Optimizing the Retrieval of Pertinent Answers for NL Questions with the E-Librarian Service

[1]Serge Linckels, [2,1]Harald Sack, [1]Christoph Meinel

[1]Hasso Plattner Institute for Software Systems Engineering (HPI)
University of Potsdam, Postfach 900460, D-14440 Potsdam, Germany
{linckels,meinel}@hpi.uni-potsdam.de
[2]Institut für Informatik, Friedrich-Schiller-Universität Jena
Fürstengraben 1, D-07743 Jena, Germany
sack@minet.uni-jena.de

Abstract. There is a growing discrepancy between the creation of digital content and its actual employment and usefulness in a learning society. Technologies for recording lectures have become readily available and the sheer number and size of such objects produced grows exponentially. However, in practice most recordings are monolithic entities that cannot be integrated into an active learning process offhand. To overcome this problem, recorded lectures have to be semantically annotated to become full-fledged e-learning objects facilitating automated reasoning over their content. We present a running web-based system — the e-Librarian Service CHESt — that is able to match a user's question given in natural language to a selection of semantically pertinent learning objects based on an adapted best cover algorithm. We show with empirical data that the precision of our e-Librarian Service is much more efficient than traditional keyword-based information retrieval; it yields a correct answer in most of the cases (93% of the queries), and mostly with a high precision, i.e., without supplementary hits. We also describe some ideas to improve the retrieval performance by user feedback.

1 Introduction

The World Wide Web (WWW) is the largest knowledge base that ever existed. The availability of material with educational content in the WWW increases dramatically. However, its usage in an educational environment is poor, mainly due to two facts [16, 6, 18, 22]. First, there is currently no reliable mechanism to prove the correctness of the data. Second, there is way too much information, in particular redundant and not relevant information, so that finding appropriate answers in an efficient way is a rather difficult task being reliant on the user's interaction. The user is charged with the awkward, time consuming and diverting task of filtering the pertinent information out of the noise. Turning large knowledge bases as the WWW into useful educational resources requires to identify correct, reliable, and machine understandable resources, as well as to develop simple but highly efficient search tools with the ability to perform logical inferences over these resources. This idea is fully in the stream of current Semantic Web thinking.

In this paper we describe a running system[1] — the e-Librarian Service CHESt [12, 13] — that is able to understand a user's questions given in natural language (NL) and to retrieve semantically pertinent resources. We call such resources *Learning Objects* (LOs). By LO we refer to an entity about a precise subject that may be used for learning, education or training [20], e.g., a multimedia sequence including machine processable metadata that semantically describe its content. Our E-Librarian Service can be perceived as a specialization of passage retrieval techniques; see [14] for an overview.

It has been realized that digital libraries do benefit from having its content understandable and available in a machine processable form, and it is widely agreed that ontologies will play a key role in providing the infrastructure to achieve this goal. One of the basic building blocks of our e-Librarian Service is a common domain ontology, which has a double use. First, it is used for the translation of the NL user questions into a formal language, i.e., Description Logics (DLs). DLs are a family of knowledge representation formalisms that allow to represent the knowledge of an application domain in a structured way and to reason about this knowledge [1]. The semantic interpretation, i.e., the translation of a NL user question into a DL is described in [12]. Second, the domain ontology is used to describe the LOs in the knowledge base with additional semantic metadata. We developed a tool that helps to semi-automatically generate the semantic metadata based on the textual content of the LOs.

Our e-Librarian Service implements a retrieval algorithm that is based on the concept covering problem. Among all the LOs that have some common information with the user query, our algorithm is able to identify the most pertinent match(es), keeping in mind that the user in general expects an exhaustive answer while preferring a concise answer with only little or no information overhead. The evaluation of our algorithm shows that in an educational environment our e-Librarian Service is much more appropriate than a traditional keyword-based search engine, because it delivers much less information overhead while simultaneously providing a higher precision.

The paper is structured as follows. After this introduction, section 2 discusses related work and projects. The main contribution of the paper is the algorithm for retrieving semantically pertinent LOs from a given knowledge base. The algorithm is presented in section 3, and explicitly discussed and evaluated in section 4. Section 5 provides an outlook and discussion how the system can be improved by user contributions and feedback, while section 6 concludes the paper with a brief summary of achieved results.

