Semantic Interpretation of Natural Language User Input to Improve Search in Multimedia Knowledge Base

Semantische Interpretation einer Benutzer-Eingabe in natürlicher Sprache für eine verbesserte Suche in einer multimedialen Wissensdatenbank

Serge Linckels, Christoph Meinel, Hasso-Plattner-Institut (HPI), Potsdam

Summary In this article we present an *e-librarian service* which is able to retrieve multimedia resources from a knowledge base in a more efficient way than by browsing through an index or by using a simple keyword search. Our premise is that more pertinent results would be retrieved if the e-librarian service had a semantic search engine which understood the sense of the user's query. We explored the approach to allow the user to formulate a complete question in natural language.

We present our background theory, which is composed of three steps. Firstly, there is the linguistic pre-processing of the user question. Secondly, there is the semantic interpretation of the user question into a logical and unambiguous form, i. e., $\mathcal{ALC}$ terminology. Thirdly, there is the generation of a semantic query, and the retrieval of pertinent documents.

The background theory was implemented in two prototypes. We report on experiments that confirm the feasibility, the quality and the benefits of such an e-librarian service. From 229 different user questions, the system returned for 97% of the questions the right answer, and for nearly half of the questions only one answer (the best one). ►►► **Zusammenfassung** In diesem Artikel stellen wir unseren *E-Bibliothekar Dienst* vor, der in der Lage ist, multimediale Ressourcen aus einer Wissensdatenbank in einer effizienteren Weise zu finden als es durch das Suchen in einem Index oder durch Benutzen einer Stichwortsuche möglich ist. Wir sind überzeugt, dass konkretere Resultate gefunden werden, wenn der E-Bibliothekar Dienst über eine semantische Suchmaschine verfügt, die den Sinn der Benutzeranfrage versteht. Wir erforschen die Möglichkeit der Benutzereingabe einer ganzen Frage in natürlicher Sprache.

Unsere Lösung besteht aus drei Schritten. Erstens gibt es die linguistische Aufbereitung der Benutzerfrage. Zweitens gibt es die semantische Interpretation der Benutzerfrage in eine logische und eindeutige Form, in unserem Fall in eine $\mathcal{ALC}$ Terminologie. Drittens gibt es die Erzeugung einer semantischen Frage und das Finden der passenden Dokumente.

Unsere Lösung wurde in zwei Prototypen implementiert. Wir berichten über Experimente, die die Durchführbarkeit, die Qualität und den Nutzen eines solchen E-Bibliothekar Dienstes bestätigen. Von 229 unterschiedlichen Benutzerfragen fand unser System in 97% der Fälle die passende Antwort und für die Hälfte der Fragen nur eine einzige – und zwar die beste – Antwort.

KEYWORDS H.3.7 [Information Storage and Retrieval: Digital Libraries], K.3.1 [Computers and Education: Computer Uses in Education: Computer-Assisted Instruction], Multimedia, Semantic Search Engine, Natural Language, Information Retrieval, Performance, e-Learning

1 Introduction

Our vision is to create an *e-librarian service* which is able to retrieve multimedia resources from a knowledge base in a more efficient way than by browsing through an index or by using a simple keyword search. Our premise is that more pertinent results would be retrieved if the e-librarian service had a *semantic search engine* which understood the sense of the user's query. This requires that the user must be given the means to enter semantics. We explored the approach to allow the user to formulate a complete question in natural language (NL). Linguistic relations within the user's NL question and a given context, i. e., an ontology, are used to extract precise semantics and to generate a semantic query. The e-librarian service does not return the answer to the user's question, but it retrieves the most pertinent document(s) in which the user finds the answer to her/his question.

The results of our research work are, firstly, a founded background theory that improves domain search engines so that they retrieve fewer but more pertinent documents. It is based on the semantic interpretation of a complete question that is expressed in NL, which is to be translated into an unambiguous logical form, i. e., an $\mathcal{ALC}$ terminology. Then, a semantic query is generated and executed. Secondly, we provide empirical data that prove the feasibility and the effectiveness of our underlying background theory. We developed two prototypes: CHESt (*Computer History Expert System*) with a knowledge base about computer history and MatES (*Mathematics Expert System*) with a knowledge base about fractions in mathematics. We report on experiments with these prototypes that confirm the feasibility, the quality, and the benefits of such an e-librarian service. From 229 different user questions, the system returned for 97% of the questions the right answer, and for nearly half of the questions only one answer, the best one.

In this paper we focus on the translation of a complete NL question into a semantic query. This process is done in three steps: the linguistic pre-processing (Section 3), the mapping of the question to an ontology (Section 4), and the generation of a semantic query (Section 5). We present an algorithm (the focus function) that resolves ambiguities in the user question. The outcomes of the experiments are described in Section 6. We conclude with some (dis)advantages in Section 7. We start in Section 2 with some related projects.

