. the effect their destination nodes have on the calculation of the retrieval weight of the source node. Consider the example in Figure 1, annotation a_1, being an interpretation, would raise the context-based retrieval weight of d, but never lower it. On the other hand, the effect of the counterargument relation between a_1 and a_2 would mean that the context-based weight of a_1 is lowered according to the content-based weight of a_2 w.r.t. the query, which in turn would decrease the overall retrieval weight of document d. It should be noted that if we had a counterargument a_3 to a_2, this would lower the context-based weight of a_2 and in turn raise the weight of a_1.

The categorisation of link types into positive and negative ones can be done by creating appropriate rules like

```
pos_link(X,Y) :- interpretation(X,Y).
pos_link(X,Y) :- elaboration(X,Y).
neg_link(X,Y) :- counterargument(X,Y).
```

4.2.1.4 Access Probability

In order to make NARAlog customisable w.r.t. user preferences, the model should take into account the probability that an annotation is actually accessed and considered. The

access probability can depend on attributes such as the author of an annotation; users might prefer reading one author's annotations while neglecting another author's. If no such information is given, a fixed value for the access probabilities can be assumed. Access probabilities can be modeled using the `acc` predicate:

```
0.8 acc(d1,a1).
0.8 acc(a1,a2).
```

4.2.2 Context-based Retrieval Function

Having introduced all the required components of NARAlog, we will now discuss the context-based retrieval function r_{nara}. This function consists of certain rules and implements the re-weighting phase of NARA. As mentioned above, positive and negative evidence contained in annotations and links should be considered independently, which motivates an open world assumption. A logic-based approach should therefore collect positive and negative evidence and combine it accordingly, which is done with knowledge augmentation in FVPD.

Positive evidence is contained in a node n itself, so the context-based retrieval weight n in the annotation thread is first of all determined by its content-based weight, which can be expressed like this:

```
r_nara(N) :- r_content(N).
```

Positive evidence is contained in successor nodes n' of n if there exists a positive link between n and n'. In this case, the positive value of n should be increased to the degree of the context-based weight of n' and the probability that n' is actually visited:

```
r_nara(N) :- pos_link(N,N') & acc(N,N') &
             r_nara(N').
```

Negative evidence is contained in successor nodes n' of n if there exists a negative link between n and n'. The negative value of n should be increased accordingly:

```
!r_nara(N) :- neg_link(N,N') & acc(N,N') &
              r_nara(N').
```

4.2.3 Example NARAlog Program

Figure 3 shows an example of a NARAlog program. Lines 1-6 show the results of the indexing process of documents and annotations; as discussed above, both are described as probabilistic facts using the `term` and `termspace` relations. HySpirit offers the means to load such facts directly from relational databases. The nodes of the annotation thread are categorised into documents and annotations in the lines 8-10, links are modeled in lines 12 and 13. Line 15 and 16 contain access probabilities, in this example 0.8. The facts between lines 1 and 16 are static and could all be stored in the index database.

Lines 18 to 20 contain rules which categorise the link types w.r.t. their effect on the context based retrieval weight into positive and negative links. The query is reflected in lines 22 and 23. The initial phase is implemented as a rule considering the indexed content of documents and annotations (line 26). The re-weighting phase is reflected in lines 29-31, where values for positive and negative evidence are collected. `r_nara(N)` implements the re-weighting phase since it is recursively executed for every successor node. The ranking process itself is started in line 33. By inserting `document(D)` we exclude annotations from being retrieved.

4.2.4 Calculation

When traversing the annotation thread in the re-weighting phase, NARAlog collects positive and negative evidence w.r.t. the query. But since the rules in lines 30 and 31 only contain positive facts, we have to show how NARAlog (or more precisely: the underlying HySpirit system) deals with negative facts. We will show this by discussing an example run. For this, reconsider the example annotation thread in Figure 1. The probability $P(a_2 \rightarrow q)$ that a_2 implies the query q is given by its content-based value calculated in line 26 (since there are no more successors of this annotation in our example) and is 0.2456. The calculation of $P(a_1 \rightarrow q)$ is performed by including the evidence found in a_2; since there is a negative link between a_1 and a_2, we have negative evidence here. The corresponding probability is computed by combining the link type, the access probability and $P(a_2 \rightarrow q)$, so that we gain $1 \cdot 0.8 \cdot 0.2456 = 0.19648$ for $P(!\texttt{r\_nara(a1)})$. The only positive evidence for `r_nara(a1)` comes from the content-based retrieval weight `r_content(a1)` which is 0.2345. Both positive and negative evidence are combined resulting in $P(a_1 \rightarrow q) = P(\texttt{r\_nara(a1)}) \cdot (1 - P(!\texttt{r\_nara(a1)})) = 0.235 \cdot 0.80352 = 0.1888272$. The value for $P(d_1 \rightarrow q)$, which is what we are looking for, is 0.185019.