2 Related Work

Instead of the traditional Question Answering (QA) as being subject in linguistics and information retrieval [17], our approach is not targeted to compute a

[1] `http://www.linckels.lu/chest`

coherent answer being expressed in NL. We simply provide a set of interrelated resources (LOs), which contain the information that is necessary to answer the user's question. The user has to read the provided LO(s) to obtain an answer.

We address three different approaches related to document matching and retrieval based on DL inferences. First, an approach for matching documents based on non-standard inferences in the DL sub-languages $\mathcal{ALNS}, \mathcal{ALN}^*$, and $\mathcal{ALE}$ is presented in [9]. A matching problem modulo equivalence and modulo subsumption is of the form $C \equiv^? D$ and $C \sqsubseteq^? D$ respectively, where C is a description and D a pattern. A solution or matcher of these problems is a substitution σ such that $C \equiv \sigma(D)$ and $C \sqsubseteq \sigma(D)$, respectively. The solution is based on computing homomorphisms between description trees. Although this is an excellent solution for dealing with complex descriptions such as for comparing complete documents, it is less appropriate for our purpose. In our case, LOs are described by simple semantic annotations with few role-imbrications. The resulting description trees are rather flat and comprise rarely more than two levels.

Second, the concept covering problem [7] is based on DLs with structural subsumption. The proposed algorithm for identifying the best cover relies on the computation of minimal transversals in a hypergraph. The algorithm has been implemented in the project MKBEEM (Multilingual Knowledge Based European Electronic Marketplace). That solution is very pertinent for our e-Librarian Service because it always finds the best cover, i.e., the best matching LOs w.r.t. the user's question (see section 3.2).

Another definition of the concept covering problem that eliminates the limitation of DLs to provide structural subsumption has been presented in [5]. There, the concept covering problem is based on the concept abduction problem (CAP) [19], which is able to provide an explanation if subsumption does not hold. It is stated as follows: S (supply) and D (demand) are two descriptions in a DL $\mathcal{L}$, and satisfiable in a terminology $\mathcal{T}$. A CAP, identified by $< \mathcal{L}, S, D, \mathcal{T} >$, is finding a concept $H \in \mathcal{L}$ (hypotheses) such that $\mathcal{T} \models S \sqcap H \sqsubseteq D$, and moreover $S \sqcap H \not\equiv \bot$. The algorithm was implemented in a project for semantic-based discovery of matches and negotiation spaces in an e-marketplace. One of the weaknesses of this solution is that does not always return an optimal cover.

We decided to base our e-Librarian Service on the concept covering problem as presented in [7] because for our application DLs with structural subsumption provide sufficient expressiveness. Furthermore, our system must always return an optimal cover. Finally, the solution is simple and adapted to our LO descriptions.

3 The LO Retrieval Problem

In this section we describe the retrieval aspect of our e-Librarian Service that can be perceived as a specialization of *passage retrieval* techniques. Passage retrieval techniques have been extensively used in standard IR settings, and have proven effective for document retrieval when documents are long or when there are topic changes within a document, thus making it an appealing candidate for the present work [14]. By *retrieval* we refer to answering a user's question by identifying only the semantically most pertinent LOs according to the given question. In addition, the system must be able to quantify the quality of the yielded results, i.e., to measure the semantic distance between the user's query and the identified LOs. This measure is also used to rank similar results.

Our solution is based on the *concept covering problem* and on the quantification of the *semantic difference*. The novelty of our approach is that it always proposes a solution to the user, even if the system concludes that there is no exhaustive answer. By quantifying the missing and supplementary information, the system is able to compute and visualize the quality and pertinence of the yielded LO(s).

3.1 Least Common Subsumer and Semantic Difference

The least common subsumer (lcs) [2] stands for the least concept description (w.r.t. subsumption) that subsumes a given set of concept descriptions.

Definition 1 (Least Common Subsumer). *Let $\mathcal{L}$ be a DL and C, D, E be $\mathcal{L}$-concept descriptions. The concept E is a lcs of C, D iff it satisfies:*

- $C \sqsubseteq E$ *and* $D \sqsubseteq E$, *and*
- E *is the least $\mathcal{L}$-concept description with this property, i.e., if E' is an $\mathcal{L}$ concept description satisfying $C \sqsubseteq E'$ and $D \sqsubseteq E'$, then $E \sqsubseteq E'$.*