2 Related Work

START [6] is the first question-answering system available on the Web. Several improvements have been made since it came online in 1993 [7; 8] which make of START a powerful search engine. However, the NL is not always sound, e. g., the question "What did Jodie Foster before she became an actress?" returns "I don't know what Jodie fostered before the actress became an actress". Also, the question "Who invented the transistor?" yields two answers: the inventors of the transistor, but also a description about the transistor (the answer to the question: "What is a transistor?").
AquaLog [10] is a portable question-answering system which takes queries expressed in NL and an ontology as input, and returns answers drawn from one or more knowledge bases. User questions are expressed as triples: ⟨subject, predicate, object⟩. If the several translation mechanisms fail, then the user is asked for disambiguation. The system also uses an interesting learning component to adapt to the user's "jargon". AquaLog has currently a very limited knowledge space. In a benchmark test over 76 different questions, 37 (48.68%) where handled correctly.

The prototype PRECISE [13] uses ontology technologies to map semantically tractable NL questions to the corresponding SQL query. It was tested on several hundred questions drawn from user studies over three benchmark databases. Over 80% of the questions are semantically tractable questions, which PRECISE answered correctly, and recognized the 20% it could not handle, and requests a paraphrase. The problem of finding a mapping from the tokenization to the database requires that all tokens must be distinct; questions with unknown words are not semantically tractable and cannot be handled.

FALCON [5] is an answer engine that handles questions in NL. When the question concept indicating the answer type is identified, it is mapped into an answer taxonomy. The top categories are connected to several word classes from WordNet. Also, FALCON gives a cached answer if the similar question has already been asked before; a similarity measure is calculated to see if the given question is a reformulation of a previous one. In TREC-9, FALCON generated a score of 58% for short answers and 76% for long answers, which was actually the best score.

LASSO [12] relies on a combination of syntactic and semantic techniques, and lightweight abductive inference to find answers. The search for the answer is based on a form of indexing called paragraph indexing. The advantage of processing paragraphs instead of full documents determines a faster syntactic parsing. The extraction and evaluation of the answer correctness are based on empirical abduction. A score of 55.5% for short answers and 64.5% for long answers was achieved in TREC-8.

Medicine is one of the best examples of application domains where ontologies have already been deployed at large scale and demonstrated their utility. The generation, maintenance and evolution of a Semantic Web-based ontology in the context of an information system for pathology are described in [2]. The system combines Semantic Web and

NL techniques to support a content-based storage and retrieval of medical reports and digital images.

The MKBEEM [4] mediation system allows to fill the gap between customers queries (possibly expressed in NL) and diverse specific providers offers. They provide a consensual representation of the e-commerce field allowing the exchanges independently of the language of the end user, the service, or the content provider. The dynamic discovery mechanism converts the user query into an ontological formula, then into a concept description using Description Logics. Finally, the relevant e-service is selected. The MKBEEM prototype has been validated with the languages Finnish, English, French, and Spanish, in two fields: business to consumer on-lines sales, and Web based traval/tourism services.

3 Linguistic Pre-Processing

The objective of the linguistic pre-processing step is to convert a stream of symbols into a structured stream of words, and to retrieve linguistic information about these words and the complete sentence. A search mechanism returns better results if the inference is done over a complete sentence by considering the relations between words – the syntax – than by only considering the isolated words. In fact, the syntactic structure of a sentence indicates the way words are related to each other, e. g., how the words are grouped together into phrases, which words modify which other words, and which words are of central importance in the sentence.

In our prototypes, the linguistic pre-processing is performed with a part-of-speech (POS) tagger; we use *TreeTagger* (IMS Stuttgart). The linguistic pre-processing step contributes in three points. Firstly, the word category of each word is made explicit, e. g., article, verb. Secondly, the tagger returns the canonical form (*lemma*) for each word (*token*). This considerably reduces the size of the ontology dictionary. Thirdly, the sentence is split into linguistic clauses. A linguistic clause is a triple of the form ⟨subject; verb; object⟩. Each triple is then processed individually, e. g., the question $q =$ "Who invented the transistor and who founded IBM?" is split into the two clauses: $q'_1 =$ [Who invented the transistor?], conj = [and], $q'_2 =$ [Who founded IBM?].

4 Ontology Mapping

In this section, we present the elaborated background theory for translating a linguistic pre-processed user question into a computer readable and unambiguous form w.r.t. a given ontology.

4.1 Ontology Preliminaries

The e-librarian service masters a domain language L_H over an alphabet Σ^*, which may or may not contain all the possible words L used by the user to formulate his question, so that $L_H \subseteq L \subseteq \Sigma^*$. The semantics are attached to each word by classification in the knowledge source, e. g., a dictionary, which is structured in a hierarchical way like *hyperonym*, *hyponym*, *synonym*, and *homonyms*. In most of the related projects (Section 2), an existing knowledge source is used, normally *WordNet*. The major problem of such a knowledge source is that it is not dedicated to a domain. Like other large scale dictionaries, WordNet on the one hand lacks of specific domain expressions, but on the other hand contains too much knowledge about other domains. This increases the problem of ambiguous interpretations for a given word. We created our own dictionary with only domain relevant words, and which is organized in a hierarchical way – similar to WordNet – w.r.t. our ontology. Furthermore, the size of the dictionary is considerably reduced by the fact that it contains all words from the domain language L_H only in their canonical form. This reduces also the possibility of ambiguous interpretations.