5. RELATED WORK

5.1 Annotations and Annotation Systems

Marshall *et al.* [16, 17, 18] provide results of empirical investigations of annotations. These considerations provide useful insights into how to incorporate annotations in a digital library. Several dimensions of annotations are identified, like formal vs. informal, explicit vs. tacit, personal vs. global annotations, etc [17]. Phelps and Wilensky discuss several properties of digital annotations which are realised in their multivalent annotation framework [21]. Annotations are seen as the basis for collaborative work.

Ovsiannikov *et al.* provide an overview of annotation systems and identify four main aspect of annotation usage: to remember, think, clarify and share [20]. Concepts of annotation technologies are derived based on these considerations. Agosti and Ferro create a conceptual model of annotations based on two dimensions: the meaning of annotations (e.g., comprehension, interpretation, cooperation and revision) and the sign of an annotation (textual, graphical or referential). They present an architecture of an annotation service in OpenDLib [2]. [3] contains an examination of annotations from a syntactic, semantic and pragmatic view. Some options on annotation-based information retrieval are discussed there. Furuta *et al.* present Walden's Path, a system to create new hypertext paths with annotations [11].

There are several other annotation systems and prototypes.

```
0.2 term(d1, "political").       0.3 term(d1, "censorship").
0.5 term(a1, "political").       0.5 term(a1, "reasons").
0.4 term(a2, "political").       0.6 term(a2, "reasons").

0.2 termspace("political").      0.5 termspace("censorship").
0.3 termspace("reason").

document(d1).
annotation(a1).
annotation(a2).

interpretation(d1,a1).
counterargument(a1,a2).

0.8 acc(d1,a1).
0.8 acc(a1,a2).

pos_link(X,Y) :- interpretation(X,Y).
pos_link(X,Y) :- elaboration(X,Y).
neg_link(X,Y) :- counterargument(X,Y).

qterm("political").
qterm("reasons").

# Initial content-based phase
r_content(N) :- qterm(T) & termspace(T) & term(N,T).

# Annotation-based re-weighting phase
r_nara(N) :- r_content(N).
r_nara(N) :- pos_link(N,N') & acc(N,N') & r_nara(N').
!r_nara(N) :- neg_link(N,N') & acc(N,N') & r_nara(N').

?- document(D) & r_nara(D).
```

Figure 3: Example NARAlog program

Yawas [7] is a Web annotation tool with which the user can annotate the document content or provide information, e.g., about the document type. Annotea [13] is another Web annotation tool which can deal with nested annotations and several annotation types (realised as typed links between annotations). DEBORA [19] is a digital library for Renaissance books. Annotations are used to share information and to create virtual books. In our COLLATE prototype, nested annotations are used to enable scientific discussions [9]. Some commercial systems like Word and Acrobat provide means for document annotation.

Although most of the work presented in this subsection did not focus on information retrieval, they give a valuable insight into the nature of annotations and how to model them, which in turn can be used to develop appropriate retrieval functions.

5.2 Hypertext Information Retrieval and Categorisation

Since annotations connected to their annotated resources can be seen as a specific hypertext [3], it is worth investigating common hypertext information retrieval (HIR) approaches w.r.t. their applicability for annotation-based information retrieval. [4] gives a good overview of some research in HIR. An interesting numeric approach is presented by Frei and Stieger [8] using spreading activation. Similar to NARA, there is an initialisation phase (where content-based RSVs are calculated), and a navigation phase where the hypertext structure is traversed to bias the initial RSV. The approach can cope with link types, but the possible incorporation of negative links does not suffice since, as described in Section 4.2.1.3, two consecutive negative links might again have a positive effect on the contextual RSV, which is not covered by the algorithms presented in [8].