Definition 2 (Semantic Difference). *[21] Let $\mathcal{L}$ be a DL and $C, D \in \mathcal{L}$ two concept descriptions with $C \sqsubseteq D$. Then the semantic difference $C - D$ is defined by:*

$$C - D = max_{\sqsubseteq}\{E \in \mathcal{L} : E \sqcap D \equiv C\}.$$

This definition of semantic difference requires that the second argument subsumes the first one. However, the semantic difference $C - D$ between two incomparable descriptions C and D can be given by computing the least common subsumer of C and D:

$$C - D = C - lcs(C, D).$$

3.2 Finding Pertinent Documents

Although the principle of the concept covering problem (see section 2) is the most pertinent solution for our E-Librarian Service, we think that a user might not be satisfied if the delivered answer to his/her precise question is a concatenation of different — normally not related — resources from the knowledge base. First, there is no transition between the different LOs in the answer. Second, we risk that there is mean to much information because the original concept covering problems adds all LOs to the answer until the answer is covered completely.

We learned from experiments [11] that users prefer few but precise answers even if these answers are not complete, rather than a set of different concatenated documents. This assertion is confirmed by pedagogical analyzes, e.g., [10, 6, 8, 4] that students are searching for one — the best — answer, and do not consider different delivered search results. They would rather reformulated their query until they receive only a few results, or until they find the perfect result.

Our modified concept covering problem defines a cover as a concept description C w.r.t. a terminology $\mathcal{T}$ that shares some information with another concept description Q w.r.t. $\mathcal{T}$.

Definition 3 (Cover). *Let $\mathcal{L}$ be a DL with structural subsumption, $\mathcal{T}$ be an $\mathcal{L}$-terminology and $C_\mathcal{T} = \{C_i \not\equiv \bot, i \in [1,n]\}$ the set of concept descriptions occurring in $\mathcal{T}$. Then $C_j \in C_\mathcal{T}$ is a cover of a $\mathcal{L}$-concept description $Q \not\equiv \bot$ if $Q - lcs_\mathcal{T}(Q, C_j) \not\equiv Q$.*

The best cover can be defined based on the remaining information in the query (denoted as $Miss$) and in the cover (denoted as *Rest*). The Miss is the part of the query that is not part of the cover, and the Rest is the information that is part of the cover but not required by the query.

Definition 4 (Miss and Rest). *Let Q, C be be two $\mathcal{L}$-concept descriptions.*

- *The Miss of Q w.r.t. C, denoted as $Miss(Q, C)$ is defined as follows:* $Miss(Q, C) = Q - lcs_\mathcal{T}(Q, C)$.
- *The Rest of Q w.r.t. C denoted as $Rest(Q, C)$ is defined as follows:* $Rest(Q, C) = C - lcs_\mathcal{T}(Q, C)$.

The best cover can be assumed as being the cover with the smallest Miss and Rest. Therefore, we have to quantify the Miss and the Rest, i.e., measure the size of a $\mathcal{L}$-concept description.

Definition 5 (Size of a Concept Description). *The size of a $\mathcal{L}$-concept description, denoted as $|\cdot|$ is inductively defined by:*

- $|\bot| = |\top| = 0$,
- $|A| = |\neg A| = 1$,
- $|\exists r.C| = |\forall r.C| = 2 + |C|$,
- $|C \sqcap D| = |C \sqcup D| = |C| + |D|$,
- $|\neg C| = |C|$.

Definition 6 (Best Cover). *Let C, D be two $\mathcal{L}$-concept descriptions. A cover C is called a best cover w.r.t. Q using a terminology $\mathcal{T}$ iff:*

- *C is a cover w.r.t. Q using $\mathcal{T}$, and*
- *there does not exists any cover C' of Q using $\mathcal{T}$ such that*

$$(|Miss(Q, C')|, |Rest(Q, C')|) < (|Miss(Q, C)|, |Rest(Q, C)|)$$

where $<$ stands for the lexicographic order.

By choosing a lexicographical order we give preference to a minimized Miss, e.g., for (Miss,Rest), the couple (1,2) $<$ (2,1) because the first couple has a smaller Miss than the second one. In fact, the e-Librarian Service aims to give an exhaustive answer in the first place, i.e., to yield an answer that covers the user's query as much as possible, even if there is more information in the answer than required. Only in the second place, the Rest is considered in order to rank the results that have the same Miss.