Definition 1. *A concept taxonomy $H = (V, E, v_0)$ is a directed acyclic graph where each node, except the root-node (v_0), has one or more parents. E is the set of all edges and V is the set of all nodes (vertices) with $V = \{(s, T) \mid s \in S\}$ where s is a unique label, S the set of all labels in the ontology, and T is a set of words from L_H that are associated to a node so that $T \subseteq L_H$.*

An example of a concept taxonomy about computer history is given in Fig. 1. Here, a document describing the transistor would be placed in the concept "EComponent" (electronic components), which is a hyponym of "Hardware".

A node v_i represents a concept. The words that refer to this concept are regrouped in T_i. We assume that each set of words T_i is semantically related to the concept that the node v_i represents. The example in Fig. 2 shows that words like "transistor", "diode" or "LED" semantically refer to the same concept, namely electronic components. Therefore, these three words are synonyms in the given

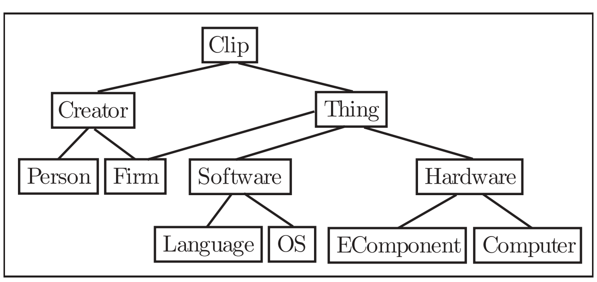

Figure 1 Example of a concept taxonomy about computer history.

Electronic components
$s = EComponent$
$T = \{\text{Transistor, Diode, LED}\}$

Figure 2 Example of a node in the taxonomy about the concept EComponent (electronic components).

ontology. Of course, a certain word can refer to different concepts; e. g., "Ada" is the name of a programming language but also the name of a person. Not all words in L_H must be associated with a concept. Only words that are semantically relevant are classified. In general, nouns and verbs are best indicators of the sense of a question. The difference between words that are semantically irrelevant and words that are not contained in L_H is that for the second ones, the system has absolutely no idea if they are relevant or not.

4.2 Semantic Interpretation

The representation of context-independent meaning is called the *logical form*, and the process of mapping a sentence to its logical form is called *semantic interpretation* [1]. The logical form is expressed in a certain knowledge representation language; we use *Description Logics* (DL). Firstly, DL have the advantage that they come with well defined semantics and correct algorithms. Furthermore, the link between DL and NL has already been established [15]. Finally, translating the user question into DL allows direct reasoning over the OWL-DL encoded knowledge base (Section 5).

A DL terminology is composed, firstly, of *concepts* (unary predicates), which are generally nouns, question words (*w-words*) and proper names, and secondly, of *roles* (binary predicates), which are generally verbs, adjectives and adverbs. We use the language $\mathcal{ALC}$ [16], which is sufficiently expressive for our purposes. $\mathcal{ALC}$ concepts are built using a set of concept names (NC) and role names (NR). Valid concepts (C) are defined by the following syntax, $C ::= A \mid \top \mid \bot \mid \neg A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \forall R.C \mid \exists R.C$ with $A \in \text{NC}$ is a concept name and $R \in \text{NR}$ is a role name (Fig. 3).

A core part of the semantic interpretation is a mapping algorithm. This step – commonly called *non-standard inference* [9] – maps each word from the user question to one or more ontology concepts/roles, and resolves the arguments of each role by analyzing the syntactic structure of the sentence.

Definition 2. *The function $\pi : L, L \to \mathbb{R}$ quantifies the* similarity *of two given words $\pi(a,b)$ so that a and b are said to be equivalent w.r.t. a given tolerance ε, written $a \equiv b$, iff $\pi(a,b) \leq \varepsilon$.*

Definition 3. *The meaning of each word $w_k \in L$ is made explicit with the* mapping function *$\varphi : L \to V$ over an ontology dictionary $L_H \subseteq L \subseteq \Sigma^*$ and an $\mathcal{ALC}$ concept taxonomy $H = (V, E, v_0)$ so that $\varphi(w_k)$ returns a set of interpretations Φ defined as follows, $\Phi = \varphi(w_k)$ and*

$$\Phi = \{v_i \mid \exists x \in ft(v_i) : w_k \equiv x\} .$$

The function $ft(v_i)$ returns the set of words T_i associated to the node v_i (Def. 1), and $w_k \equiv x$ are two equivalent words. Technically, for a given lemma from the user question, the equivalence function π uses the *Levenshtein function* to check if this word is contained in the ontology dictionary L_H given a certain allowed tolerance ε. That tolerance is calculated relative to the length of the lemma.

