HASSO-PLATTNER-INSTITUT
für Softwaresystemtechnik an der Universität Potsdam

An e-Librarian Service – Natural Language Interface for an Efficient Semantic Search within Multimedia Resources

Serge Linckels, Christoph Meinel

Technische Berichte Nr. 12

des Hasso-Plattner-Instituts
für Softwaresystemtechnik an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik
an der Universität Potsdam

Nr. 12

An e-Librarian Service – Natural Language Interface for an Efficient Semantic Search within Multimedia Resources

Serge Linckels, Christoph Meinel

Potsdam 2006

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Die Reihe *Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam* erscheint aperiodisch.

Herausgeber: Professoren des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam

Redaktion: Serge Linckels, Christoph Meinel
Email: Serge.linckels; christoph.meinel}@.hpi.uni-potsdam.de

Vertrieb: Universitätsverlag Potsdam
Postfach 60 15 53
14415 Potsdam
Fon +49 (0) 331 977 4517
Fax +49 (0) 331 977 4625
e-mail: ubpub@uni-potsdamde
http://info.ub.uni-potsdam.de/verlag.htm

Druck: allprintmedia gmbH
Blomberger Weg 6a
13437 Berlin
email: info@allprint-media.de

Heft 12 (2006)
ISBN 3-937786-89-9
ISBN 978-3-937786-89-6
ISSN 1613-5652

TECHNISCHER BERICHT

An e-Librarian Service — Natural Language Interface for an Efficient Semantic Search within Multimedia Resources

Serge Linckels
http://www.linckels.lu
linckels@hpi.uni-potsdam.de

Christoph Meinel
http://www.hpi.uni-potsdam.de/~meinel
meinel@hpi.uni-potsdam.de

Contents

1 Introduction

This document gives an overview of the current state of the research project about an e-Librarian Service from Serge Linckels and Christoph Meinel from Hasso Plattner Institute (HPI) at Potsdam University, in collaboration with Luxembourg International Advanced Studies in Information Technologies (LIASIT) at Luxembourg University.

1.1 Project formulation

On the basis of the overwhelming experiences of using tele-TASK (`http://www.tele-task.de`) in university teaching, we started to investigate whether it can be used in (High) schools as well. Our vision is to create a *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 librarian service had a semantic search engine which understood the sense of the user's query. Thus, the user must be given the means to enter semantics. A simple approach to be explored is to allow the user to formulate a complete question in natural langauge (NL) and to enter it in the search engine. It is to be tested, in how far linguistic relations within the user's NL question and a given context are required to extract precise semantics and to generate a semantic query. The feasibility, the quality and the benefits of such a librarian service have to be explored and documented. An educational prototype must be developed with a corresponding knowledge base.

1.2 Our contribution

This versatile project focuses on the design, elaboration, and testing of a novel approach of retrieval systems by bringing together the most appropriated knowledge and technologies from different research domains. The result of the research work is, firstly, a founded background theory that improves domain search engines to be able to retrieve pertinent documents from a multimedia knowledge base by a semantic interpretation of a complete question expressed in NL. Secondly, we provide empirical data that prove the feasibility, and the effectiveness of our underlying background theory. This data was collected by experiments made in an educational environment with a prototype of our librarian service. Our research work covers the following aspects:

- We started by analyzing the pedagogical requirements and advantages of such a librarian service in a practical educational environment (section 2).
- We elaborated the background theory for a librarian service as ontology driven experts system, which is able to logically infer over the knowledge base (section 3).
- We express how to translate a user question which is formulated in NL into a computer readable and unambiguous form (section 4).
- We explain the process of generating semantic queries, and retrieving pertinent documents from the knowledge base (section 5).
- We enumerate some technical contributions that result from this project (section 6), e.g. three prototypes that were developed (section 6.1).
- We report on experiments that were made to test the elaborated background theory and the developed prototypes (section 7).
- We discuss some (dis)advantages of our librarian service, and give some suggestions for future work (section 8).

Academy benefits The ambition of this project is to elaborate a founded background theory to significantly improve domain ontology search engines; fewer results, but more pertinent ones are returned by our semantic search engine. This project will also contribute to current ontology research by providing a rich and documented schemata for representing ontologies about computer history, mathematics and IT security. Finally, this project can probably contribute to the discussion in how far a search engine would yield better results if the query was formulated in NL.

Industrial benefits The resulting three prototypes of this project are free to be used in schools. The large multimedia knowledge bases and the efficient semantic search engine are an attractive contribution to education and coherent with today's pedagogy. Although our librarian service was implemented in educational prototypes, it can easily be used in different commercial areas, like online helpdesks, or travel planers. We could imagine that clients requiring some help, e.g. with their Internet connection, could contact a virtual online help desk and express questions in NL. The expert system would understand the sense of the customer's question and could propose a short but pertinent answer.

2 Pedagogical Aspect

In this section we focus on the pedagogical aspect of our project. We start with a presentation of modern teaching methods and technologies. Then we describe our contribution by pointing out the exploratory learning approach, the simple human-machine interaction, and the idea of presenting the information in the form of short multimedia documents.

> *Pedagogy is the art or science of teaching. The word comes from the ancient Greek "paidagogos", the slave who took children to and from school. The word "paida" refers to children, which is why some like to make the distinction between pedagogy (teaching children) and andragogy (teaching adults).*
> Wikipedia

2.1 Modern teaching

New attention has been given to teaching methods — the pedagogy — in a computer-based educational environment. Current educational thinking aims to allow students to be able to master, retain, and generalize new knowledge when they are actively involved in constructing that knowledge in a learning-by-doing situation [You98]. This leads to the expression of *exploratory learning*, which usually emphasizes using computers as tools rather than as teachers. Furthermore, it seems that everything in school must be fun and that students must be entertained. This leads to the expression of *edutainement*.

Every person is different in her/his sense of perception. Some understand better if they hear the explanation by the means of verbal communication, some need to write it down, others must see it in the form of a text or a picture and others again have to touch it [Ows97]. A good educational tool must present the same information in different forms in order to activate as many senses as possible. The psychological foundations were proven by the work of [MG90] and [MS94]; *information that is presented at the same time in different forms improves the understanding of the information.*

Students are spoiled, even dazzled, by the attractive graphical user interfaces (GUI) of computer software and the possibilities of multimedia applications. New software without a GUI in vogue is doomed to failure. That is exactly why students prefer Web sites with colors, images, sound, and animations, rather than books as learning syllabuses. An educational tool must have an ergonomic and appropriated GUI, which is neither too complicated, nor too simple. It must allow the student to concentrate on the essence of the problem instead ov being diverted by an overloaded presentation.

The amount of "items" a student can assimilate in a given time depends strongly on his intellectual capacities, and thus is directly related to his age [WMSB01]. However, search mechanisms are often based on simple keyword searches and return large number of results. Dialoging tools seem to become more and more popular. Smart tools, which are able to deliver pertinent and useful answers to students can bee seen as a kind of virtual teacher. In a certain sense this allows students to be taught individually by computers; which teacher can actually do this?

The demand of home learning increases quickly with the raising availability of broadband access to the Internet [Bon01]. All kind of distance education and online education are more and more offered by educational institutions to allow potentially everyone to acquire new knowledge or review some topics from home.

Conclusion Today, modern schools are built with "cyberage technologies" in mind. New teaching technologies like online encyclopedias, Web search engines, virtual reality environments, etc. are more and more used in all day classes. But although the features of e-learning sound very promising, the real gain compared to traditional courses has not yet been proved. Furthermore, and as paradoxical as it seems, the more technological means are used to access information, the more difficult it is to find pertinent and correct information.

2.2 Our Contribution

Although teaching is much more than only transmitting knowledge, this is the task which can be improved by technological means. In fact, a computer tool cannot explain a difficult topic better than a teacher. It can only present the right information in another form, maybe a clearer or more exhaustive one. Our librarian service fulfills most of the needs and requirements of a modern e-learning tool that were mentioned above; it fosters autonomous and exploratory learning, it allows the user to ask questions in a very human way, and it returns pertinent and short multimedia answers. The pedagogical advantages of our librarian service were published in a more exhaustive form in [LM05a].

2.2.1 Autonomous and exploratory learning

Our librarian service can be used without special hardware requirements at home or in school, individually or in a group (section 6.1).

- At home, our librarian service can be seen as the student's personal teacher to which (s)he can place questions and get pertinent and short answers, e.g.:
  - The student can use it to do her/his homework or to review an important topic before a test.
  - The student can ask for information about topics that were not dealt with in class but which draw the student's attention, or topics of which (s)he needs further explanation for a better understanding.
- We see our librarian service as a complement to conventional lessons. It is up to the teacher to decide which is an appropriated occasion to use it, e.g.:
  - To introduce a new subject by letting the students discover new information for themselves.
  - To use it as a complement to classical syllabuses or the blackboard to find and show illustrations for certain topics in a more attractive form, i.e. multimedia documents.

In a certain sense, the student creates his/her own course content by assembling different documents. This autonomous and exploratory approach is certainly more motivating for the students. In general, more motivated students learn better and have better results in school.

2.2.2 Human machine interaction

The interaction between machines and humans is still surprisingly complicated. Searching information in a knowledge base means, browsing through an index, or formulating and entering a query normally by entering keywords. In both cases, the user must adapt himself to the machine in order to give precise and machine-readable instructions, e.g. combine keywords with Boolean operators. Also, clicking on some icons on the screen is certainly very simple for browsing through the content of a knowledge base, but it is not a very effective way of searching. Formulating a question in a computer understandable form is also not an easy task.

We investigated how to improve this interaction by allowing the user to communicate with the machine in a more human way. We mainly explored the approach of letting the user freely formulate a question in NL, i.e. by entering a complete question in German[1]. The outcome of our research is that indeed fewer but pertinent results can be found, as far as the user does not formulate too complex questions like explained in section 4.2.4. The processing of a NL question and the retrieval of documents from the knowledge base are detailed in section 4 and section 5 respectively. Experiments showed, that the large majority of the students would prefer to enter complete questions in NL instead of keywords if this yielded better results (section 7). Students were also impressed by the fast answers of the system and the interesting form in which the information was presented.

2.2.3 Short multimedia clips

Our librarian service explains requested topics with short multimedia documents (*clips*). The length of the stored items in the knowledge base (the duration of the clips) is essential in our concept. The younger the user, the shorter the time during which (s)he will concentrate on the information displayed on the screen. Furthermore, 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. Therefore, we divided all our multimedia data into small clips. The duration of each clip varies from several seconds to three or four minutes. Each clip documents one subject or a part of a subject. Together, all the clips of the knowledge base cover one large topic (section 6.1).