As mentioned earlier in this paper, Fuhr and Rölleke use four-valued logics in order to retrieve complex objects [22, 10]. Their work is the foundation of the NARAlog approach presented here.

The idea of exploiting what others said about a document is formulated as well in [5]. In their approach of categorisation by context Attardi *et al.* use anchor texts and the surroundings of a hyperlink as additional information about the document. This information can be seen as annotations of a document since it often contains summaries, interpretations or reformulations of document content. Being tailored

to the World Wide Web, the approach presented in [5] does not use any link types and does not consider any positive or negative links.

5.3 Annotation-based Information Retrieval

It is noticable that there has not been much work presented so far in information retrieval using textual annotations. Some annotation systems provide simple full-text search mechanisms on annotations [20], but do not support annotation-based document search. After finding appropriate annotations, a user still has to browse to the according document. Results from document and annotation search are not combined. Yawas [7] offers some means to use annotations for document search, e.g. by enabling users to search for a specific document type considering annotations. The approach does not consider nested nor negative annotations.

An interesting approach using annotation-generated queries with relevance feedback is introduced by Golovchinsky *et al.* [12]. Here, annotations are markings given by users who judge certain parts of a document as being important when emphasising them. Evaluations show that using these kinds of annotations provides better retrieval effectiveness than classic relevance feedback. Being based on relevance judgements and annotations as markings rather than content, this approach is totally different from what we propose here.

6. FUTURE WORK

Experiments have yet to conclusively prove the impact and significance of using annotations as we proposed in this paper on retrieval effectiveness. We will perform experiments using the COLLATE collection, which at the moment consists of 6980 documents and 1994 annotations. Possible baselines for this would be neglecting annotations and the structure of an annotation thread. Besides effectiveness experiments have also to address the efficiency of our approach; possible solutions of making our approach more efficient would be to prune the annotation thread at a certain depth. Another important issue is to find other appropriate test collections containing more documents and annotations as in the COLLATE repository. Following the idea proposed in [5], annotations extracted from hypertext test corpora might gain a suitable collection for evalutation purposes.

Further research is needed to determine suitable weights for parameters like the access probability $P(\texttt{acc})$ described in Section 4.2.1.4. We think this probability is an important parameter to cover user's interests, but some suitable values have to be found in case this information is not available. One simple approach would be to assign a global value for all access probabilities.

In COLLATE we let users explicitly state which kind of annotation they create, so we know about the link type. It is required that some kind of atomicity is given. As an example, an annotation like "there were no political but economical reasons" would not be valid in our framework, since besides a counterargument ("there were no political reasons") it also contains an interpretation ("there were economical reasons"). The annotation author would have to create two annotations to formulate her point. Our model, being suitable for an experimental system like COLLATE, is not convenient w.r.t. usability. So it is desirable to automatically recognise link types like discourse relations between annotations or even parts of annotations, at least w.r.t. their effect (negative or positive). The work by Marcu and Echihabi [15] where they use Naive Bayes classifiers for determining discourse relation seems to be an interesting starting point for calculating probabilities for link types like the ones discussed in Section 4.2.1.2.

7. CONCLUSION

In this paper we have discussed NARA, which is an approach to perform information retrieval using nested annotation. Such nested annotation together with typed links between them build an annotation thread. NARA biases content-based retrieval status values by analysing the annotation thread. For this, NARA has to deal with negative and contradictory statements.

With NARAlog we have presented an implementation of NARA based on four-valued probabilistic datalog. Using four-valued logics and an open world assumption, NARAlog is capable of coping with nested annotation as well as negative and contradictory ones. The components of NARAlog, including content- and context-based retrieval functions, were discussed.

8. REFERENCES

[1] M. Agosti. An Overview of Hypertext. In Agosti and Smeaton [4], pages 27–47.

[2] M. Agosti and N. Ferro. Annotations: Enriching a Digital Library. In Koch and Sølvberg [14], pages 88–100.

[3] M. Agosti, N. Ferro, I. Frommholz, and U. Thiel. Annotations in digital libraries and collaboratories – facets, models and usage. In R. Heery and L. Lyon, editors, *Proc. 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, 2004. To appear.

[4] M. Agosti and A. Smeaton, editors. *Information Retrieval and Hypertext*. Kluwer Academic Publishers, Norwell (MA), USA, 1996.