Applying the Levenshtein function is not necessary if the morphological analyzer of the POS tagger identifies the exact token. But it is necessary if the tagger is not able to identify the word, e. g., if the user makes spelling errors. Furthermore, only the best matching is considered for the mapping, e. g., the word "comXmon" will be considered as "common", and not as "uncommon". Both words, "common" and "uncommon", will be considered for the mapping of "comXXmon". The ambiguity will be resolved in a further step (focus function).

Definition 4. *A word w_k is* semantically relevant *if there is at least one concept in the ontology H to which w_k can be mapped so that $\varphi(w_k) \neq \emptyset$.*

It is possible that a word can be mapped to different concepts at once, so that $|\Phi| > 1$. We introduce the notion of *focus* to resolve this ambiguity. The focus is a function (f) which returns the best interpretation for a given word in the context of the complete user question.

Definition 5. *The* focus *of a set of interpretations Φ is made explicit by the function f which returns the best interpretation for a given word in the context of the complete question q. The focus, written $f_q(\varphi(w_k \in q)) = v'$, guarantees the following,*

1. *$v' \in \varphi(w_k)$; The focused word is a valid interpretation.*
2. *$|f_q(\varphi(w_k))| = [0,1]$; The focus function returns 0 or 1 result.*

$$\begin{aligned}
Clip &\doteq \exists hasName.String \sqcap Creator \sqcup Thing \\
Creator &\doteq Person \sqcup Firm \\
Person &\doteq \exists wasBorn.Date \sqcap \exists isDeceased.Date \\
Thing &\doteq Firm \sqcup Software \sqcup Hardware \sqcup Net \\
&\quad \sqcap \exists wasInventedBy.Creator \\
Software &\doteq Language \sqcup OS \\
Hardware &\doteq EComponent \sqcup Computer
\end{aligned}$$

Figure 3 Example of a concept taxonomy (TBox) about computer history as $\mathcal{ALC}$ terminology.

3. $\top \leq v' \leq \bot$, *if* $f_q(\varphi(w_k)) \neq \emptyset$; *If the focusing is successful, then the word is inside the context of the domain ontology.*
4. $\pi(w_k, x \in ft(v')) \leq \pi(w_k, y \in ft(v_i \in \varphi(w_k)))$; *The returned interpretation contains the best matching word of all possible interpretations.*

Let us consider as illustration the word "Ada", which is called a multiple-sense word. In fact, in the context of computer history, "Ada" can refer to the programming language named "Ada", but it can also be the name of the person "Augusta Ada Lovelace". The correct interpretation can only be retrieved accurately by putting the ambiguous word in the context of a complete question. For example, the context of the sentences "Who invented Ada?" and "Did the firms Bull and Honeywell create Ada?" reveals that here Ada is the programming language, and not the person Ada.

Technically, the focus function uses the role's signature. A role $r \in \mathrm{NR}$ has the signature $r(s_1, s_2)$, where s_1 and s_2 are labels. The signature of each role defines the kind of arguments that are possible. For example *wasInventedBy*(*Thing,Creator*) is the role $r = wasInventedBy$ that has the arguments $s_1 = Thing$ and $s_2 = Creator$.

In the question $q =$"Who invented Ada?", "invented" is mapped to the role *wasInventedBy*, and "Who" is mapped to the concept *Creator*. The system detects an ambiguity for the word "Ada", which is mapped to an instance of the concept *Person*, but also to an instance of the concept *Language*, so that

$$\varphi(\text{"Ada"}) = \{Person, Language\}\,.$$

The focus function computes the following combinations to resolve the ambiguity:

1. Was Ada invented by who?*
2. Was Ada invented by Ada?
3. Was who invented by Ada?*
4. Was who invented by who?*

Cyclic combinations like (2) and (4) are not allowed. As for (3), it does not match the role's signature because $s_1 = Creator$ ("Who"), but *Thing* is required. As for (1), s_1 can be *Person* or *Language* ("Ada"). The role's signature requires *Thing*, therefore *Person* is excluded as valid interpretation because $Person \not\sqsubseteq Thing$. As $Language \sqsubseteq Thing$, a valid interpretation is found, and in the context of this question the word "Ada" refers to the programming language Ada. Finally, the result of the focus function is:

$$f_q(\varphi(\text{"Ada"})) = Language\,.$$

In deed, (1) represents the question "Who invented Ada?". It is still possible that the focus function cannot resolve an ambiguity, e. g., a given word has more interpretations but the focus function returns no result. In a such case, the system will generate a semantic query for each possible interpretation. Based on our practical experience we know that users generally enter simple questions where the disambiguation is normally successful.

Definition 6. *Let q be the user question, which is composed of linguistic clauses, written $q = \{q'_1, \ldots, q'_m\}$, with $m \geq 1$. The* sematic interpretation *of a user question q is the translation of each linguistic clause into an $\mathcal{ALC}$ terminology w.r.t. a given ontology H written,*

$$q_i^H = \prod_{k=1}^{n} f_{q'_i}\left(\varphi(w_k \in q'_i)\right)$$

with q'_i a linguistic clause $q'_i \in q$, and n the number of words in the linguistic clause q'_i.