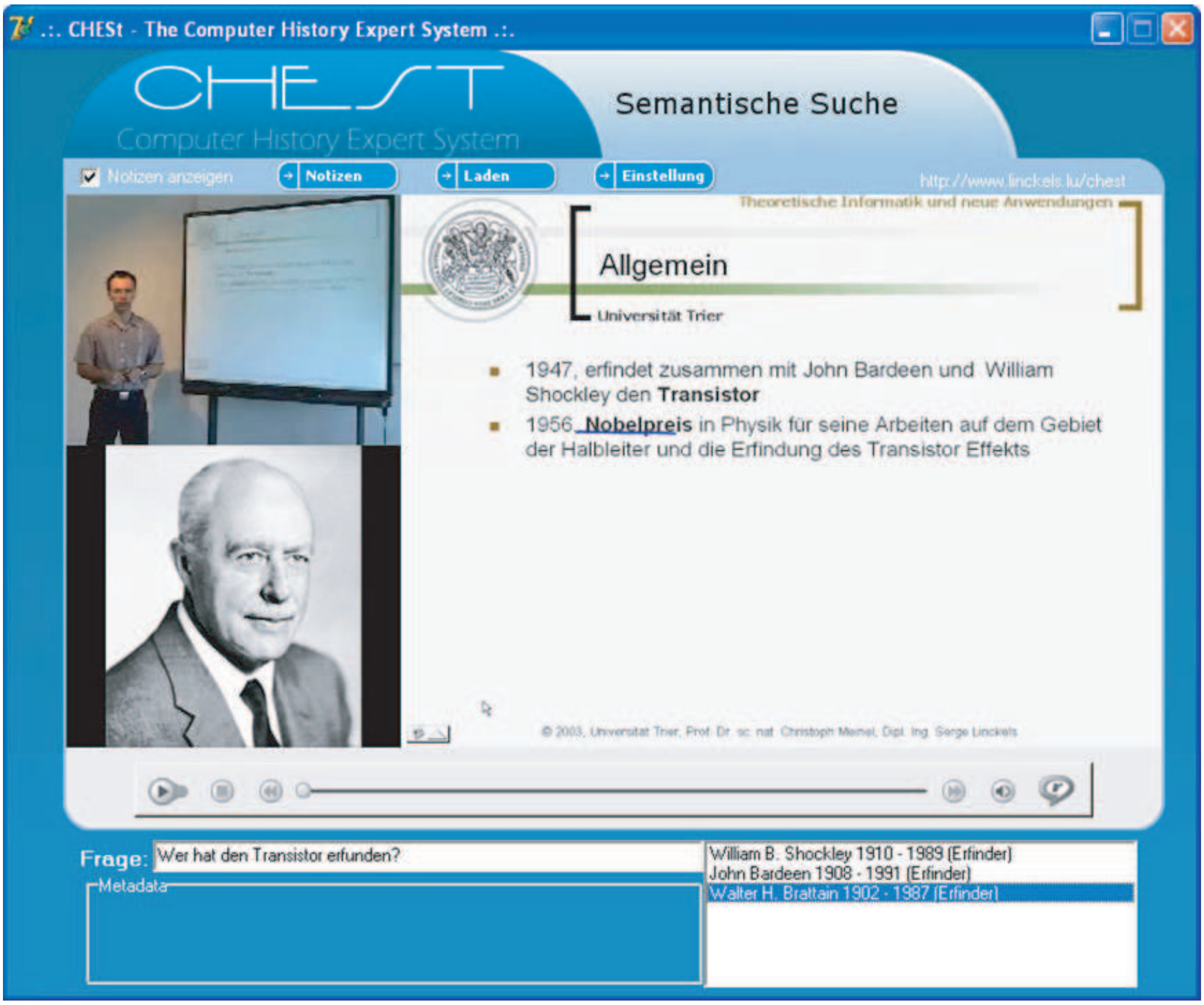
Splitting a large topic like computer history into a lot of small pieces is much easier than we assumed at the beginning. We are now convinced that most courses taught in schools or at universities can be divided into smaller atomic units where each covers one precise subject. Our assertion is based on three topics for which a knowledge base with small clips was produced; computer history, IT security, and fractions in mathematics (section 6.1). Another concrete test was made in biology where a teacher used our tool to explain the basic function of the heart. Furthermore, teachers of different fields confirmed that this concept could be used in their classes too, e.g. in a language course, a teacher could record one clip per grammatical rule.

One more advantage of that clip approach is the simplicity of administration. If the tool does not cover a certain topic, a new clip can be recorded and added to the knowledge base. The intervention of a computer-science expert is not necessary.

Basically, the clips are organized in three windows (figure 1). The first window (video and audio) shows a teacher explaining something on the whiteboard. This is the student's common view in a classroom and should create a kind of virtual classroom atmosphere. Based on practical teaching experience we can confirm that students often take lessons where they use a new computer tool or do research on the Web, as a kind of game, without relation to the normal lessons. The video sequence should keep them concentrated on what they do and draw their attention to what the teacher is explaining. The second window represents the usual blackboard. It is, in fact, a zoom on

[1] Instead of typing a question, we could also imagine that the user speaks the question into a microphone.

Figure 1: CHESt with a semantic search and the question: "Who invented the transistor?".



the whiteboard that the teacher uses in the video (first window). Although the blackboard is the medium used most frequently in schools, it has many disadvantages, such as the following:

- It is impossible to represent pictures.
- It is difficult and time-consuming for the teacher to create a complex drawing.
- It is time-consuming for students to reproduce its content in their books.
- The content is not available for later lessons and must be reproduced.

In our clips, we use an interactive *SmartBoard*, which offers a lot of advantages compared to an ordinary blackboard:

- The teacher can use this area for an on-screen presentation (e.g. PowerPoint).
- The teacher can add handwritten information to the SmartBoard, which is reproduced in this window both simultaneously and in exactly the same way.
- The teacher can also display the desktop of his/her connected laptop in order to explain a certain application, to show a Web site, or to demonstrate the settings of the computer, and so on.

The third window can be used for any purpose. It can contain links to a photo gallery, hyperlinks to additional information on the Web, book references or just a single picture of the subject about which the teacher is speaking. We used tele-TASK [SM02] (`http://www.tele-task.de`) to record the lessons in order to create one well-structured multimedia stream.

3 Ontology Aspect

Our librarian service is an expert system organized as domain ontology. In this section we give a general overview of ontology driven systems. Our contribution regarding this aspect of the project is the formalization of a background theory to organize a librarian service as domain ontology. We define and explain the principles of domain language, concept taxonomy, knowledge base, and formal representation terminology.

> *An expert system is a system that employs knowledge about its application domain and uses an inferencing (reasoning) procedure to solve problems that would otherwise require human competence or expertise.* Wikipedia

> *In information science, an ontology is the product of an attempt to formulate an exhaustive and rigorous conceptual schema about a domain. An ontology is typically a hierarchical data structure containing all the relevant entities and their relationships and rules within that domain. A domain ontology is an ontology tied to a specific domain. A foundation ontology is a form of ontology that tries to be less specific, and in that way more generally applicable.* Wikipedia

3.1 Ontology driven expert systems

Users rather accept the results of a computer tool if it is able to explain its reasoning [CEE+01]. In an educational environment, such tools — commonly known as expert systems — can be perceived as a virtual teacher with whom the student can maintain a dialogue, and it delivers answers based on logical reasoning rather than on sequential or recursive searches. General knowledge sources — also known as foundation ontologies — like Cyc (`http://www.cyc.com/` and `http://www.opencyc.org/`) or even the Web figured out to have the same major disadvantages for educational purpose. Firstly, they contain too much general information and often lack specific domain knowledge. Secondly, filter functions are only keyword-based and return too many and too inappropriated results. But the demands for education service are extremely varied and call for special knowledge about a certain topic. Such specific systems — often called domain ontologies — rely on a specialized and hierarchically organized knowledge base and on a specific reasoning engine. Especially in the field of intelligent information retrieval, such ontology-driven tools seem the most reliable.

Since the birth of the Semantic Web, and the related ontology technologies, a lot of research has started about ontology driven expert systems. The Institute of Computer Science form the FU-Berlin works on an ontology for Pathology (`http://www.inf.fu-berlin.de/inst/ag-nbi/research/swpatho/`). The university Blaise Pascal Clermont2 and the university Claude Bernard Lyon1 have recently published their work on an algorithm to automatically rank documents returned by a search engine in the Web following a query in NL, and based on ontology technologies [KBHS04]. The KIM [PKK+03] platform provides a novel Knowledge and Information Management (KIM) infrastructure and services for automatic semantic annotation, indexing, and retrieval of documents using RDF[2]. The prototype PRECISE [PEK03] uses ontology technologies to map semantically tractable NL questions to the corresponding SQL query. An ontology-driven semantic search is presented in [BCLB04] that allows to set up semantic level relevance feedback for query concept focalization, generalization, etc. [QYJ04] present a goal-oriented platform for a new model of distance education system. [AMO+03] report the results of an ontology-based query and answering system in chemistry called OntoNova. The system

[2]The *Resource Description Framework* (RDF) was introduced to build the Semantic Web, `http://www.w3.org/RDF/`.

is able to logically infer over the domain specific knowledge base, and to justify its answers in detail giving NL explanations. Other interesting projects that are more related to the retrieval aspect of our project are mentioned in section 5.1.

Conclusion There exists a variety of ontology driven expert systems, which aim to improve search engines by using new Semantic Web technologies and/or by mapping NL queries into other formal representations. However, as far as the authors know, no project focuses a librarian service adapted to an educational environment, that allows in addition to formulate questions in NL in order to return less but more pertinent multimedia answers.

3.2 Our contribution

Our librarian service is an ontology driven expert system about a given domain (e.g. computer history, IT security, fractions in mathematics), which is composed of a common language and a concept taxonomy. In this section, we will present the background theory for such a domain ontology. We define and explain the ideas of domain language, concept taxonomy, knowledge base, and formal representation terminology. Most of our work was published in [LM05c, LM05b]. The processing of a question in NL is covered in section 4.

3.2.1 Ontology language

In our librarian service, the user can freely formulate a question in NL. The librarian service masters a domain language, which may or may not contain all the possible words used by the user.

Definition 1 (Domain language)**.** *Let L be the set of all existing words over a certain alphabet that might be used to formulate a query, so that $L \subseteq \Sigma^*$. A domain language L_H is the set of all words that are known in the ontology H (we will call them the well-known-words, wkw, in the remaining part of the document) so that $L_H \subseteq L$.*

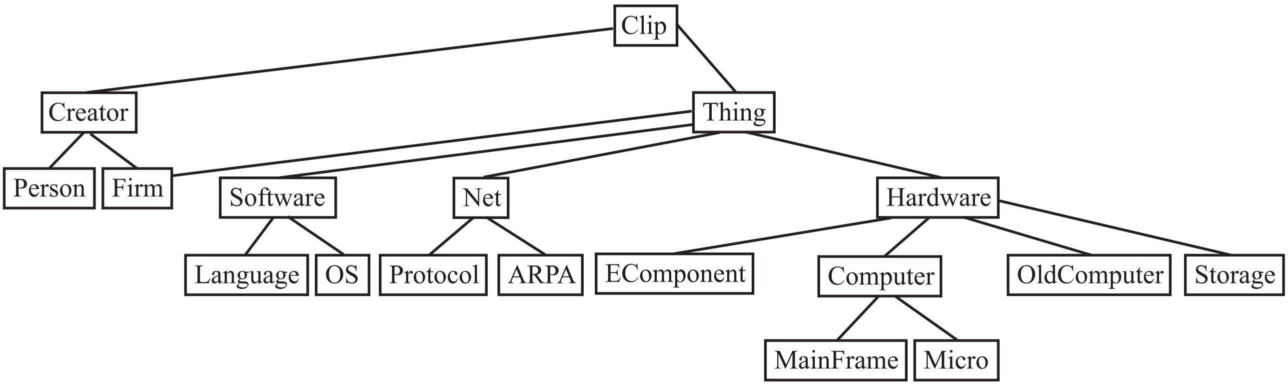
A domain language can contain verbs, nouns, articles as well as names, numbers, etc. Furthermore, the same domain language can contain words in different languages. The domain language gives no semantic or other description to its elements. It is just a set of stored words. The semantics are attached to each word by classification in a concept taxonomy (section 3.2.2). An ontology classically uses a domain dictionary as knowledge source, which is structured in a hierarchical way like,

- *Hyperonym*: a word with a more general meaning (e.g. animal is a hypernonym of cat)
- *Hyponym*: a word with a more specific meaning (e.g. cat is a hyponym of animal)
- *Synonym*: a word with identical meaning (e.g. car and automobile are synonyms)
- *Homonyms*: words with identical spelling but different meaning (e.g. Ada is a programming language but also a person)

In most of the related projects, an existing knowledge source was used as ontology dictionary, normally *WordNet* [Ma98] (`http://wordnet.princeton.edu/`). We investigated how useful *GermaNet* (the German version of WordNet) could be for our project. First of all, GermaNet is not dedicated to a domain. Like other large scale dictionaries, GermaNet lacks on the one hand 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[3]. Secondly, a GermaNet research license requires major financial investments.

[3]For example, WordNet returns three interpretations for the noun "Pascal": a unit of pressure, the name of a French mathematician, and the name of a programming language. Only the last interpretation would be interesting in the context of computer history.

Figure 2: Example of a concept taxonomy for CHESt.



We decided to create our own dictionary w.r.t. the ontology's concept taxonomy. In a first approach, all the words used for building the clips (section 2.2.3) were used to fill the ontology dictionary. We developed a tool (`http://www.linckels.lu/logiciels/ppt2txt.zip`) to convert the PowerPoint documents into pure text. In a later version, only the canonical form of the words and not all morphological forms were stored in the dictionary (section 5). Words that the user uses in his query and that are not found in the ontology dictionary are logged and can be added to the dictionary.