3.3 Algorithm for the LO Retrieval Problem

Our best cover algorithm is called LOFind (see figure 1). As input a query Q is expected that was translated into a $\mathcal{L}$-concept description, and a $\mathcal{L}$-terminology $\mathcal{T}$, i.e., a set of semantic descriptions of LOs. The output of LOFind is the set E of best covers w.r.t. Q using $\mathcal{T}$.

Require: a query $Q \not\equiv \bot$, a set of concept descriptions $C_{\mathcal{T}} = \{C_i \not\equiv \bot, i \in [1, n]\}$
Ensure: a set of best covers $E = \{C_j \in C_{\mathcal{T}}, j \in [0..n]\}$
1: $E \leftarrow \emptyset$
2: $MinMiss \leftarrow +\infty$
3: **for** each $C_i \in C_{\mathcal{T}}$ **do**
4:   **if** $Q - lcs(Q, C_i \not\equiv Q)$ **then**
5:     **if** $|Miss(Q, C_i)| < MinMiss$ **then**
6:       $E \leftarrow C_i$
7:       $MinMiss \leftarrow |Miss(Q, C_i)|$
8:     **else if** $|Miss(Q, C_i)| = MinMiss$ **then**
9:       $E \leftarrow E \cup C_i$
10:     **end if**
11:   **end if**
12: **end for**

Fig. 1. The algorithm LOFind

The algorithm works as follows. Let us suppose that $C_{\mathcal{T}}$ is the set of semantic descriptions of the LOs in our knowledge base. Then, each LO is tested if it is a cover (line 4). If so, then it will only be maintained, if either the size of its Miss is smaller than (line 5) or equal to (line 8) the smallest Miss found up to now. In the first case, the current LO replaces all the former best cover-candidates (lines 6 + 7). In the second case, the current LO is added to the best cover-candidates found up to now (line 9).

3.4 Illustrating Example

$\mathsf{LO}_1 \equiv \mathsf{Protocol}$
$\mathsf{LO}_2 \equiv \exists \mathsf{howWorks} \sqcap \mathsf{TCP/IP}$
$\mathsf{LO}_3 \equiv \mathsf{Protocol} \sqcap \exists \mathsf{hasTask.ErrorHandling}$
$\mathsf{LO}_4 \equiv \mathsf{Protocol} \sqcap \exists \mathsf{hasTask.FlowControl}$
$\mathsf{LO}_5 \equiv \mathsf{FlowControl}$

Fig. 2. Example of a terminology of LO definitions.

For the sake of simplicity, let us suppose that there are 5 LOs in the knowledge base. The corresponding semantic descriptions are shown in figure 2. We use the

DL sub-language $\mathcal{EL}$ that has structural subsumption and allows conjunction ($\sqcap$), existential restriction ($\exists r.C$), and the top concept ($\top$). The content of the LOs deals with the following topics:

LO_1: information about protocols in general,
LO_2: explanation how the protocol TCP/IP works,
LO_3: explanation that error handling is a task of a protocol,
LO_4: explanation that flow control is a task of a protocol,
LO_5: explanation of flow control.

Step 1: Expanding the Terminology. Expanding the terminology means, making explicit some implicit knowledge. The expanded terminology uses the example taxonomy about networking (see figure 3) and is shown in figure 4.

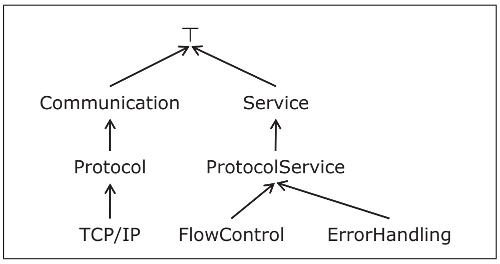

Fig. 3. Sample of a taxonomy about networking.

$$
\begin{aligned}
&\mathsf{LO}_1 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication}\\
&\mathsf{LO}_2 \equiv \exists \mathsf{howWorks} \sqcap \mathsf{TCP/IP} \sqcap \mathsf{Protocol} \sqcap \mathsf{Communication}\\
&\mathsf{LO}_3 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.(\mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})\\
&\mathsf{LO}_4 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.(\mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})\\
&\mathsf{LO}_5 \equiv \mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service}
\end{aligned}
$$

Fig. 4. Example of an expanded terminology.