[5] G. Attardi, A. Gullí, and F. Sebastiani. Automatic Web page categorization by link and context analysis. In C. Hutchison and G. Lanzarone, editors, *Proceedings of THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence*, pages 105–119, Varese, IT, 1999.

[6] H. Brocks, A. Stein, U. Thiel, I. Frommholz, and A. Dirsch-Weigand. How to incorporate collaborative discourse in cultural digital libraries. In *Proceedings of the ECAI 2002 Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM02)*, Lyon, France, July 2002.

[7] L. Denoue and L. Vignollet. An annotation tool for web browsers and its applications to information retrieval. In *Proceedings of RIAO 2000, Paris, April 2000*, April 2000.

[8] H. P. Frei and D. Stieger. The use of semantic links in hypertext information retrieval. *Information*

Processing and Management: an International Journal, 31(1):1–13, 1995.

[9] I. Frommholz, H. Brocks, U. Thiel, E. Neuhold, L. Iannone, G. Semeraro, M. Berardi, and M. Ceci. Document-centered collaboration for scholars in the humanities - the COLLATE system. In Koch and Sølvberg [14], pages 434–445.

[10] N. Fuhr and T. Rölleke. HySpirit – a probabilistic inference engine for hypermedia retrieval in large databases. In H.-J. Schek, F. Saltor, I. Ramos, and G. Alonso, editors, *Proceedings of the 6th International Conference on Extending Database Technology (EDBT), Valencia, Spain*, Lecture Notes in Computer Science, pages 24–38, Heidelberg et al., 1998. Springer.

[11] R. Furuta, F. M. Shipman, C. C. Marshall, D. Brenner, and H. Hsieh. Hypertext paths and the world-wide web: Experiences with Walden's Path. In *Hypertext '97: the Eighth ACM Conference on Hypertext*, pages 167–176. ACM Inc., UK, 1997.

[12] G. Golovchinsky, M. N. Price, and B. N. Schilit. From reading to retrieval: Freeform ink annotations as queries. In F. Gey, M. Hearst, and R. Tong, editors, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–25, New York, 1999. ACM Press.

[13] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An open rdf infrastructure for shared web annotations. In *Proceedings of the WWW10 International Conference*, Hong Kong, May 2001.

[14] T. Koch and I. T. Sølvberg, editors. *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.

[15] D. Marcu and A. Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 368–375, July 2002.

[16] C. C. Marshall. Annotation: from Paper Books to the Digital Library. In R. B. Allen and E. Rasmussen, editors, *Proc. 2nd ACM International Conference on Digital Libraries (DL 1997)*, pages 233–240. ACM Press, New York, USA, 1997.

[17] C. C. Marshall. Toward an Ecology of Hypertext Annotation. In R. Akscyn, editor, *Proc. 9th ACM Conference on Hypertext and Hypermedia (HT 1998): links, objects, time and space-structure in hypermedia systems*, pages 40–49. ACM Press, New York, USA, 1998.

[18] C. C. Marshall and A. J. B. Brush. From Personal to Shared Annotations. In L. Terveen and D. Wixon, editors, *Proc. Conference on Human Factors and Computing Systems (CHI 2002) – Extended Abstracts on Human Factors in Computer Systems*, pages 812–813. ACM Press, New York, USA, 2002.

[19] D. Nichols, D. Pemberton, S. Dalhoumi, O. Larouk, C. Belisle, and T. M.B. DEBORA: Developing an Interface to Support Collaboration in a Digital Library. In J. Borbinha and T. Baker, editors, *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000)*, Lecture Notes in Computer Science, pages 239–248, Berlin et al., 2000. Springer.

[20] I. A. Ovsiannikov, M. A. Arbib, and T. H. McNeill. Annotation technology. *Int. J. Hum.-Comput. Stud.*, 50(4):329–362, 1999.

[21] T. A. Phelps and R. Wilensky. Multivalent Annotations. In C. Peters and C. Thanos, editors, *Proc. 1st European Conference on Research and Advanced Technology for Digital Libraries (ECDL 1997)*, pages 287–303. Lecture Notes in Computer Science (LNCS) 1324, Springer, Heidelberg, Germany, 1997.

[22] T. Rölleke and N. Fuhr. Retrieval of complex objects using a four-valued logic. In *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, New York, 1996. ACM.

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