3.2.2 Concept Taxonomy

Ontologies are classically organized in a concept taxonomy, also called a *hierarchy of concepts* (HC) [BSZ03]. HC's are structures having the explicit purpose of organizing/classifying some kind of data such as documents. As an illustration, the concept taxonomy for our protoype CHESt (section 6.1) is shown in figure 2. Here, a document describing the transistor would be placed in the concept "EComponent" (electronic component), which is a hyponym of "Hardware". On the one hand, the more detailed the HC is, the more exact the system can classify the documents. On the other hand, a very detailed HC reduces the tolerance for the user question, so that it must be very well and precisely formulated.

Definition 2 (Concept taxonomy)**.** *A concept taxonomy $H = (V, E, v_0)$ is a rooted and oriented tree 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 $V = \{(s, T)|s \in S\}$ where s is a unique label (definition 3) and T is a set of wkw (definition 1), associated to a node, $T = \{t|t \in L_H\}$.*

A node represents a concept. The words that refer to this concept are regrouped in T. We assume that each set of words T_i is semantically related to the concept that the node v_i represents. Figure 3 illustrates this idea with the example of the concept "operating system" according to the CHESt taxonomy (figure 2). Here, words like "Windows" or "Linux" semantically refer to the same concept. Of course, a certain term, for example "Ada" could refer to different concepts; Ada is the name of a programming language but also the name of a person (Augusta Ada Lovelace). Not all words in L_H must be associated to a concept. Only words that are semantically relevant are classified. In general, nouns and verbs are best indicators of the sense of a question [Kup93]. 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.

Figure 3: Example of a node in the taxonomy.

Concept: **Operating System**
$s =$ chest:OS $T = \{$Windows, Linux, MS-DOS, VMS$\}$

Figure 4: Example of an OWL annotation.

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

Definition 3 (Label)**.** *A label is a unique identifier for a concept in a HC so that for a given label s one can find the corresponding concept and vice versa. S is the set of all existing labels in a HC.*

Technically, the kind of labels used depends on the encoding framework for annotating the documents in the knowledge base (section 3.2.3). In our case, a label is a namespace prefix (e.g. `chest`) and a local name (e.g. `EComponent`). Together they form the label of a node (e.g. `chest:EComponent`) like illustrated in figure 4.

Definition 4 (Classification)**.** *Be D the set of all documents in the knowledge base, then a document $d \in D$ is classified under a concept k, if d is about k and there is not a more specific concept k' under which d could be classified.*

In certain cases, a document can be classified in more than one concept. For example the document introducing the ARPA is classified in a concept named "Net" but also in a concept named "Firm".

3.2.3 Knowledge base annotation

An ontology has a well-structured knowledge base over which inference is possible. Therefor, the expert system is able to find implicit consequences of its explicitly represented knowledge. As consequence, the content of a clip is only of minor importance, but the meaning of the clip as a whole has to be machine readable. The meaning of each clip is described by additional data — called metadata — that are encoded using a specific ontology framework. We use the W3C recommendation *Web Ontology Language* (OWL) (`http://www.w3.org/2004/OWL/`) to describe all documents (*resources*) with metadata. An example is shown in figure 4. OWL exists in three different flavors, we use OWL DL.

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraint features. For example, while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1.
- *OWL DL* supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL DL was designed to support the existing Description Logics (section 3.2.4) business segment and has desirable computational properties for reasoning systems.
- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

Figure 5: Example of an $\mathcal{ALC}$-terminology for the TBox $\mathcal{T}$ w.r.t. to the concept taxonomy presented in figure 2.

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

3.2.4 Description Logics

Description Logics (DL) [BCM+03] are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally clear way, so that logical inference is possible. DL are a subset of first order predicate logic where the conceptual knowledge of an application domain is represented in terms of *concepts* (unary predicates) that are interpreted as sets of *individuals*, and *roles* (binary predicates) that are interpreted as binary relations between individuals. The semantics of a concept description is defined by the notion of interpretations as given in appendix A.

A typical DL knowledge base comprises two components: a terminology, also called a *TBox*, and assertions, also called an *ABox*. The TBox defines the vocabulary to use in the knowledge base by terms of concepts and roles. The concepts are either defined as new concepts or by using previously defined concepts. An example is presented in figure 5. In the ABox, one introduces individuals, by giving them names, and one establishes properties for these individuals. Figure 6 shows some examples of assertions translated into DL. A DL terminological knowledge base can easily be serialized as OWL.

We published in [LM05b] how DL can be used in our librarian service, where the concept taxonomy of the ontology is translated into an acyclic $\mathcal{ALC}$-concept description. The language $\mathcal{ALC}$ [SSS91] is sufficiently expressive for our purposes. It is in fact a subset of the logics implemented in most "state of the art" DL systems, for example those based on highly optimized *tableaux algorithms* like Racer [HM01], FaCT [Hor98], or Pellet [SP04]. $\mathcal{ALC}$ concepts are built using a set of concept names (NC) and role names (NR). Valid concepts 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 $A \in$ NC is a concept name and $R \in$ NR is a role name.

Reasoning in a DL knowledge base is mainly based on determining subsumption relationships and satisfiability w.r.t. the axioms in the TBox, and instance checking with respect to the assertions in the ABox (section A). The retrieval of documents from a DL knowledge base is described in section 5.

4 Natural language approach

We will start in this section with a short overview of natural language processing (NLP) in computer science. Then we describe the NLP implications in our project and our contribution. We present three different strategies that we explored to transform a user question from NL into a logical form. Then

Figure 6: Examples of concept assertions w.r.t. to the TBox defined in figure 5. The person Gary Kildall (1942-1994) founded the firm Digital Research in 1973, which has published the operating system CP/M in 1974.

Person(Kildall)	Firm(DR)	OS(CPM)
String("Gary Kildall")	String("Digital Research")	String("CP/M")
Date(1942)	Date(1973)	Date(1974)
Date(1994)	hasTitle(DR,"Digital Research")	hasTitle(CPM,"CP/M")
hasName(Kildall,"GaryKildall")	wasCreatedIn(DR,1973)	wasCreatedIn(CPM,1974)
wasBorn(Kildall,1942)	wasCreatedBy(DR,Kildall)	wasInventedBy(CPM,DR)
isDeceased(Kildall,1994)		

we detail the retained strategy and present our background theory for a semantic interpretation of a NL question.

> *Natural language processing is a subfield of artificial intelligence and linguistics. It studies the problems inherent in the processing and manipulation of natural language, and natural language understanding devoted to making computers "understand" statements written in human languages.*
>
> Wikipedia

4.1 Natural language processing in computer science

One goal of artificial intelligence (AI) work in NL is to enable communication between people and computers in a natural "human-like" way, e.g. by means of verbal communication. NL understanding is sometimes referred to as an AI-complete problem[4], because NL recognition seems to require extensive knowledge about the outside world and the ability to manipulate it. The definition of "understanding" is one of the major problems in NLP.

When we as humans process language, we are continually making guesses about meaning using our rich knowledge of the world and of the current culture to try and work out what is being communicated. For example, if asked "Is there water in the fridge?", most humans would consider this question as referring to a bottle of mineral water in the fridge. In fact, we use language with a rich knowledge of "normal circumstances". When using language in these situations, we do not need to state them explicitly. They are assumed to be known to all communicators. This unspoken context may comprise 90% of a communication, and allows the language which builds on this foundation to be very concise [Inm04]. But computers cannot hold even one reasonably rich view of context, culture or normal circumstances. For a computer, everything is an endless muddle of possibility with almost no way to sort out the "normal circumstance".

Today, the importance and the benefits of NL interfaces in computer science is not yet proved. This topic — especially concerning the Semantic Web and ontologies — is only poorly covered in literature. On the one hand, it seems that NL interfaces to applications have become more acute with new technologies from the Semantic Web and computational linguistics. Nontechnical people could access information through their web browsers, PDAs, cell phones, navigation systems, etc. in a very easy and user-friendly way. On the other hand, there are experts who are convinced that the main effort must be invested in the annotation and the representation of the knowledge rather than in the creation of NL interfaces.

[4]AI-complete is, by analogy to NP-completeness in complexity theory to indicate that the difficulty of a computational problem is equivalent to solving the central AI problem, i.e. making computers as intelligent as people.

Figure 7: Example of a head-lexicalized context-free grammar.

N	S, WMP, WPRO, VPA, VHFIN, ART, NP, NN
T	Wer, hat, den, Transistor, erfunden
R	S → VPA'
	VPA → WMP VPA', VHFIN VPA', NP VPA', VV
	WMP → WPRO'
	WPRO → Wer'
	VHFIN → hat'
	NP → ART NN'
	ART → den'
	NN → Transistor'
	VV → erfunden'
S	S

Regarding NL as a set of strings, a large part of language structures can be modelled using context-free descriptions. For that reason, *context-free grammars*[5] have become a significant means in the analysis of NL phenomena [All94, CEE$^+$01, CDM99]. But context-free grammars fail in providing structural and lexical preferences [Sch03]. Therefore, a probabilistic environment and a lexicalisation of the grammar framework are desirable extensions of the basic grammar type. An overview of probabilistic context-free grammars (PCFG) and head-lexicalised probabilistic context-free grammars (H-L PCFG) is given in appendix B. As an example, consider the H-L PCFG grammar in figure 7, with the heads of the rules marked by an apostrophe, and the respective parse tree in figure 8.

Conclusion NLP is an active research field in AI with the central aim of making computers understand humans. Although there seem to be more and more commercial needs, the real benefit of a NL interface to applications and databases has not yet been proven. H-L PCFG are a promising framework for processing NL.

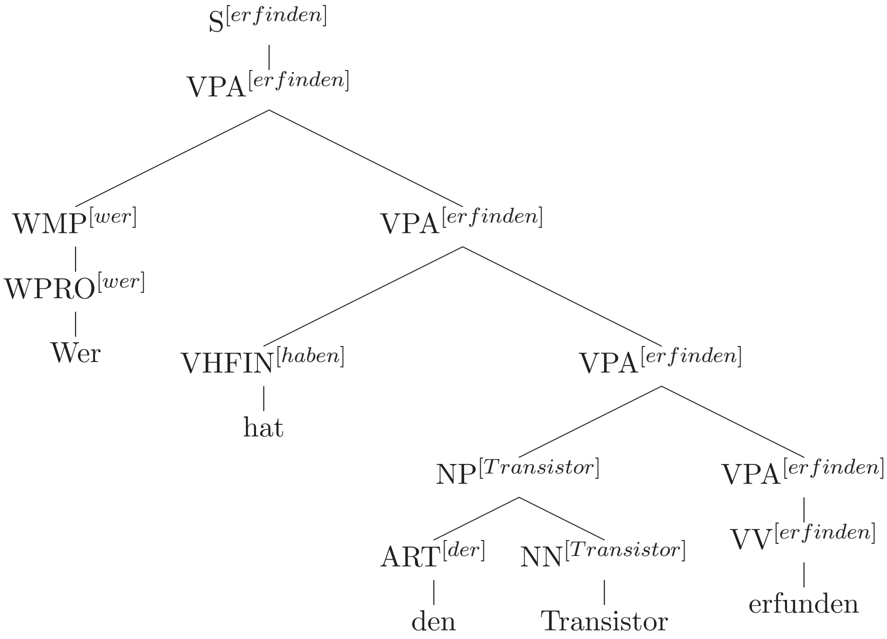
4.2 Our contribution

The objective of this project is to investigate in how far a query formulated in NL can improve the search of information in a knowledge base. Our motivation is twofold.

- Firstly, most people that are not search experts have difficulties or are note able to formulate their request in a machine optimized way, e.g. by combining search terms with Boolean operators. It is also possible that they do not use the right domain expressions. A NL interface would simplify the human-machine interaction, which is especially useful in an educational environment (section 2). Our librarian service allows the user to freely formulate a question in NL.
- Secondly, in order to create an "intelligent" search mechanism, the user must enter a query which contains enough semantics so that the librarian service understands the sense of the question in a non-ambiguous way, and to be able to logically infer over the knowledge base. In principle, a complete question in NL contains more semantics than just isolated or combined keywords. Our librarian service considers linguistic relations within the user question, and the given context from the domain ontology (section 3) to understand the sense of the sentence, and to translate it into a logical form.

[5]Context-free means that a certain sub-tree in a sentence analysis is analyzed in the same way no matter where in the sentence parse it is situated.

Figure 8: Example of a parse tree for the sentence "Wer hat den Transistor erfunden".