5 Query Generation

We will start with the assumptions that firstly, all documents in the knowledge base $\mathcal{K}$ are semantically described with metadata, i. e., using OWL-DL (Fig. 5) w.r.t. an ontology H, and that secondly the user question q was translated into a DL terminology w.r.t. the same ontology H (Section 4). Even if we currently do not profit from the full expressivity of OWL-DL, which is $\mathcal{SHOIN}(D)$, it allows to have compatible semantics between the OWL-DL knowledge base, and the less expressive $\mathcal{ALC}$ user question. Logical inference over the non-empty ABox from $\mathcal{K}$ is possible by using a classical DL reasoner; we use *Pellet* [17]. The returned results are logical consequences of the inference rather than of keyword matchings.

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$, the domain of the interpretation, and an interpretation function $\cdot^{\mathcal{I}}$ that maps each concept name to a subset of $\Delta^{\mathcal{I}}$ and each role name to a binary relation $r^{\mathcal{I}}$, subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Definition 7. *A* semantic query *over a knowledge base $\mathcal{K}$ w.r.t. an ontology H, and an user question q is an ABox query, which means to search for models $\mathcal{I}$ of $\mathcal{K}$, written $\mathcal{K} \models q^H$.*

In other words, all documents from the knowledge base that satisfy the expression q^H are potential results. An individual α in $\mathcal{I}$ that is an element of $(q^H)^{\mathcal{I}}$ is a pertinent resource according to the user question.

Technically, an ABox query (in Pellet) is expressed in a query language; we use RDQL [11] via the Jena framework [3]. Firstly, for a complete question, each semantic interpretation, that is each translated linguistic clause, is transformed into a semantic query. Secondly, the nature of the question (*open* or *close*) reveals the missing part. An *open question* contains a question word, e. g., "Who invented the transistor?", whereas a *close question* (logical- or yes/no question) does not have a question word, e. g., "Did Shockley contribute to the invention of the transistor?". As for the first kind of questions, the missing part – normally not an individual but a concept – is the subject of the question and therefore the requested result. The result of the query is the set of all models $\mathcal{I}$ in the knowledge

$q =$ "Wer hat den Transistor erfunden?"

↓ Linguistic pre-processing

$$q' = \left[\text{Wer}\ {}^{[wer]}_{\text{WPRO}} \quad \text{hat}\ {}^{[haben]}_{\text{VHFIN}} \quad \text{den}\ {}^{[der]}_{\text{ART}} \quad \text{Transistor}\ {}^{[Transistor]}_{\text{NN}} \quad \text{erfunden}\ {}^{[erfinden]}_{\text{VVpast}} \right]$$

↓ Semantic interpretation

$$q^H = Creator(x1) \sqcap \exists wasInventedBy(x2, x1) \sqcap EComponent(x2) \sqcap \exists hasName(x2, "Transistor")$$

↓ Semantic query generation

```
SELECT ?x1 WHERE (?x2 rdf:type chest:EComponent) (?x2 chest:hasName ?x2hasName) (?x2
chest:wasInventedBy ?x1) AND (?x2hasName =~/Transistor/i) USING chest for <...> rdf for <...>
```

Figure 4 Complete example for the generation of a semantic query from the user question "Who invented the transistor?".

```
<owl:Class rdf:about="EComponent">
  <rdfs:subClassOf rdf:resource="Hardware" />
</owl:Class>

<owl:ObjectProperty rdf:about="wasInventedBy">
  <rdfs:domain rdf:resource="Thing" />
  <rdfs:range rdf:resource="Creator" />
</owl:ObjectProperty>

<EComponent rdf:about=".../transistor.rm">
  <hasName>Transistor</hasName>
  <hasCreationYear>1947</hasCreationYear>
  <wasInventedBy rdf:resource="../shockley.rm" />
  <wasInventedBy rdf:resource="../bardeen.rm" />
  <wasInventedBy rdf:resource="../brattain.rm" />
</EComponent>
```

Figure 5 Example of the concept taxonomy (TBox) serialized as OWL.

base $\mathcal{K}$. As for the second kind of questions, there is no missing part. Therefore, the answer will be "yes" if $\mathcal{K} \models q^H$, otherwise it is "no". A complete example is shown in Fig. 4.

If a user question is composed of several linguistic clauses, then each one is translated separately. The logical concatenation of the different interpreted clauses q_i^H depends on the conjunction word(s) used in the user question, e. g., "Who invented the transistor *and* who founded IBM?". If no such conjunction word is found, then the "or" operator ($\sqcup$) is preferred over the "and" operator ($\sqcap$).