Step 2: Computing the Covers. Let us suppose that the user has entered the NL question "*What are the tasks of TCP/IP?*", and that the question was translated into the following $\mathcal{EL}$-concept description: Q $\equiv$ $\mathsf{TCP/IP} \sqcap \exists \mathsf{hasTask}$. In the expanded form the user's question can be denoted as:

$$Q \equiv \mathsf{TCP/IP} \sqcap \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.$$

The aim is now to identify the LOs within the expanded terminology that cover the expanded query, i.e., that have something in common with Q; these are: LO_1, LO_2, LO_3, and LO_4.

Step 3: Computing the Best Cover. Now, for each cover the according Miss and Rest have to be computed. The best cover is the one with minimal Miss and Rest, with a preference to the minimal Miss.

	size of the Miss	size of the Rest
LO_1	$\lvert \mathsf{TCP/IP} \sqcap \exists \mathsf{hasTask} \rvert = 3$	$\lvert \top \rvert = 0$
LO_2	$\lvert \exists \mathsf{hasTask} \rvert = 2$	$\lvert \exists \mathsf{howWorks} \rvert = 2$
LO_3	$\lvert \mathsf{TCP/IP} \rvert = 1$	$\lvert \mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service} \rvert = 3$
LO_4	$\lvert \mathsf{TCP/IP} \rvert = 1$	$\lvert \mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service} \rvert = 3$

Conclusion: LO_3 and LO_4 are the best covers and are delivered as an answer to the user's query. Both LOs have the same Miss and Rest, 1 and 3, respectively so that their rank is the same. It is interesting to mention that the concept TCP/IP does not appear in one of the best covers, although it appears in the query and in LO_1. This shows that the best cover is not computed on a statistical evaluation of keywords, but that it is in fact the result of the logical inference.

Other covers, usually those where the size of the Miss is greater by one than the size of the Miss of the best cover, are yielded as second choice, here: LO_2.

4 Evaluation

Our algorithm was compared in a benchmark test with a traditional keyword-based search engine. Unfortunately, no similar measurements are available for the related projects referred in section 2.

4.1 Knowledge Base and Set of Questions

We used the online tele-TASK archive[2] that contains hundreds of recorded university lectures, as knowledge base. We selected the lecture series about Internetworking, which is a set of 30 units with a total of 38 hours of recorded lectures. We split the 30 lecture units into 1000 smaller LOs. A set of 123 NL questions about the topic Internetworking has been created. We tried to work out questions as students would ask, e.g., "*What is an IP-address composed of?*", "*How does a datapacket find its way through a network?*", "*What is a switch good for?*", "*Do internetprotocols guarantee an error-free communication?*". We also indicated for each question the relevant answer(s) that should be delivered.

[2] `http://www.tele-task.de/`

4.2 Evaluation Constraints

We call an answer from the e-Librarian Service a *perfect hit* if it covers the query completely, i.e., where the Miss and the Rest compute to zero. We call an answer from the e-Librarian Service a *sufficient hit* if it covers the query completely, but the answer contains more information than necessary, i.e., where the Miss equals zero and the Rest computes to some positive value.

For the evaluation we only considered the best covers with minimal Miss, not the second choices. This means that if the e-Librarian Service did not deliver an exhaustive answer as best cover but only as second choice, then we considered the answer to be wrong.

The results achieved with our e-Librarian Service have been compared with the results of a traditional keyword-based search engine. The keyword-based search engine is working in the usual way by browsing the textual content of the LOs. The textual content was generated by converting the PowerPoint-slides into pure text. A LO is considered to be a potential answer, if at least one (relevant) keyword from the user's query can be found. The keyword-based search engine does not consider stop words, i.e., words with no semantic relevance.

4.3 Benchmark Results

The benchmark test was performed on a standard Windows XP computer with a 1.4 GHz CPU and 512 MB of RAM. The e-Librarian Service has been implemented as a Java application. The processing time of the first question is about 200 ms, while for the rest it is less than 10 ms. The outcomes of the benchmark test are the following.

First, the e-Librarian Service scored better than the keyword search regarding the pertinence of the results. In most cases the e-Librarian Service yielded the correct answer:

	perfect hits	sufficient hits	total queries
e-Librarian Service	93 (76%)	112 (91%)	123 (100%)
Keyword search	9 (7%)	103 (84%)	123 (100%)

These numbers emphasize the pertinence of our e-Librarian Service as an appropriate tool for an educational environment; in most cases the learner gets a satisfying, even perfect, answer from the system. The fact that some answers contain little more information than necessary is no problem at all and can even have a positive effect for the learner.