In this section, we present three different strategies that we explored to make our librarian service semantically understand NL. We detail the retained strategy and present the elaborated background theory for translating a NL user question into a computer readable and non-ambiguous form. We explain three wys how the similarity of two words can be quantified. Then we formulate the semantic interpretation step, which is the transformation of a NL sentence into a logical form. We conclude with an enumeration of various problems while processing NL sentences.

4.2.1 Explored strategies

The representation of context-independent meaning is called the *logical form*. The process of mapping a sentence to its logical form is called *semantic interpretation*, and the process of mapping the logical form to the final knowledge representation (KR) language is called *contextual interpretation* [All94]. Our strategy to process a NL question was improved with the evolution of the project.

Strategy 1 Each concept in the concept taxonomy refers to a number of semantically relevant words. The idea is to map each word in the user question to one or more concepts in the ontology. Semantically unimportant words are not mapped to a concept, thus will not be considered in the later retrieval. The logical form of the user question is then defined by the conjunction of the mapped concepts and their respective values in the original sentence. This strategy was implemented in CHESt v2 (section 6.1), published in [LM04a, LM05c, LM04b], and tested in experiments (section 7). It turned out that this straightforward solution returns reliable results for simple and precise questions, but not for more complex and general questions.

Strategy 2 The simple mapping of isolated words to concepts is improved by considering linguistic information. The idea is to read the *word categories* for each word from the user question, e.g. verb, noun, article. It is then possible to map word categories to taxonomy concepts, and to ignore semantically irrelevant word categories (e.g. article). The linguistic pre-processing is performed with a *part-of-speech (POS) tagger*. Most POS taggers are based on statistical observations and trained over large corpora. We tested some twenty taggers and retained *TreeTagger* (`http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`) for all further tests. This strategy was implemented in CHESt v3 (section 6.1). However, it turned out that a reliable semantic analysis of a NL sentence cannot be performed only be considering the word categories because there are too many ambiguous cases.

Strategy 3 Our recent strategy to semantically understand a NL question is to consider the *syntax* of a sentence and to read linguistic relations between the words in the sentence. This improves considerably the mapping algorithm. The syntactic structure of a sentence indicates the way that words in the sentence 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. Most syntactic representations of language is based on the notion of context-free grammars, which represent the sentence structure in terms of what phrases are subparts of other phrases. This information is often presented in a tree form (figure 8). The linguistic pre-processing is performed with a *parser*. A first idea was to build our own parser. However, a parser is a complex tool, which requires considerable skills in computational linguistics. Before investing precious resources in the development of a parser of our own, we preferred to investigate whether an existing parser could be used. There exist only few German parsers that could be used in our project. We retained *LoPar* (`http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html`), which comprises parsing with H-L PCFG. Strategy 3 was implemented in CHESt v4 (section 6.1). In the remaining part of this document, all explanations consider only strategy 3.

4.2.2 Word equivalence

Common in all 3 strategies described in section 4.2.1 is the task to compute the similarity of words from the user question with those in the concept taxonomy (section 3.2.2). In fact, the same word can appear in different morphological forms, e.g. "Who *invented* the transistor?", "Did Shockley *invent* the transistor?", etc.

Definition 5 (Word equivalence)**.** *The function $\pi(a, b)$ quantifies the similarity of two given words $(a, b \in L)$ using a logic W, so that a and b are said to be equivalent w.r.t. to a given tolerance ε, written $a \equiv b$, iff $\pi(a, b) \leq \varepsilon$.*

The choice of W depends on how expressive one wants to be in the approximation of the meaning of the concepts and on the complexity of the NLP techniques used to process words. We explored three solutions.

Solution 1 In a first and now abandoned attempt described in [LM04b], we represented each word from the ontology dictionary and from the user question as a tree where each node represents a character. It was then possible to compare two words node by node, and to compute the length of their equal trunk as well as their remaining unequal tail. However, this solution requires quite complex computations, and is not reliable to consider the morphological forms of the words. It was never used in any of the prototypes.

Solution 2 The above solution was replaced by a more efficient approach, the *Levenshtein function* also called *edit distance*. For two given string arguments a and b, the function computes the number of deletions, insertions, or substitutions required to transform a into b. The greater the Levenshtein distance, the more different the strings are. The main disadvantage of this solution is that each word must be stored with nearly all of its morphological forms in the dictionary. This solution was used in CHESt v2 (section 6.1).

Solution 3 The above solution was improved by considering the canonical form of the words. We used linguistic tools (e.g. tagger or parser), which return for each word (*token*) its canonical form (*lemma*). Therefore, we only need to store the lemmatized words in the ontology dictionary (section 1), which is also an important gain in space. For a given lemma from the user question, the equivalence

Figure 9: Example of a parsed sentence and the corresponding semantic interpretation.

$q =$ Wer hat den Transistor erfunden?

$\downarrow$ H-L PCFG parsing

Wer $^{[wer]}_{\text{WPRO.subst.Nom}}$ hat $^{[haben]}_{\text{VHFIN}}$ den $^{[der]}_{\text{ART.Akk}}$ Transistor $^{[Transistor]}_{\text{NN.Akk}}$ erfunden $^{[erfinden]}_{\text{VVpast}}$

$\downarrow$ Semantic interpretation

$$q^{\mathcal{I}} = Creator(x1) \wedge wasInventedBy(x2, x1) \wedge EComponent(x2) \wedge hasTitle(x2, "Transistor")$$

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, relative to the length of the lemma. This solution gives good results, even if the user makes spelling errors.

4.2.3 Semantic interpretation

The semantic interpretation in our librarian service is the translation of a NL question into a logical form using a given strategy (section 4.2.1). An illustration is shown in figure 9. We decided to use *Description Logics* (section 3.2.4) as knowledge representation language. 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 [Sch93]. Finally, in our librarian service DL are the common knowledge representation language for the knowledge base encoded in OWL DL (section 3.2.3) and the semantical interpreted user question. A DL terminology is composed of *concepts* (unary predicates), these are generally nouns, question words (*w-words*) and proper names, and *roles* (binary predicates), these are generally verbs, adjectives and adverbs.

A core part of the semantic interpretation in our librarian service is a mapping algorithm. This step — commonly called *non-standard inference* [Küs01] — maps each word from the user question to one or more ontology concepts, and resolves the arguments of each role according to the syntactic structure of the sentence.

Definition 6 (User question)**.** *A user question q is a sentence that the user formulates in NL, so that $q \subseteq L$ and which is composed of words, so that $q = \{w_1, ..., w_n\}$ with $n \geq 1$ and $w_k \in L$.*

Definition 7 (Mapping)**.** *The meaning of each word $w_k \in L$ is made explicit with the mapping function $\varphi : L \to V_H$ over an ontology dictionary $L_H \subseteq L \subseteq \Sigma^*$ and an $\mathcal{ALC}$ concept taxonomy tree $H = (V, E, v_0)$, so that $\varphi(w_k)$ returns a set of interpretations Φ defined as follows,*

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

The function $ft(v_i)$ returns the set of wkw (T_i) associated to the node v_i (definition 2), and $w_k \equiv x$ are two equivalent words (definition 5).

Definition 8 (Semantic relevance)**.** *A word w_k is semantically relevant if there exists 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 Φ contains several different interpretations, i.e. a word can be mapped to different concepts at once. 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.

Figure 10: Example of interpretations.

w	$\varphi(w) = \Phi$	Number of interpretations
the	{ }	0
Windows	$\{(v_\alpha,$ "Windows"$,\sigma_\alpha)\}$	1
Ada	$\{(v_\beta,$ "Ada"$,\sigma_\beta)\}$,$\{(v_\gamma,$ "Ada"$,\sigma_\gamma)\}$	2

Figure 11: Example of ambiguity resolution using focus.

Variable	w	$\varphi(w)$	$f(\varphi(w))$
$x1$	Who	$Creator(x1)$	$Creator(x1)$
role (r)	invented	$wasInventedBy$(`chest:Thing`,`chest:Creator`)	$wasInventedBy(x2, x1)$
$x2$	Ada	$Person(x2) \sqsubseteq Creator(x2)$	
		$Language(x2) \sqsubseteq Thing(x2)$	$Language(x2)$

Definition 9 (Focus)**.** *The focus of a set of interpretations Φ in the context of a given question q is made explicit by the function f which returns the best interpretation for a given word in the context of the complete question. The focus, written $f(\varphi(w_k)) = v'$, guarantees the following,*

1. $v' \in \varphi(w_k)$
2. $|f(\varphi(w_k))| = [0, 1]$
3. $\top \leq v' \leq \bot$*, iff* $f(\varphi(w_k)) \neq \emptyset$
4. $\pi(w_k, x \in ft(v')) \geq \pi(w_k, y \in ft(v_i \in \varphi(w_k)))$

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 be the programming language named "Ada", but it can also be the name of the person "Augusta Ada Lovelace", the person of confidence of Charles Babbage. Only by putting the word in the context of a given question, the right interpretation can be crystalized. For example, the context of the sentence "Who invented Ada?" reveals that here Ada is the programming language and not the person Ada.

Technically, the focus function uses the signature of the role(s) in the user question, with $r(a_1, a_2)$, where $r \in$ NR (section 3.2.4) and $a_1, a_2 \in S$ (definition 3). The signature of each role defines the kind of arguments that can be used, e.g. $wasInventedBy$(`chest:Thing`,`chest:Creator`). Figure 11 illustrates ambiguity resolution with the focus function for the question "Who invented Ada?". The combinations $wasInventedBy(x2, x1)$ and $wasInventedBy(-, x2)$ are possible. Cyclic (*in breeding*) combinations like $wasInventedBy(x2, x2)$ are not allowed, though theoretically possible. The solution $wasInventedBy(x2, x1)$ is the best and only one that satisfies all identified concepts in the user question.

Definition 10 (Semantic interpretation)**.** *Let q be the parsed user question, which is composed of linguistic clauses, written $q = \{q'_1, ..., q'_m\}$, with $m \geq 1$. The sematic interpretation of a parsed user question q is the translation of each linguistic clause into an $\mathcal{ALC}$ terminology w.r.t. a given ontology H written,*

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

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

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 is preferred over the "and" operator.

The syntactic structure of the sentence returned by the parser, as well as the focus function help to resolve the role arguments like illustrated in the above example,

$$\begin{aligned} q_1 &= \text{Who invented Microsoft?} \\ (q_1)^{\mathcal{I}} &= Creator(x1) \wedge hasInventor(\boldsymbol{x2, x1}) \wedge Firm(x2) \wedge hasTitle(x2, "Microsoft") \end{aligned}$$

$$\begin{aligned} q_2 &= \text{What did Microsoft invent?} \\ (q_2)^{\mathcal{I}} &= Invention(x1) \wedge hasInventor(\boldsymbol{x1, x2}) \wedge Firm(x2) \wedge hasTitle(x2, "Microsoft") \end{aligned}$$

4.2.4 Various problems

First names A general problem is the semantic interpretation of first names. Empirical data showed us, that first names are rarely semantically relevant (definition 8), but can lead to ambiguities. For example, the name "Bill" can be misinterpreted because it can be a popular American first name, a US-banknote, a statement of money, etc. We suggest considering all first names as semantically irrelevant.

Sub-sentences The problem of sub-sentences is one reason why strategy 2 (section 4.2.1) was abandoned. For example, "Did William Shokley invent the transistor and who founded IBM?" Here, it is not clear what the arguments for the roles are without analyzing the syntax of the sentence. Our solution is to introduce a linguistic pre-processing step and to translate the user question (q) into a linguistic clausal form (q'), where each sub-sentence is represented by a separate clause. The different clauses are then interpreted as explained in section 4.2.3. For example,

$$\begin{aligned} q &= \text{Did William Shokley invent the transistor and who founded IBM?} \\ q' &= \text{[Did William Shokley invent the transistor] [and] [who founded IBM] [?]} \\ q^{\mathcal{I}} &= [Person(x1) \wedge hasTitle(x1, "Shokley") \wedge EComponent(x2) \wedge hasTitle(x2, "Transistor") \\ &\quad \wedge wasCreatedBy(x2, x1)] \wedge \\ &\quad [Creator(x1) \wedge Firm(x2) \wedge hasTitle(x2, "IBM") \wedge wasCreatedBy(x2, x1)] \end{aligned}$$

Inter-clausal dependencies A problem which is not yet solved are inter-clausal dependencies. For example, "Is the person who invented the transistor also the person who built the computer?". This kind of complex sentence — even if very improbable that a student formulates such a question — cannot be resolved correctly in the current state of our librarian service.