6 Implementation and Experiments

Our background theory was implemented prototypically in two educational tools; one about computer history (CHESt), and one about fractions in mathematics (MatES). Both prototypes can be used at home or in a classroom either as Web application, or as stand-alone application (e. g., from a DVD/CD-ROM). The user can freely formulate a question in NL, and submit it to the e-librarian service. Then, the e-librarian service returns one (or more) document(s) which explain(s) the answer to the user's question (Fig. 6). The knowledge base is composed of short multimedia documents (*clips*), which were recorded with tele-TASK (http://www.tele-task.de) [14]. Each clip documents one subject or a part of a subject. The duration of each clip varies from several seconds to three or four minutes. This has two reasons, firstly, the younger the user, the shorter the time during which (s)he will concentrate on the information displayed on the screen [18]. Secondly, it is not easy to find the appropriate information inside a large piece of data, e. g., in an online lesson that lasts 90 minutes.

In a **first experiment** made in a secondary school with CHESt, we aimed to investigate, firstly, how useful our e-librarian service is as an e-learning tool, and secondly, in how far students accept to enter complete questions into a search engine instead of only keywords. Some 60 students took part in the assessment. In the first place, let us point out that nearly all students approved of the appealing multimedia presentations. They agreed that the explanations were sufficiently complete to understand the subject. Several appreciated the short length of the clips; a few stated that the clips were too long. Some added that they appreciated the short response time of the system. Finally, asked if they accepted to enter complete questions into a search engine, 22% of the students answered that they would accept, 69% accepted to enter complete questions instead of keywords only if this yielded better results, and 8% disliked this option.

In a **second experiment** we used MatES to measure the performance of our semantic search engine. A testing set of 229 different questions about this topic was created by a mathematic teacher, who was not involved in the development of the prototype. The teacher also indicated manually the best possible clip, as well as a list of further

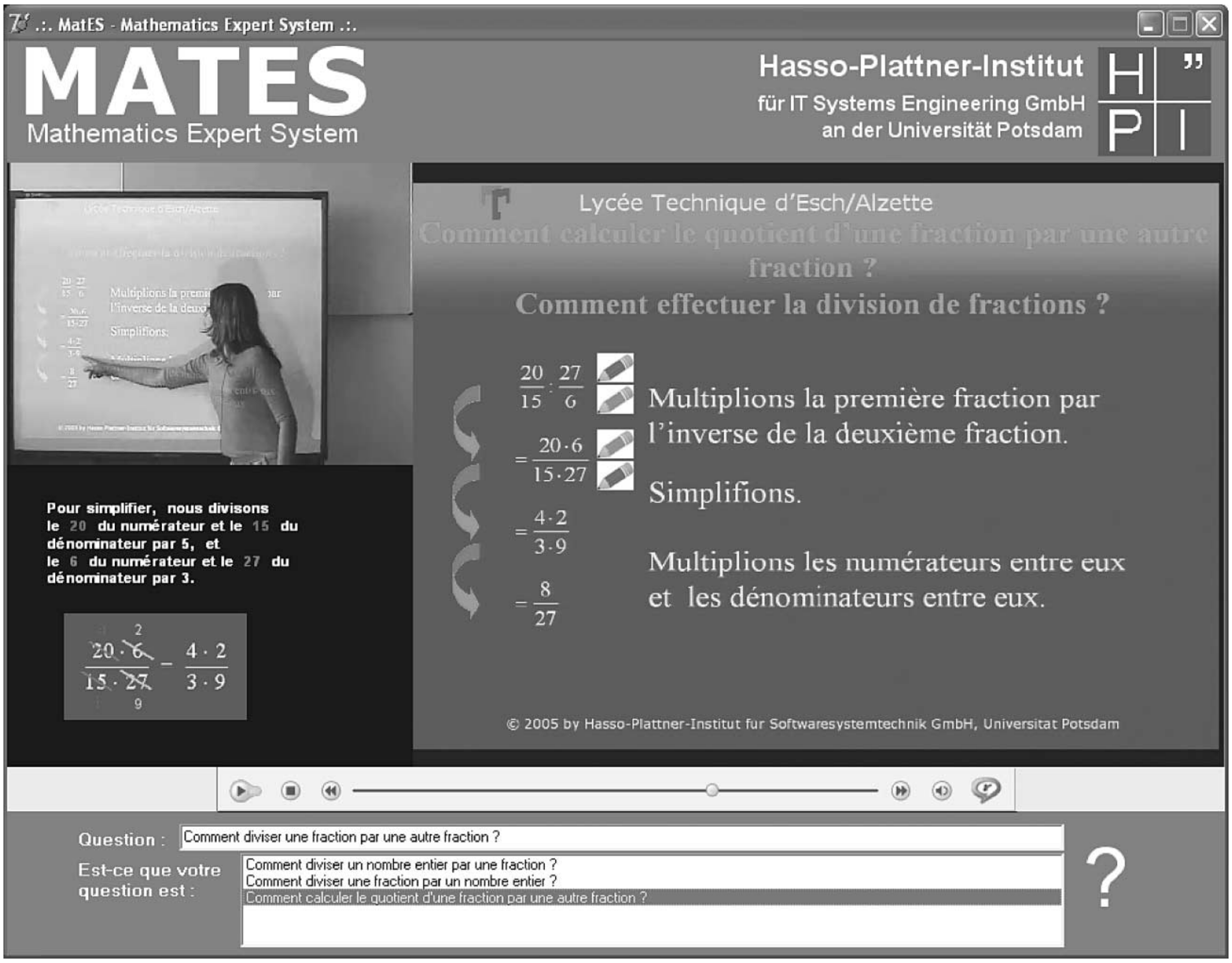

Figure 6 MatES with the question: "How to divide one fraction by another fraction?".