Second, the precision of our solution is confirmed by the fact that in average less than 0.7 LOs are delivered in addition to the perfect answer (compared to 6 LOs for the keyword-based search). Figure 5 shows the number of supplementary LOs being delivered in addition to the expected answer. This important outcome points out that the e-Librarian Service usually is achieving the correct answer with no additional information (for 93 out of 123), and in a few cases one (12 out of 123) or two (6 out of 123) supplementary LOs. The keyword-based search engine in general delivers a lot more of secondary LOs.

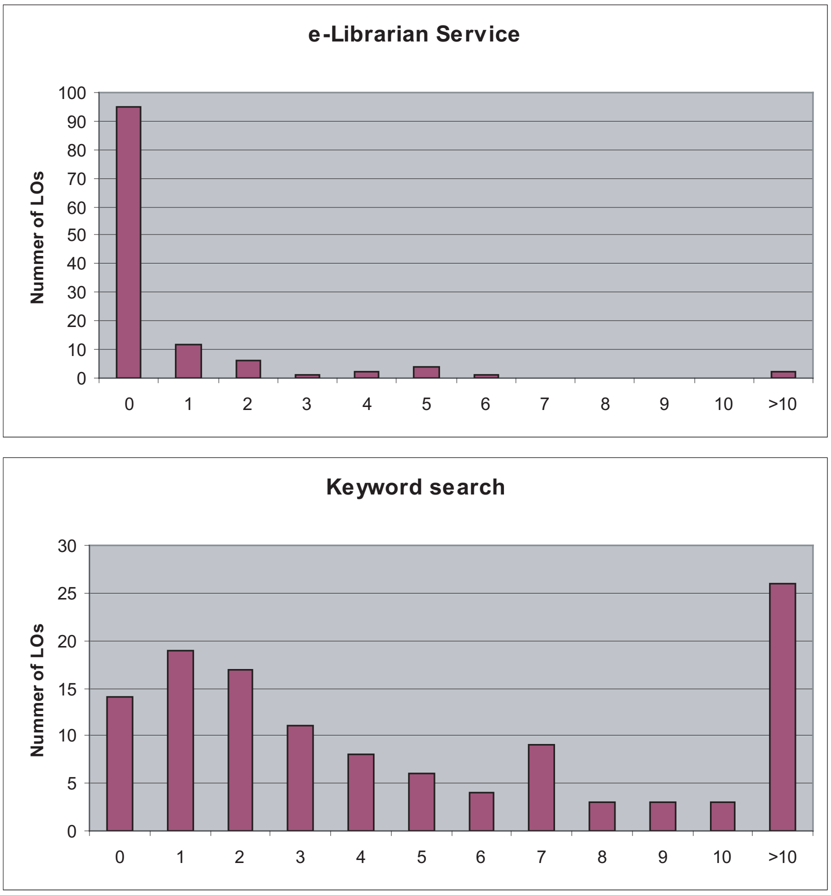

Fig. 5. Number of supplementary LOs yielded with the optimal answer.

This result is an important evidence for the pertinence of our tool in an educational environment; the user asks a precise question (or enters a keyword phrase) and expects few but concise answers. However, the keyword-based search leaves the user with the awkward task of filtering the pertinent answers out of the noise.

Third, in information retrieval the performance of a retrieval algorithm is measured by *recall* and *precision* [3]. Let use emphasize that for each question in the test set, there are only few relevant documents to be retrieved (in average 1.29 relevant answers per question). For this reason, we refer only to an average recall-level rather than to the 11 standard recall-levels. For an average recall-level, the precision of the e-Librarian Service is 84.41%, compared to 40.42% for the keyword-based search. These numbers confirm the previous outcome that our algorithm has a very high precision about the pertinence of the yielded answers; its average precision is more than twice as much than the precision achieved with the keyword-based search.

5 Improving Search Result Quality with User Feedback

As shown in the previous section, our e-Librarian Service is able to provide sufficient, even perfect, answers for most user questions. To further improve the quality of the search results, we decided to make use of the user's intellectual capabilities. The user has the possibility to vote for appropriate answers. Furthermore, we discuss possible diversification of user feedback and address the problem of general scalability of the e-Librarian Service.