Adjectives and negations Although our background theory is generally applicable for all kind of sentences, we have not yet implemented a correct interpretation for complex formulations using adjectives or negations like,

$$\begin{aligned} q &= \text{Who did not invent Microsoft?} \\ q^{\mathcal{I}} &= \neg[Creator(x1) \wedge hasInventor(x2, x1) \wedge Firm(x2) \wedge hasTitle(x2, "Microsoft")] \end{aligned}$$

$$\begin{aligned} q &= \text{Who invented the first processor?} \\ q^{\mathcal{I}} &= ? \end{aligned}$$

5 Information Retrieval Aspect

The essence of our librarian service is to retrieve pertinent information from a multimedia knowledge base by allowing the user to formulate complete questions in NL. In this section we start with an overview of modern information retrieval (IR) systems. Then we present our contribution that is the generation of a semantic query, and the measure of the quality of the retrieval.

> *Information retrieval is the art and science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within databases, whether relational stand alone databases or hypertext networked databases such as the Internet or intranets, for text, sound, images or data.* Wikipedia

5.1 Modern information retrieval

A multitude of classical models exist in IR to search for pertinent information in a knowledge source, e.g. Boolean Model, Vector Model, Probabilistic Model [BYRN99]. Most of these models assign weights to index terms in queries and in documents. These term weights are ultimately used to compute the *degree of similarity* between each document stored in the system and the user query.

In addition to the work cited in section 3.1, we mention some interesting and related projects here. In [BJNS93] the optimization of semantic queries in object-oriented databases is analyzed. The outcome is that reasoning techniques for concept languages developed in AI apply to this problem. A query optimizer that recognizes subset relationships between query and a view (a simpler query whose answer is stored) in polynomial time is proposed. An integrated system for knowledge representation, called $\mathcal{AL}$-log, based on DL and the deductive database language Datalog is presented in [DLNS98]. A technique that can be used to provide an expressive query language for DL based knowledge representation systems is presented in [HT00]. One of the main advantages of this approach is that, being based on a reduction to knowledge base satisfiability, it can easily be adapted to most existing (and future) DL implementations.

Conclusion Classical IR models mostly work by assigning weights to keywords. A lot of research is done in the recent years to improve such classical IR models. The cited projects use logical interference to improve queries.

5.2 Our contribution

In general, a classical IR model (e.g. Vector Model) cannot be used in our project for two major reasons. Firstly, in our knowledge base there are multimedia resources, which may not have a textual content. Secondly, even if the content, e.g. the textual part of the slides or the spoken words by the presenter, might be accessible in form of index terms, the Vector Model, as well as most other IR models mentioned above, would not be helpful. Retrieving resources based on the similarity of index terms is by definition in contradiction with a semantic retrieval model. A semantic search engine is based on logical inference over the sense of the query and the completeness of the knowledge base. The benefit to map queries expressed in NL to domain ontology terminologies is only poorly explored.

In our librarian service, the retrieval itself is a simple extension of the semantic interpretation of the user question (section 4.2.3). In fact, a given semantic interpretation is used to generate a semantic query, and to logically infer over the knowledge base. The quality of the retrieval can be measured by quantifying the semantic relatedness.

5.2.1 Semantic query generation

We will start with the assumption that all documents in the knowledge base $\mathcal{K}$ are represented by DL terminologies w.r.t. an ontology H (section 3.2.3). Furthermore, we assume that the user question q, expressed in NL, is translated into a DL terminology w.r.t. the same ontology H (section 4.2.3). The logical form of the user question is used with a classical DL reasoner like Racer [HM01], FaCT [Hor98], or Pellet [SP04], which infers over the non-empty ABox. The returned results are the models of the semantic interpretation of the user question. They are logical consequences of the inference rather than the result of keyword matchings.

Definition 11 (Semantic query)**.** *A semantic query over a knowledge base $\mathcal{K}$ w.r.t. a domain ontology H, and a query in an $\mathcal{ALC}$ terminology means that there must exist at least one model $\mathcal{I}$ of $\mathcal{K}$ such $(R_q)^{\mathcal{I}} \neq \emptyset$, written $\mathcal{K} \models R_q$.*

In other words, there must exist an individual α in $\mathcal{I}$ that is an element of $(R_q)^{\mathcal{I}}$, i.e. α is a pertinent resource from the knowledge base according to the user question. All documents from the knowledge base that satisfy the expression R_q are potential results. As illustration, consider the following example,

$$\begin{aligned} q &= \text{Who invented the operating system CP/M?} \\ \mathcal{K} \models R_q &= Creator(x1) \wedge hasInventor(x2, x1) \wedge OS(x2) \wedge hasTitle(x2, "CP/M") \end{aligned}$$

According to the TBox shown in figure 5 every inventor, i.e. a person and firm, is potentially a result to the question because of the hierarchical information: $(Person \sqcup Firm) \sqsubset Creator \sqsubset Clip$. According to the ABox shown in figure 6, the individual $Firm(DR)$ would be yielded by the inference engine as a logical result.

Technically, the semantically interpreted question is used to generate a RDQL [MSR02] query. Firstly, for a complete question, each semantic interpretation, that is each translated linguistic clause, is transformed into a 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 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 figure 12.

5.2.2 Semantic relatedness

The best possibility to measure the pertinence of the results yielded by the search engine — commonly known as *semantic relatedness* — is to quantify the difference between the yielded result(s) and the "perfect answer". At the current state of the project, an automatic ranking of the results is not yet operational. Several techniques have been explored, but none gives satisfactory results.

We used two straightforward solutions to measure the quality of the search results. Firstly, we manually analyzed the log files that contain the user questions and the yielded results. Secondly, we collected the feedback of users, who were asked in experiments (section 7) to give their opinion about how they are satisfied with the results yielded by the semantic search engine (compared to those yielded by a simple keyword search) concerning the number of results and the pertinence of the results.

Figure 12: Complete example for the generation of a semantic query from the user question "Who invented the transistor?".

$$q = \text{"Wer hat den Transistor erfunden?"}$$

$$\Big\downarrow \text{ H-L PCFG parsing}$$

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

$$\Big\downarrow \text{ Semantic interpretation}$$

$$q^H = Creator(x1) \wedge wasInventedBy(x2, x1) \wedge EComponent(x2) \wedge hasName(x2, \text{"}Transistor\text{"})$$

$$\Big\downarrow \text{ Semantic query generation}$$

$\mathcal{K} \models q^H \hat{=}$

```
SELECT  ?x1
WHERE   (?x2 rdf:type chest:EComponent)
        (?x2 chest:hasName ?x2hasName)
        (?x2 chest:wasInventedBy ?x1)

AND     (?x2hasName =~/Transistor/i)
USING   chest for <http://www.linckels.lu/chest/elements/1.1#>
        rdf for <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

6 Implementation

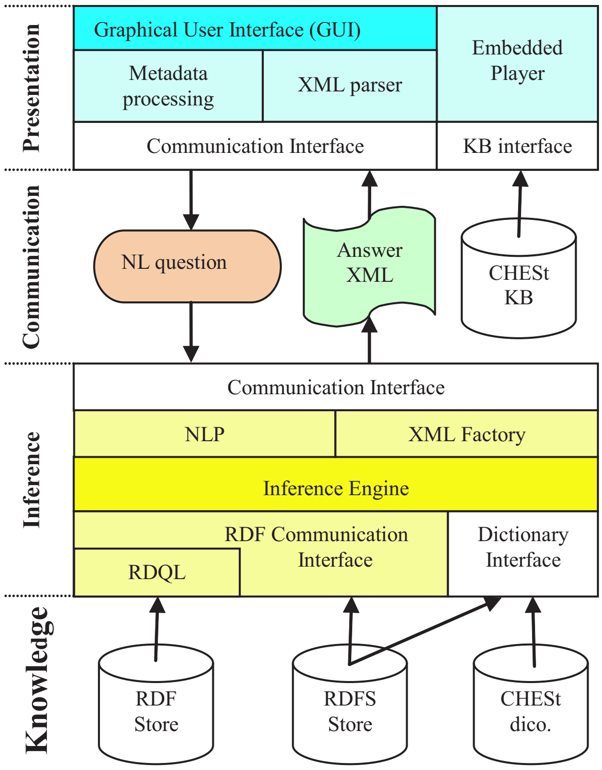
Our background theory for a librarian service was implemented prototypically in three different educational tools. In this section we will present the different prototypes, the semantic layer architecture that we elaborated, and some technical details about the development.

6.1 Prototypes

Computer History Expert System A first prototype of our librarian service is CHESt (*Computer History Expert System*), an e-learning tool where the user can freely formulate his question in NL (`http://www.linckels.lu/chest`). The prototype was firstly published in [LM04a]. CHESt understands the user's question and returns a precise and short answer in multimedia form. The tool has a knowledge base with 300 multimedia clips that cover the main events in computer history. CHESt exists in four different versions, these are,

Version	Description
CHESt v1	Simple keyword search
CHESt v2	Mapping of NL words to ontology concepts
CHESt v3	Using linguistic word categories to map words to ontology concepts, and to translate the NL question into a logical form
CHESt v4	Using a linguistic parser and considering the complete syntax of a NL question to translate it into a logical form

Figure 13: Semantic Layer Architecture.



Mathematic Expert System A second prototype of our librarian service is MatES (*Mathematic Expert System*), which explores a slightly different pedagogical approach. The knowledge base covers the topic of fractions taught in schools (7^{th} grade). All clips were recorded, mainly by pupils, w.r.t. the official school programme. The tool has a knowledge base with 115 multimedia clips that cover all questions about fractions for this level.

Information Technology Security Expert System A third prototype of our librarian service is ITExS (*IT Security Expert System*), which was developed for educational and professional usage. The resources in the knowledge base cover the official school programmes about that topic, so that ITExS can be tested in schools in different classes. Furthermore, ITExS disposes of enough knowledge so as to be used by professional IT people.

6.2 Semantic layer architecture

We developed a semantic layer architecture for our librarian service, which was published in [LM04c]. It is basically composed of 4 layers (figure 13),

- The *knowledge layer* that includes the knowledge base and the semantic repositories.
- The *inference layer* that includes the search engine.
- The *communication layer* that guarantees a transparent communication between the user and the search engine.
- The *presentation layer* that is responsible for the interaction with the user.

Except for the presentation layer, all other layers are platform-independent. Furthermore, each layer can be located somewhere on the Internet or on the local machine. The user question in NL is transmitted to the inference layer, which tries to find the best matching clip(s). The *Uniform Resource Identifiers* (URI) of the resulting clips are returned as an XML file to the presentation layer. Then, the user can select the clip(s) (s)he wants to watch.

The interaction between the inference layer and the presentation layer could be seen as distant procedure call. A beautiful solution would be to use the *Java Native Interface* (JNI) to execute some Java code inside the client application. Unfortunately, not all Java programs are compatible with any JNI. We must admit that we were not able to develop a working solution for generating the XML encoded answer via a JNI-call. Therefore, we opted for the more straightforward socket solution instead of JNI. Using sockets has three advantages.

- A socket communication is based on TCP/IP which is the most popular protocol at the moment and most people have it installed on their computer.
- The whole handshaking and error correction is assumed by the protocol stack. In fact, TCP/IP offers an error-free transmission. All details and used technologies (e.g. LAN adapter or analogous modem) are transparent for the user.
- All popular development environments offer components to easily implement a socket communication. This allows developers to create their own GUI that communicates with the inference engine.

This layer architecture guarantees a certain modularity of our librarian service. For example, no matter what linguistic tool (e.g. parser) will be used for the semantic pre-processing (section 4), it will not affect the rest of the system. The NLP module has a given interface. Changing some components, e.g. using another parser, only involves developing a piece of software, which transforms the parser's output into the required structure for the interface of the NLP module.

6.3 Development

The inference engine is developed in Java in order to guarantee a platform independency. We used the Jena API [CDD+04] and RDQL [MSR02] to access our knowledge repositories. Because Jena lacks of subsumption functionalities, we implemented our own algorithm (figure 14). The limited reasoning capacities in Jena are used in the current state of the project. Some tests were already made, to use an external reasoner, i.e. Racer [HM01].

Two solutions were explored for the client front-end. Firstly, a client application for Windows was developed. In fact, the client application must be operating system dependent because it embeds the native player for the multimedia clips. In our implementation we used Delphi. We managed to store the complete knowledge base (e.g. CHESt) with the client application on a single CD-ROM. No installation or configuration is necessary. The librarian service can be started immediately from the CD-ROM. Other configurations are possible, e.g. to copy the knowledge base on a server in a LAN and to access the server with the client application via the network. Secondly, we also created a Web interface that was developed in PHP. The Web server executes the inference engine on the server side.

7 Experiments

A lot of literature is available about experiments made with e-learning tools in schools, such as search engines. We refer to some of them in this section. We made experiments with CHESt v2 (section 6.1) at the Lyce Technique d'Esch/Alzette (`http://www.lte.lu`), a technical school in Luxembourg, at the beginning of the year 2005. Our main objectives were to investigate how useful CHESt is

Figure 14: Our subsumption function in Jena, which receives two labels (definition 3) as arguments, URI1 and URI2, and checks if URI1 $\sqsubseteq$ URI2.