clips, that should be yielded as correct answer. The questions were linguistic correct, and short sentences like students in a secondary school would ask, e. g., "How can I simplify a fraction?", "What is the sum of $\frac{2}{3}$ and $\frac{7}{4}$?", "What are fractions good for?", "Who invented the fractions?", etc. This benchmark test was compared with the performance of a keyword search engine. The keyword search was slightly optimized to filter out stop words (words with no relevance, e. g., articles) from the textual content of the knowledge base and from the questions entered. The semantic search engine answered 97% of the questions (223 out of 229) correctly, whereas the keyword search engine yielded only a correct answer (i. e., a pertinent clip) in 70% of the questions (161 out of 229).

It is also interesting to notice that for 86 questions, the semantic search engine yielded just one – the semantically best matching – answer

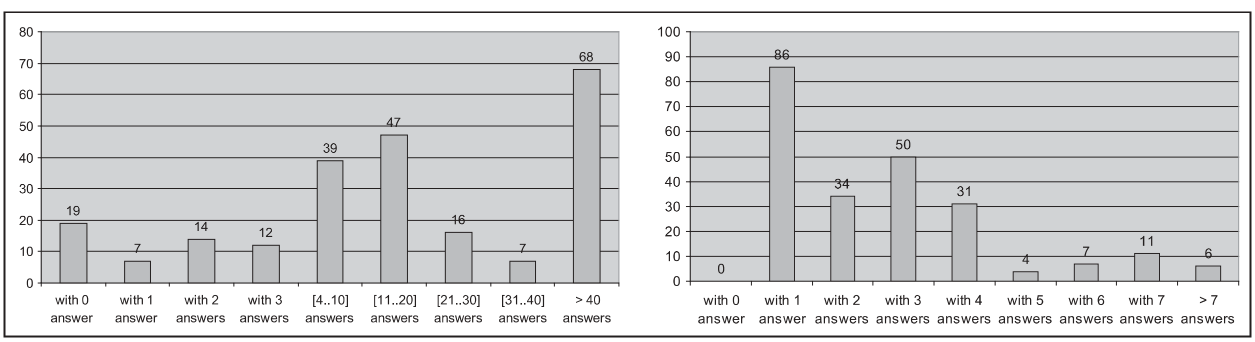

Figure 7 Number of results yielded by a (1) keyword and by a (2) semantic search engine with a set of 229 questions.

(Fig. 7). For 75% of the questions (170 out of 229) the semantic search engine yielded just a few results (one, two or three answers), whereas the keyword search yielded for only 14% of the questions less than 4 answers; mostly (138 questions out of 229) more than 10 answers. Our e-librarian service returned always at least one result. This is important because we learned from former experiments in school that students dislike getting no result at all.

For example, the semantic interpretation of the question "What is the sum of $\frac{2}{3}$ and $\frac{7}{4}$?" is the following valid $\mathcal{ALC}$ terminology: $Fraction(x1) \sqcap \exists hasOperation(x1, x2) \sqcap Operation(x2, sum)$.

Then the semantic query retrieves one clip, which explained how to add two fractions. This was the best clip that could be found in the knowledge base[1]. This means also that questions like "How can I add two fractions", "What is $\frac{11}{0.5}$ plus $\frac{5}{5}$", etc. would yield the same clip. The keyword search engine yields all clips, in which keywords like "sum" are found, e. g., a clip that explains how to represent a complex function in terms of additions, and a clip that explain how to describe situations with simple fractions.

The experiments revealed also two major weaknesses of our e-librarian service that should be improved in future. Firstly, the system is not able to make the difference between a question, where there is no answer in the knowledge base, and a question that is out of the topic, e. g., "Who invented penicillin?". Secondly, in its current state, the e-librarian service does not handle number restrictions, e. g., "How many machines did Konrad Zuse invent?". The response will be the list of Zuse's machines, but not a number. Furthermore, the question "What is the designation of the third model of Apple computers?" will yield a list of all models of Apple computers.

[1] Remember that the system returns clips that explain the answer to the student's question, but they do note give the precise answer, e. g., it does not compute the sum of the two fractions.

7 Conclusion

In this paper we presented an e-librarian service that allows the user to communicate by means of complete questions in NL, and that retrieves pertinent multimedia resources from a knowledge base. The background theory is composed of three steps: the linguistic pre-processing of the user's NL input, the semantic interpretation of the NL sentence into a logical form, and the generation of a semantic query. It uses Description Logics and Semantic Web technologies like OWL for the semantic interpretation of NL questions. We also presented an algorithm to resolve ambiguities in the user question. Experiments with two prototypes confirmed that this background theory is reliable and can be implemented, e. g., in an educational tool.