5.1 Direct User Feedback

Direct user feedback can be achieved in different forms. The most simple way is to let the user determine whether a given result set of LOs really is appropriate according to his/her question or not. As usual, the user enters a query and the e-Librarian Service returns a list of LOs ordered by their computed rank. For each result a check box is displayed and the user has the possibility to indicate the appropriateness (and therefore indirectly also the quality) of the answer by leaving a mark in the check box. The e-Librarian Service has to keep track of user feedback and to channel that data into the rank computation of the LO result sets. Of course, different users might have different opinions about the accuracy of given answers.

The e-Librarian Service faces the problem to provide both an *objective answer* as well as a feedback-driven and therefore more or less *subjective answer*. For keeping track of the user feedback, an index data structure is maintained to provide efficient access. In the index, the users' questions (translated into DL formulas) are mapped with appropriate LOs and connected with a feedback-based computed rank (feedback rank) of each LO w.r.t. the user's question. In the simplest approach, the feedback rank corresponds to the number of users giving a positive feedback. The index is of the following form:

$$< user\_question,\ \{LO,\ feedback\_rank\} > .$$

For each user question the index provides access to the most appropriate LOs according to the user given feedback.

To avoid the aforementioned problem of objective and subjective answers, the e-Librarian Service displays both the (objective) best covers and the (subjective) feedback-based results. Thus, the user has the possibility to see objectively computed results and results according to the opinion of other users. If both results fit in the way that they both display the same top-rank result, the quality of our algorithm is confirmed.

User feedback might also serve as a personalization feature. For registered users the e-Librarian service is able to provide answers that have already been confirmed by the user's personal feedback. For this reason, the index data structure has to be extended to include also a set of user names for all users that have given feedback for a distinct LO:

$$< user\_question,\ \{LO,\ feedback\_rank, \{user\_names\}\} > .$$

In the same way, a more distinguished feedback is possible by giving the user the possibility to quantify the result's accuracy of fit within a given range of numbers. Instead of marking a simple check box (range: 0/1), the user has to enter a number corresponding to the appropriateness of the result set, e.g., -3 =does not fit at all ... $+3$ =fits perfectly. Now, the index data structure has to provide the average user feedback for each LO as well as a set of user names including each user's personal feedback:

$$< user\_question,\ \{LO,\ avg.\ feedback\_rank, \{user\_name,\ user\_feedback\}\} > .$$

The probability that any two users are asking the same complex question obviously is rather low. Thus, in addition to an index entry corresponding to the complex (composite) user questions, supplementary index entries can be created for all single context literals of the DL formula that represent the user's question (by *concept literal* we refer to any atomic DL formula or its negation). There, we have to take the following into account: A complex user question might perfectly match with a given answer. But, for a single concept literal within the user's question this answer might indeed be appropriate but not perfect. Therefore, feedback-hits for complex questions have to get full feedback score, while feedback-hits for single concept literals (within the user's question) do only get a partial feedback score.

5.2 Diversification of User Feedback

Besides taking into account simple user feedback data, the question, if a given LO is well suited to provide the right answer to a user's question also depends on the user's expectations. Different users asking the same question might expect different answers. This comes, because different users prefer different levels of complexity, of difficulty, and of elaborateness [15]. Moreover, different users come from different background, have different motivations, and thus, different context.

Simple user feedback can be extended in different dimensions by providing facilities to express the users customized requirements and by giving the user the possibility to quantify those characteristics for given LO result sets. The user must be able to specify, if (s)he prefers complex and precise LOs or if a short overview about the requested topic is sufficient for his/her purposes. The other way around, the user should also be able to provide feedback data about the characteristics of a given LO. In this case, for each result set of LOs several switches have to be implemented (checkboxes, text fields, or sliders) to give the user the possibility to indicate his/her opinion about the diverse qualities of the presented LOs.

If the e-Librarian Service keeps track of the user's actions, also statistics can be gathered about LO usage. If a user has already accessed and used a given LO, this information can be used to customize the computation of the best cover w.r.t. the previous knowledge of the user. Anyway, connecting the logical inference capabilities of the e-Librarian Service with sophisticated user feedback information seems to be a promising approach to augment the quality of the computed search results and will be subject of further research.

6 Conclusion

In this paper we have proposed the e-Librarian Service CHESt based on a retrieval algorithm that returns only semantically pertinent LOs from a multimedia repository w.r.t. a user's query given in NL. We have applied two non-standard inferences of DLs — the least common subsumer (lcs), and the difference operation — to compute the best cover of the user's query. The e-Librarian Service has been developed in the context of the Web University project[3], which aims at exploring novel internet- and IT-technologies in order to enhance university teaching and research. Our solution is particularly interesting for education in a self-directed learning environment, where it fosters autonomous and exploratory learning [11].