```
boolean subsumes(String URI1,String URI2) {
   String TOPconcept = "http://www.w3.org/2002/07/owl#Thing";

   // Subsumption confirmed
   if (URI1.equals(URI2)) return true;

   // No subsumption found (arrived at top)
   else if (URI1.equals(TOPconcept)) return false;

   // Continue search
   else {
      // Get resource for current URI
      Resource r = model.getResource(URI1);

      // If this resource has no subclass (should not happen), then there can be no subsumption
      if (!r.hasProperty(RDFS.subClassOf)) return false;

      // The resource inherited from at least one subclass -> test all of them
      else {
         // Iterate through subclass' properties (normally just 1)
         boolean found = false;
         StmtIterator it = r.listProperties(RDFS.subClassOf);
         while ((it.hasNext()) && (!found)) {
            Statement st = (Statement)it.next();
            String p = st.getResource().toString();
            found = subsumes(p,URI2);
         } //while
         return found;
      } // hasProperty
   } // continue search
} // subsumes
```

as a complement to traditional courses, and how students accept to enter complete questions in the semantic search engine to reduce the number of results. The complete experiment was published in [LME05, RLME05].

7.1 Description of the experiments

Students from the upper secondary school level were asked to try out both versions of CHESt and to provide feedback on the three main characteristics of the tool: the number of results, the pertinence of the results and the satisfaction with the possibility to enter complete question(s) in natural language instead of keywords. Three consecutive assessment sessions took place, which differed from each other only in concern of two variables, with one variable being revised from session one to session two, and another variable from session two to session three (see figure 15).

7.2 General characteristics of the three sessions, instructions and procedure

For each of the three assessments, a different group of students was to try out both versions of CHESt. None of the subjects had further domain knowledge about computer history. One half of each group started with the keyword search, the other half with the semantic search. After 20 minutes, the students were asked their opinion about the just tested CHESt version on a number of questions, and then continued within a second trial — again lasting 20 minutes — with the respective other version of the search engine. In order to provide the subjects some general context within which they could search for information, three questions (in the following named "frame questions") were presented at

Figure 15: Settings of both variables for the three assessment sessions.

	Types of frame questions	Instructions given about how to use the search engines
Session 1	Precise questions, e.g. "Who invented the Z3-machine?"	Students were instructed to enter single or multiple words, the way they thought they would obtain the most pertinent results.
Session 2	General questions, e.g. "Describe the early years of the Internet.	Idem session 1
Session 3	Idem session 2	Students were told to enter questions while using the semantic search, and keywords while using the keyword search engine.

the beginning of each trial (i.e., six questions per task for each student; see further below for some examples of frame questions).

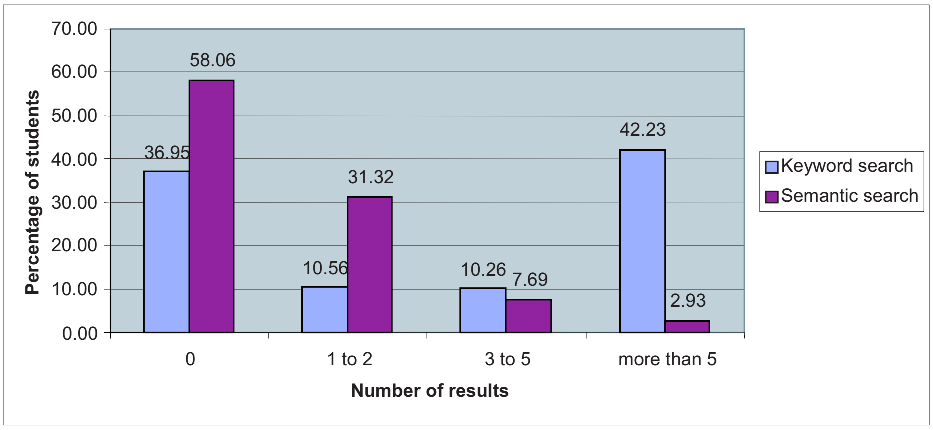
At the beginning of each session, the students were informed that two search engines allowing searching for information from the domain of computer history would be presented to them. They were told that not their successful answering to the frame questions would be the aim of the session, but rather their personal judging of the efficiency and their general liking of the respective search engine. They were also briefed that the graphical user interface (GUI) would be the same for both versions, and that no questions would have the GUI as target. The students were informed that their main job would consist in judging whether the search results yielded by the respective search engine would match their queries, and whether they really found the information they had been looking for. After the respective version of CHESt had been tested, the students answered questions focusing on the following issues:

- Did the just tested search engine yield too few, too many, or an adequate number of results? This question aimed at clarifying the personal judgment concerning the quantity of the results. Students might actually find what they searched for, but they might have expected more results.
- Did the search results exactly fit the queries? This question aimed at knowing whether, in general, the subjects had the impression that the result(s) listed was/were pertinent in regard to the keywords or questions they entered within the search field.
- Did the subjects find the information they have been searching for? This question is considered separately from others about the general fitting of the results, as the user might have found results that fitted the queries well, but still might have been unable to find what (s)he actually had been looking for.

After both versions of CHESt had been tested by the subjects, they were questioned on the following issues:

- Some of the questions asked for a comparison between both versions, aiming at finding out whether the participants had the impression that the one or other version would provide the more fitting results.
- The users were also asked which version they would choose if they had an exam within the domain of computer history during which they were allowed to use one of the CHESt versions. This was to find out about the general preference for one of the two versions within a concrete context.
- Finally, one question raised the issue about the students' opinion about having the possibility of asking complete questions instead of keywords.

Figure 16: Number of results per CHESt version: percentage of total number of results.



7.3 First Session

18 male students from the 13th (terminal) grade of secondary school (technical formation; mean age 21.25) participated within this first evaluative assessment. No information was provided about the difference between the two search engines; students were instructed to enter single or multiple words, even complete questions, just the way they thought they would obtain the most pertinent results. Further, the participants received precise frame questions such as the following ones:

- Did Howard Aiken and Charles Babbage know each other?
- Find three interesting inventions from the software domain (e.g. operating system, programming language, application). Which company or person has invented this and when was it published?

Asked about the number of the yielded results, the majority of the students think there is either an adequate number of results (seven students) or even too many results (seven students) generated by the keyword search. Meanwhile, considerable 14 out of 18 students asserted that the semantic search function yielded too few results. The real number of results generated by the respective search engines (figure 16) confirms that a higher percentage of queries within the semantic search than within the keyword search yielded no results. In the meantime, the keyword search led to more than five results in 42% of the search initials.

The question about the pertinence of the yielded results revealed an obvious superiority of the semantic search function. While 78of the subjects said that in only a few or in approximately half of the cases the keyword search would have provided fitting results, 78% considered that "most of the results" (61.1%) or "all of the results" (17%) fitted the search subjects within the semantic search engine.

There was not a similar obvious difference between the two search engines concerning the question whether students found what they were looking for. The subjects judged the efficiency of both search engines quite similarly (figure 17). Why is there this small discrepancy between the fitting of the results and the success in finding what was looked for? If taking a look at those last two questions in combination, one can observe that the incongruity is especially due to only a few of the students. Actually, for the keyword search, only one student meant that, although all of the results fitted the keywords, he only found what he looked for in approximately half of the search initials. Concerning the semantic search, three students (out of 16) said that most or all of the results fitted their search subjects, but that they still only rarely found what they had been looking for.

Figure 17: Within the listed results I found what I have been looking for.

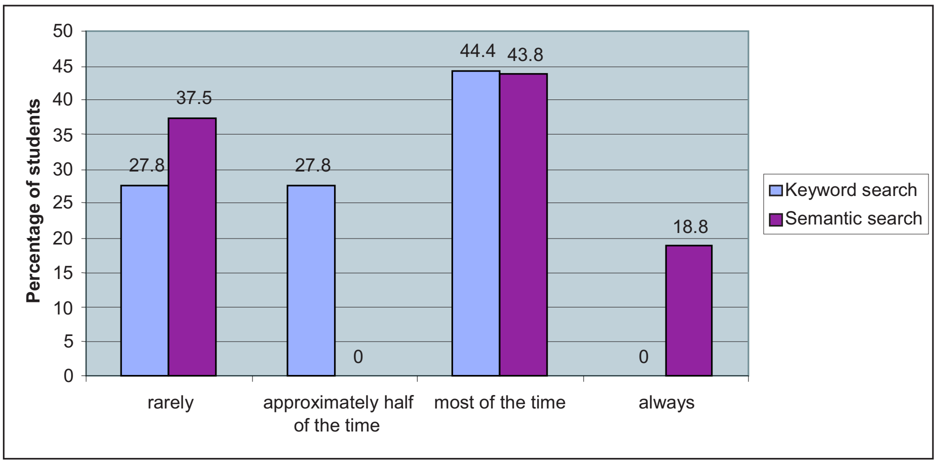


Figure 18: What version did the users prefer? Choice of the version 1=keyword search, 2=semantic search, 3=both versions equivalent, 4=none of the versions.

		Frequency	Percent	Cumulative Percent
Valid	1	5	27.8	27.8
	1 and 4	1	5.6	5.6
	2	7	38.9	38.9
	2 and 4	1	5.6	77.8
	3	2	11.1	88.9
	4	2	11.1	100.0
Total		18	100.0	

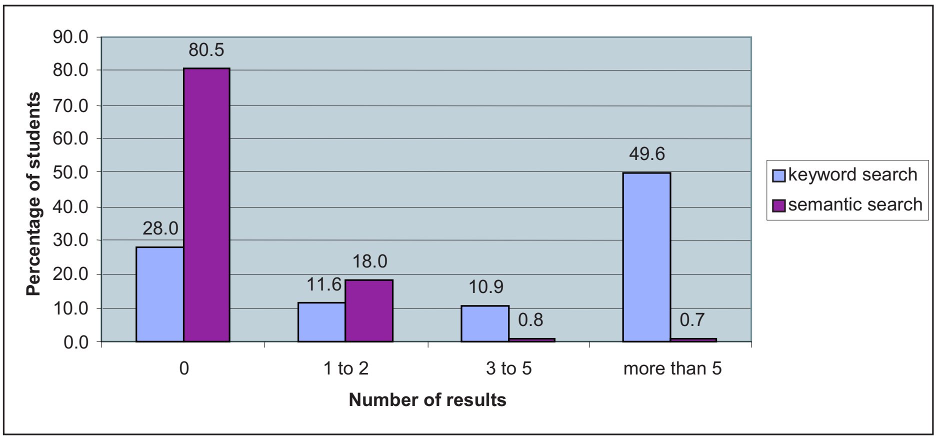
Asking students which one of the search engines they would prefer if they had an exam on the subject of computer history, the answers given were not clearly pointing in one or another direction (figure 18). Although most of the students were fairly satisfied with the fitting of the results from the semantic search engine, and although there was no greater difference between the keyword search and the semantic search in terms of finding what has been searched for, only 39% would choose the semantic search engine, compared to 28% announcing their preference for the keyword search, and 22% not deciding on any of both versions.

Finally, when asked about their liking of the possibility of entering whole questions instead of single keywords, half of the students (N=9; 50%) indicated that this possibility is only considered to be advantageous if it also yields better results than a keyword search.

To summarize, the first evaluation session revealed that, although most of the subjects were rather satisfied with the fitting of the results provided by the semantic search engine, they were not completely convinced of the (possible) advantages of the semantic version of CHESt.

Before discussing these results in greater detail, the realization and results of the two other evaluative sessions shall be described. The principal aim of the subsequent session was to replicate the results of the first session with more general frame questions. Indeed, the analysis of the keywords and sentences entered showed that most of the students were sticking all to strictly to the respective frame questions in their formulation of the questions and keywords in the question bar. In order to

Figure 19: Number of results per CHESt version: percentage of total number of results.



investigate whether similar results would be obtained when the students are given greater liberty in their searching for information on the subject of computer history, more general tasks were formulated for the second and the third evaluation session, described in the following sections.

7.4 Second Session

18 students (17 male) from the 12th grade of secondary school (technical formation; mean age 19.76 years) participated within this second evaluative session. This time, the frame questions were more general than in the previous session; examples of frame questions are as follows:

- Give an overview of the last 60 years of computer evolution.
- Explain why, especially around World War II, computers had been developed. Name three examples of such computers and their respective inventor(s).

In correspondence to the results from the first assessment, the students in this second session showed to be more satisfied with the number of the results listed by the keyword search engine. 56asserted that an adequate number of results were listed by the keyword search engine, only 17% said it had been too few. This can be contrasted with respectable 78% of the students thinking the semantic search had generated too few results. The mean number of results yielded by the semantic search engine is way below the one in the first session, with no results yielded at all in considerable 80.5% of the search initials (figure 19). The keyword search in the meantime yielded no results in only 28% of the searches, while half of the search initials (50%) led to more than five results.

This time the question about the pertinence of the yielded results revealed a different answer pattern than within the first testing session. Actually, the keyword search was judged a little bit more positive than in the previous testing, while the semantic search engine was given a worse evaluation. This led to the results pattern of a nearly equivalent number of students (nine and 10, for the keyword and semantic search, respectively) asserting that "most of the results" or "all of the results" fitted the search keywords. Two students asserted not having had any fitting results at all during the semantic search (while none of the users did so concerning the keyword search).

As far as the question of whether the students were able to find what they were looking for is concerned, a rather negative picture is being presented for the semantic search. Three people never found what they were looking for, and four users were only rarely able to locate some interesting