In our further work, we will try to improve the translation from the NL question into an $\mathcal{ALC}$ terminology, e. g., use number restrictions. We also want to investigate if a more precise grammatical analyze of the user question can help in the interpretation step, or if this would reduce the users liking of the interface (because of the smaller tolerance of the system). Another important topic is the maintenance facilities; how can unknown words from the user query (i. e., the user's "jargon") be included in the dictionary, and how can external "thrusted" knowledge sources be accessed by the e-librarian service?

References

[1] J. Allen. *Natural Language Understanding*. Addison Wesley, 1994.

[2] E. P. Bontas, S. Tietz, R. Tolksdorf, and T. Schrader. Engineering a semantic web for pathology. In *Int'l Conf. Web Engineering, Munich, Germany*, volume 3140 of *Lecture Notes in Computer Science*, pages 585–586, 2004.

[3] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: implementing the semantic web recommendations. In *13th Int' Conf. on World Wide Web – Alternate Track Papers & Posters, New York, NY, USA, May 17–20*, pages 74–83, 2004.

[4] Ó. Corcho, A. Gómez-Pérez, A. Leger, C. Rey, and F. Toumani. An ontology-based mediation architecture for e-commerce applications. In *Intelligent Information Processing and Web Mining (IIS: IIPWM), Zakopane, Poland*, pages 477–486, 2003.

[5] S. M. Harabagiu, D. I. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. C. Bunescu, R. Girju, V. Rus, and P. Morarescu. Falcon: Boosting knowledge for answer engines. In *9th Text Retrieval Conf. (TREC 2000), Gaithersburg, Maryland*, 2000.

[6] B. Katz. Annotating the world wide web using natural language. In *Computer Assisted Information Searching on the Internet (RIO), Montreal, Canada*, 1997.

[7] B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. J. Lin, G. Marton, A. J. McFarland, and B. Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *Applications of Natural Language to Information Systems (NLDB), Stockholm, Sweden*, pages 230–234, 2002.

[8] B. Katz and J. Lin. Annotating the semantic web using natural language. In *2nd Workshop on NLP and XML (NLPXML-2002) at COLING*, 2002.

[9] R. Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001. Ph.D. thesis.

[10] V. Lopez, M. Pasin, and E. Motta. Aqualog: An ontology-portable question answering system for the semantic web. In *European Semantic Web Conf. (ESWC), Heraklion, Greece*, pages 546–562, 2005.

[11] L. Miller, A. Seaborne, and A. Reggiori. Three implementations of squishql, a simple rdf query language. In *1st Int'l Semantic Web Conf. (ISWC), Sardinia, Italy, June 9–12*, pages 423–435, 2002.

[12] D. I. Moldovan, S. M. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju, and V. Rus. Lasso: A tool for surfing the answer net. In *8th Text Retrieval Conf. (TREC-8), Gaithersburg, Maryland*, 1999.

[13] A.-M. Popescu, O. Etzioni, and H. A. Kautz. Towards a theory of natural language interfaces to databases. In *Intelligent User Interfaces, Miami, FL, USA*, pages 149–157, 2003.

[14] V. Schillings and C. Meinel. tele-task: teleteaching anywhere solution kit. In *30th Annual ACM SIGUCCS Conf. on User services, Providence, Rhode Island, USA, Nov. 20–23*, pages 130–133, 2002.

[15] R. A. Schmidt. Terminological representation, natural language & relation algebra. In *German AI Conf. (GWAI)*, volume 671 of *Lecture Notes in Artificial Intelligence*, pages 357–371, 1993.

[16] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[17] E. Sirin and B. Parsia. Pellet: An owl dl reasoner. In *Int'l Workshop on Description Logics (DL2004), Whistler, Canada*, volume 104 of *CEUR Workshop Proc.*, 2004.

[18] W. M. Williams, F. Markle, R. J. Sternberg, and M. Brigockas. *Educational Psychology*. Allyn & Bacon, 2001.


1


2

1 Dipl.-Ing. Serge Linckels is computer-science teacher at the Lycée Technique d'Esch/Alzette, a technical high-school in Luxembourg, and PhD student at the Hasso-Plattner-Institut (HPI) of Prof. Dr. Christoph Meinel. His interests are the study and improvement of the way students learn, especially in a computer based learning environment.
Address: Hasso-Plattner-Institut (HPI), Prof.-Dr.-Helmert-Str. 2–3, 14482 Potsdam, Tel.: +49-0331-5509222, Fax: +49-0331-550325, E-Mail: linckels@hpi.uni-potsdam.de

2 Prof. Dr. Christoph Meinel is director of the Hasso-Plattner-Institut (HPI), and head of the chair "Internet Technologies and –Systems". His main work concerns security engineering technologies and methods, telemedicine applications and innovative forms of teaching (Teleteaching) and learning (E-Learning).
Address: Hasso-Plattner-Institut (HPI), Prof.-Dr.-Helmert-Str. 2–3, 14482 Potsdam, Tel.: +49-0331-5509222, Fax: +49-0331-550325, E-Mail: meinel@hpi.uni-potsdam.de