A similar e-Librarian Service for learning fractions in mathematics with a different retrieval algorithm has already been tested successfully in school [11]. We were able to measure a relevant improvement in the students' scores. This is mainly attributed to the fact that the students were more motivated by using our system — because they quickly found the pertinent answer to their question(s) — and therefore put more effort into learning and acquiring new knowledge.

Currently, we are working to improve the quality of the achieved results by implementing approaches concerning the integration of user feedback and social networking information as described in section 5.

References

1. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
2. Franz Baader, Ralf Küsters, and Ralf Molitor. Computing Least Common Subsumers in Description Logics with Existential Restrictions. In *16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 96–101, 1999.
3. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.
4. François-Marie Blondel. La recherche d'informations sur internet par des lycéens, analyse et assistance à l'apprentissage. In Peyrin J.P. Vries E., Pernin J.P., editor, *Hypermédias et Apprentissages 5 : Actes du cinquième colloque*, pages 119–133, 2001.

[3] http://www.hpi.uni-potsdam.de/~meinel/research/web_university.html

5. Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Azzurra Ragone. Semantic-based automated composition of distributed learning objects for personalized e-learning. In *European Semantic Web Conference (ESWC)*, pages 633–648, 2005.
6. Raya Fidel, Rachel K. Davies, Mary H. Douglass, Jenny K. Holder, Carla J. Hopkins, Elisabeth J. Kushner, Bryan K. Miyagishima, and Christina D. Toney. A visit to the information mall: Web searching behavior of high school students. *Journal of the American Society for Information Science*, 50(1):24–37, 1999.
7. Mohand-Saïd Hacid, Alain Leger, Christophe Rey, and Farouk Toumani. Computing concept covers: A preliminary report. In *Workshop on Description Logics*, 2002.
8. Christoph Hölscher and Gerhard Strube. Web search behavior of internet experts and newbies. *Computer Networks*, 33(1-6):337–346, 2000.
9. Ralf Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.
10. Tessa Lau and Eric Horvitz. Patterns of search: Analyzing and modeling web query refinement. In ACM Press, editor, *Proceedings of the Seventh International Conference on User Modeling*, 1999.
11. Serge Linckels, Carole Dording, and Christoph Meinel. Better results in mathematics lessons with a virtual personal teacher. In *ACM SIGUCCS*, pages 201–209, 2006.
12. Serge Linckels and Christoph Meinel. Resolving ambiguities in the semantic interpretation of natural language questions. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, volume 4224 of *LNCS*, pages 612–619, 2006.
13. Serge Linckels, Stephan Repp, Naouel Karam, and Christoph Meinel. The virtual tele-task professor: semantic search in recorded lectures. In *Technical Symposium on Computer Science Education (ACM SIGCSE)*, pages 50–54, 2007.
14. Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Conference on Information and Knowledge Management (CIKM)*, pages 375–382, 2002.
15. Ulrike Lucke, Djamshid Tavangarian, and Denny Voigt. Multidimensional educational multimedia with <ml>^{3}. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)*, 2003.
16. Philippe Martin. *Web Intelligence*, chapter Knowledge Representation, Sharing and Retrieval on the Web, pages 263–297. Springer-Verlag, 2003.
17. Dan Moldovan, Sanda Harabagiu, Roxana Girju, Paul Morarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. LCC tools for question answering. In *Text REtrieval Conference (TREC) TREC*, 2002.
18. Raquel Navarro-Prieto, Mike Scaife, and Yvonne Rogers. Cognitive strategies in web searching. In *Conference on Human Factors & the Web*, 1999.
19. Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Marina Mongiello. Abductive matchmaking using description logics. In *18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 337–342, 2003.
20. Institute of Electrical and Electronics Engineers Learning Technology Standards Committee. IEEE standard for learning object metadata (draft). IEEE standard 1484.12.1, 2002.
21. Gunnar Teege. Making the difference: a subtraction operation for description logics. In *Principles of Knowledge Representation (KR)*, pages 540–550, 1994.
22. Christine Youngblut. Educational uses of virtual reality technology. Technical Report IDA Document D-2128, Defense Advanced Research Projects Agency, `http://www.hitl.washington.edu/scivw/youngblut-edvr/D2128.pdf`, Jan 1998.