Figure 20: What version did the users prefer? Choice of the version 1=keyword search, 2=semantic search, 3=both versions equivalent, 4=none of the versions (more answers permitted).

		Frequency	Percent	Cumulative Percent
Valid	1	17	94.4	94.4
	2 and 4	1	5.6	100.0
Total		18	100.0	

information. This is the case "only" two times within the keyword search. Clearly most of the students (76%) confirmed being successful in finding the information they had been looking for with the keyword search, compared to 47% concerning the semantic search engine.

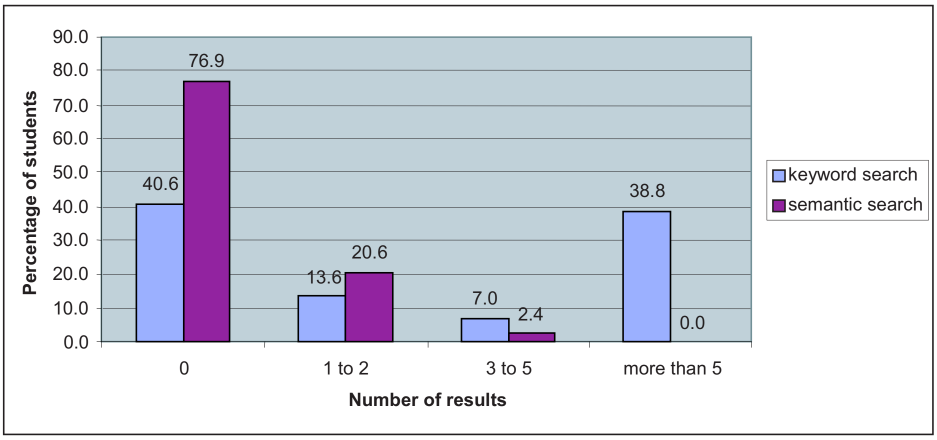
If comparing and combining the answers to these last two questions (fitting of the results and finding what one has been looking for), we noticed that only for very few of the students there was an incongruity between the judgment on the general fitting of the search results to the entered queries on the one hand, and the judgment about whether the searched information was found on the other hand. For the keyword search one student (out of 17) believed that, although most of the results fitted the keywords, (s)he rarely found what (s)he had been looking for. Two students asserted that, although only few of the keywords fitted the results, they still found what they had looked for in most of the search initials. In concern of the semantic search, a similar pattern of the results is obtained, with two students saying that most or all of the results fitted their search words, although they only rarely found what they had looked for. One user indicated that although only few of the keywords fitted the results, (s)he still found what (s)he had looked for in most of the search initials.

Even though the previous questions already revealed that users did not have such an obvious judgment in the one or other direction about one of the CHESt versions any more, asking them which version they would prefer to use during an exam in the domain of computer history revealed an obvious preference of nearly all of the students (17 out of 18) for the keyword search (figure 20). Approximately half of the students (10 out of 18) finally stated that the possibility to enter whole questions instead of single keywords would be a good option only if this would lead to better results than with keywords.

In summary, the second evaluative session - using more general frame questions and thus providing the participants a greater liberty in their information search - revealed a slightly more negative judgment of the semantic search engine than the first session. Firstly, the students indicated being more satisfied with the number of results provided by the keyword search than those by the semantic search. Secondly, although the judgment about the fitting of the listed results was comparable for both CHESt versions, more users pointed out finding what they were searching for within the keyword search rather than with the semantic search. The clear preference for the keyword search finally underlines that the characteristics of the keyword version of CHESt are the more appealing ones.

One final point still remains to be pointed out in this context: actually, the analysis of the log-file of this session reveals that students did not refer very often to the possibility of combining keywords and/or even entering whole questions, which would ensure optimal "communication" with the semantic version of CHESt. We hypothesized that the high number of semantic search initials yielding no results at all is due to this problem. In order to investigate whether the rather negative results concerning the semantic version of CHESt are the result of whether users do or do not fully exploit the potential of CHESt, a third evaluative session took place in which the explicit instruction was given to enter complete questions in the search field when using the semantic version of CHESt.

Figure 21: Number of results per CHESt version: percentage of total number of results.



7.5 Third Session

14 students (11 male) from 12th grade of secondary school (general technical education; 17 to 20 years old) participated in this third assessment. While the frame questions remained rather general (as within the second session), the students were, this time, explicitly told to enter complete questions while using the semantic search, and to enter single or multiple words while using the keyword search. The judgments given by the students during this third session were still comparable to the ones provided in the second assessment, as outlined in the following section.

Comparable to the results from the previous assessment, the students were more satisfied with the number of results listed by the keyword search than with the number generated by the semantic search. While 64% asserted that an adequate number of results were listed by the keyword search engine, only 21% said so with relation to the semantic search engine. 79% had the opinion that the semantic search listed too few. The analysis of the mean number of results yielded by the respective queries further consolidates the finding out of a very elevated percentage of semantic search initials not having generated any results at all (77%) during the previous sessions (figure 21).

The overall satisfaction with the fitting of the results listed by the semantic search engine was even lower than in the previous assessments. While 57.1% affirmed that most of the results fitted the words they entered when using the keyword search, only 33% (25% saying "most of them", 8% saying "all") said so as to the results yielded by the semantic search. The same is true for the success the users experienced concerning the finding of what they were looking for: with the keyword search, eight people (out of 13) asserted having found what they were looking for "most of the time" or "always", but only two people were equally confident concerning of the semantic search engine. No greater discrepancies were found as to these two questions, with only one student asserting that, although only few of the results fitted the queries, (s)he still found what (s)he had looked for most of the time (both CHESt versions).

Asking students which version they would prefer to use during an exam in the domain of computer history revealed an obvious voting of the great majority of the students (11 out of 14) for the keyword search. Finally, 79% of the users emphasized that they would like the option of being able to ask complete questions instead of keywords only if this yielded the better results.

In summary, this last assessment session left us with a rather devastating image of the semantic search engine. Firstly, students were again not as satisfied with the number of results provided by the semantic search as those listed by the keyword search. Also, even when giving concrete instructions as to how to use the search engines in order to guarantee that all qualities of the semantic engine are exploited, a very high percentage of the queries still remain without any yield of results. In addition, even few students than in the previous sessions really seemed to be satisfied by the pertinence of the listed results. The same is true for the judgment of the success in finding what the users were looking for. This third assessment session thus confirms and consolidates the finding that several of the qualities of CHESt have to be revised in greater detail.

7.6 Discussion and conclusion

The present investigation aimed at evaluating the qualities of the keyword and the semantic version of the e-learning tool CHESt in an educational environment. Students from the upper secondary school level were asked to test both versions and judge them on the number and the pertinence of the yielded results, as well as to give their opinion about the possibility to enter questions instead of keywords. The results from the three evaluative sessions especially revealed two things in particular:

The first striking result was that the subjects generally preferred the keyword search to the semantic search. This was found to be independent from the judgments on the appropriateness of the yielded results. Even in the first session, where the majority of the users claimed that the semantic version yielded pertinent results, compared to only a few concerning the keyword search, most of the users decided to use the keyword version of CHESt within an exam on computer history. There are multiple reasons that could stand as explanation for this result. Actually, the students could just be more accustomed to use keyword search engines (nearly all search engines on the Web are keyword based). Further, entering keywords might have been experienced as an easier and more comfortable task than entering whole sentences. Blondel concludes in his survey [Blo01] that the students' preferred method to query search engines is without any doubt by keywords. In the light of the generally valid claim of the users that they like the option of being able to ask complete questions instead of keywords, but only if this would yield better results, we however realized that the supplementary intellectual task of thinking and formulating whole questions must not necessarily be considered as a real burden compared to the entering of single, maybe only loosely related keywords. Finally, the number of results generated by the respective search engine is considered to be a factor of central importance concerning the rating of the CHESt-versions. Actually it turned out that, more than half of the semantic search initials did not lead to any result at all. By contrast to this, the keyword search generally listed a high number of results (judged as being "too many", or - more often - as being "neither too many, nor too few"). Seeing the differences within the number of results that were in the mean yielded by a keyword and a semantic search respectively, the users might have experienced to have many more opportunities to explore the content of the knowledge base with the keyword than with the semantic search. The number of the yielded results is seen as a characteristic that might be of central importance, especially if users of an e-learning tool have not yet an elaborate knowledge within the domain of interest (such as computer history). The fact that during the last two sessions users more often found what they had searched for with the keyword than with the semantic search engine, further underlines the advantageous possibilities the students might have experienced with the higher number of results produced by the keyword search.

One second striking finding concerns the pertinence of the results. During the first evaluative session, the subjects were more confident in this concern while using the semantic search function. As opposed to this, subjects judged the keyword search more positively in this regard during the other two sessions, where more general frame questions had been given. Thus, providing the users with additional freedom to explore the knowledge base has led to search queries that provided less convenient results than queries that were initiated within a more restrictive context. We refer to three explanations concerning this pattern of results: First of all, this finding might be explained by the fact that users directly associated this question with the one about the number of the results: as a very high percentage of the search initials didn't yield any results at all during the semantic information queries, the judgment on the pertinence of the results might have been strongly influenced by this (with an interpretation such as "no results at all means no fitting"). Secondly, the difference in this matter between the first and the other two sessions might suggest that students have had general difficulties to formulate own questions in order to explore a domain such as computer history. This might have resulted in a general sticking to the rather specific frame questions during the first session. Indeed, it was possible to take exact questions/words that were given as frame questions (or to slightly change the formulation of those) and to enter these within the search field. Such a strategy was not possible any more to the same degree during the two subsequent sessions, where users were expected to think about and formulate questions more autonomously. Thirdly, this finding of a difference between the first and the other two sessions however might also suggest that we have to improve the semantic search engine concerning its main characteristic: the understanding of the users' questions. As already outlined above, although many of the yielded results were judged to fit the search subjects, just too many queries have yielded no results at all. Given the fact that both search engines access the same knowledge base, and that the judgment about the pertinence of the results did not differ significantly between sessions two and three (where emphasis was placed on the instruction to enter whole questions) this finding underlines that future efforts will need to focus on the improvement of the inference engine. It seems that the semantic search engine is weak in inferring over more general questions. Current research within our laboratory aims to improve the linguistic pre-processing of the user's question in order to extract more semantic information.

In conclusion, the results of our assessments suggest that the satisfaction users experience with a search engine like CHESt (in its two versions) seems to strongly depend on three factors:

- The practice users have with the respective search engine (with the formulation of questions). We agree with [FDD^{+}99, Blo01, NPSR99] that users need guidance in how to formulate effective queries, especially if they are not expert in the focused domain
- The background knowledge users have concerning of the focused domain (here: computer history); little knowledge within a domain of interest seems to require good basic opportunities (such as a list of domains to be explored or a search tree) to explore a given knowledge base;
- The factual success of a search engine to find the requested content - which, again, might depend on the content of the knowledge base.

8 Conclusion and future work

We conclude in this section by discussing some (dis)advantages of our project. We also present some open questions that can be explored in the future.

8.1 Conclusion

Experiments (section 7) confirmed that our background theory for a librarian service (sections 3, 4, and 5) can be implemented in an educational tool. Such a tool can be used as a complement to traditional courses. The presentation of the knowledge in the form of short multimedia clips, and the fast response time of the search engine were strongly appreciated by the students.

But we learned that our semantic search mechanism must be improved in several ways. Firstly, a more profound linguistic pre-processing can improve the semantic interpretation (section 4.2.3), e.g. correctly translating adjectives. Thus, the inference engine will be able to return still more pertinent results and filter out some "noise". Secondly, all queries are built "bottom up", without consideration of neither former queries, nor queries from other users. A collaborative information retrieval approach like the one proposed by [Kli04], or the study by [WCK05], seem interesting solutions to explore. Finally, the human-machine interaction could be improved, for example by allowing users to formulate graphical queries instead of NL questions.

We understood that it is not an easy task to use search engines as a didactical tool in schools. Firstly, in a free discussion with the students, the problem was often mentioned that the topic (e.g. computer history) was too complicated. Users need training and domain knowledge before they are able to successfully use search engines for that topic. Secondly, when using search engines, the students are relatively free to act as they like, which is quite unusual for most. As confirmed by [Mar03, FDD+99, NPSR99, Bon01], users need guidance in how to formulate effective queries even if they are free to formulate their question in NL. Finally, we agree with [HS00, Blo01] that specific search engines must be developed for specific purposes: usage in school, specific domains, specific user groups, etc.

8.2 Open questions

An interesting but maybe more technical subject is the question, if our librarian service is able to work with a larger ontology. In principle, the larger the ontology, the more ambiguities are possible and the more metadata are necessary to describe the clips. An open question is if our semantic search engine could once be used in a kind of "world ontology" like the Web.

A more pertinent question is how a given ontology could be enlarged, i.e. how can new knowledge be added in an automatic way. We mainly think about a learning process where the search engine must understand if the answer to a question is unknown. Then, there must be a mechanism to enrich the ontology, maybe by the feedback from the user or the administrator. In this way, the ontology would grow in a natural and consistent way.

For the moment, all inference over the user's question is done bottom up without considering any previous queries neither from the same nor from other users. We shall investigate how the system can learn from former questions (from the same or from other users). This would open complete new inference possibilities, and would also foster a collaborative approach. Questions from other users could be used in the reasoning process, and users could group their searching efforts together.

References

[All94] James Allen. *Natural Language Understanding.* Addison Wesley, 1994.

[AMO+03] Jürgen Angele, Eddie Mönch, Henrik Oppermann, Steffen Staab, and Dirk Wenke. Ontology-based query and answering in chemistry: Ontonova @ project halo. In Fensel et al. [FSM03], pages 913–928.

[BCLB04] Dario Bonino, Fulvio Corno, Farinetti Laura, and Alessio Bosca. Ontology driven semantic search. *WSEAS Transaction on Information Science and Application*, 1(6):1597–1605, Dec 2004.

[BCM+03] 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.

[BJNS93] Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, and Martin Staudt. Subsumption between queries to object-oriented databases. Technical Report RR-93-44, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Kaiserslautern, Germany, 1993.

[Blo01] 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*, 2001.

[Bon01] Curtis Bonk. *Online Teaching in an Online World*. CourseShare.com, `http://www.publicationshare.com/`, 2001.

[BSZ03] Paolo Bouquet, Luciano Serafini, and Stefano Zanobini. Semantic coordination: A new approach and an application. In Fensel et al. [FSM03], pages 130–145.

[BYRN99] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[CDD+04] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *WWW (Alternate Track Papers & Posters)*, pages 74–83. ACM, 2004.

[CDM99] Hinrich Schtze Christopher D. Manning. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[CEE+01] Kai-Uwe Carstenense, Christian Ebert, Cornelia Endriss, Susanne Jekat, Ralf Klabunde, and Hagen Langer, editors. *Computerlinguistik und Sprachentechnologie: Eine Einfhrung*. Spektrum Akademsicher Verlag, Heidelberg-Berlin, 2001.

[DLNS98] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Al-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.

[FDD+99] 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.

[FSM03] Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors. *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 of *Lecture Notes in Computer Science*. Springer, 2003.

[HM01] Volker Haarslev and Ralf Möller. Racer system description. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Automated Reasoning, First International Joint Conference, IJCAR 2001, Siena, Italy, June 18-23, 2001, Proceedings*, volume 2083 of *Lecture Notes in Computer Science*, pages 701–706. Springer, 2001.

[Hor98] Ian Horrocks. The FaCT system. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 307–312. Springer-Verlag, May 1998.

[HS00] Christoph Hölscher and Gerhard Strube. Web search behavior of internet experts and newbies. *Computer Networks*, 33(1-6):337–346, 2000.

[HT00] Ian Horrocks and Sergio Tessaris. A conjunctive query language for description logic aboxes. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, pages 399–404. AAAI Press / The MIT Press, 2000.

[Inm04] Dave Inman. *The possibility of Natural Language Processing by computer*. `http://www.scism.sbu.ac.uk/inmandw/tutorials/nlp/intro/intro.html`, 2004.

[KBHS04] Naouel Karam, Salima Benbernou, Mohand-Said Hacid, and Michel Schneider. Semantic matching of natural language web queries. In Koch et al. [KFW04], pages 416–429.

[KFW04] Nora Koch, Piero Fraternali, and Martin Wirsing, editors. *Web Engineering - 4th International Conference, ICWE 2004, Munich, Germany, July 26-30, 2004, Proceedings*, volume 3140 of *Lecture Notes in Computer Science*. Springer, 2004.

[Kli04] Stefan Klink. Improving document transformation techniques with collaborative learned term-based concepts. In Andreas Dengel, Markus Junker, and Anette Weisbecker, editors, *Reading and Learning*, volume 2956 of *Lecture Notes in Computer Science*, pages 281–305. Springer, 2004.

[Kup93] Julian Kupiec. Murax: A robust linguistic approach for question answering using an online encyclopedia. In Robert Korfhage, Edie M. Rasmussen, and Peter Willett, editors, *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 181–190. ACM, 1993.

[Küs01] Ralf Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001. Ph.D. thesis.

[LM04a] Serge Linckels and Christoph Meinel. An application of semantics for an educational tool. In Nuno Guimarães and Pedro Isaías, editors, *Proceedings of the IADIS International Conference Applied Computing 2004, Lisbon, Algarve, Portugal, March 23-26, 2004*, pages II–234 – II–239. IADIS, 2004.

[LM04b] Serge Linckels and Christoph Meinel. Automatic interpretation of natural language for a multimedia e-learning tool. In Koch et al. [KFW04], pages 435–439.

[LM04c] Serge Linckels and Christoph Meinel. Semantic layer architecture for an educational expert system in computer history. In *International Conference on Advances in Intelligent Systems (AISTA) - Theory and Applications, Luxembourg-Kirchberg, Luxembourg, November 15-18, 2004*, 2004.

[LM05a] Serge Linckels and Christoph Meinel. An educational tool that understands students' questions. *Journal of Distance Learning*, 2(3):13–21, 2005.

[LM05b] Serge Linckels and Christoph Meinel. A simple application of description logics for a semantic search engine. In Nuno Guimarães and Pedro Isaías, editors, *Proceedings of the IADIS International Conference Applied Computing 2005, Lisbon, Algarve, Portugal, February 22-25, 2005*, volume 2, pages 306–311. IADIS, 2005.

[LM05c] Serge Linckels and Christoph Meinel. A simple solution for an intelligent librarian system. In Nuno Guimarães and Pedro Isaías, editors, *Proceedings of the IADIS International Conference Applied Computing 2005, Lisbon, Algarve, Portugal, February 22-25, 2005*, volume 1, pages 495–503. IADIS, 2005.

[LME05] Serge Linckels, Christoph Meinel, and Thomas Engel. Teaching in the cyber-age: Technologies, experiments, and realizations. In *DeLFI 2005: 3. Deutsche e-Learning Fachtagung Informatik, der Gesellschaft für Informatik e.V. (GI) 13.-16. September 2005 in Rostock*, pages 225–236, 2005.

[Ma98] George A. Miller and al. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge - London, 1998.

[Mar03] Philippe Martin. *Web Intelligence*, chapter Knowledge Representation, Sharing and Retrieval on the Web, pages 263–297. Springer-Verlag, Jan 2003.

[MG90] Richard E. Mayer and Joan K. Gallini. When is an illustration worth ten thousand words? *Journal of Educational Psychology*, 82(4):715–726, Dec 1990.

[MS94] Richard E. Mayer and Valerie K. Sims. For whom is a picture worth a thousand words? extensions of a dual-coding theory of multimedia learning. *Journal of Educational Psychology*, 86(3):389–401, Sep 1994.

[MSR02] Libby Miller, Andy Seaborne, and Alberto Reggiori. Three implementations of squishql, a simple rdf query language. In Ian Horrocks and James A. Hendler, editors, *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 423–435. Springer, 2002.

[NPSR99] Raquel Navarro-Prieto, Mike Scaife, and Yvonne Rogers. Cognitive strategies in web searching. In *5th Conference on Human Factors & the Web*, `http://zing.ncsl.nist.gov/hfweb`, 1999.

[Ows97] Ronald D. Owston. The world wide web: A technology to enhance teaching and learning? *Educational Researcher*, 26(2):27–33, march 1997.

[PEK03] Ana-Maria Popescu, Oren Etzioni, and Henry A. Kautz. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces, January 12-15, 2003, Miami, FL, USA*, pages 149–157. ACM, 2003.

[PKK+03] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. Kim - semantic annotation platform. In Fensel et al. [FSM03], pages 834–849.

[QYJ04] Yang Qing, Yang Yang, and Chen Juan. Goal-oriented platform based on knowledge-point: A new model of distance education system. In *18th International Conference on Advanced Information Networking and Applications (AINA 2004), 29-31 March 2004, Fukuoka, Japan*, pages 528–531. IEEE Computer Society, 2004.

[RLME05] Monique Reichert, Serge Linckels, Christoph Meinel, and Thomas Engel. Student's perception of a semantic search engine. In *IADIS Cognition and Exploratory Learning in Digital Age (CELDA), Porto, Portugal, December 14-16, 2005*, pages 139–147, 2005.

[Sch93] Renate A. Schmidt. Terminological representation, natural language & relation algebra. In H. J. Ohlbach, editor, *Proceedings of the sixteenth German AI Conference (GWAI-92)*, volume 671 of *Lecture Notes in Artificial Intelligence*, pages 357–371, Berlin, 1993. Springer.

[Sch03] Sabine Schulte im Walde. *Experiments on the Automatic Induction of German Semantic Verb Classes.* PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, June 2003.

[SM02] Volker Schillings and Christoph Meinel. tele-task: teleteaching anywhere solution kit. In *30th annual ACM SIGUCCS conference on User services, Providence, Rhode Island, USA, November 20-23, 2002*, pages 130–133, 2002.

[SP04] Evren Sirin and Bijan Parsia. Pellet: An owl dl reasoner. In Volker Haarslev and Ralf Möller, editors, *Description Logics*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[SSS91] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[WCK05] Janell D. Wilson, Sheila A. Cordry, and Nina King. Building learning communities with distance learning instruction. *AECT TechTrends*, 48(6):20–22, 2005.

[WMSB01] Wendy M. Williams, Faith Markle, Robert J. Sternberg, and Melanie Brigockas. *Educational Psychology*. Allyn & Bacon, 2001.

[You98] 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.

Appendix

A Description Logics

Definition 12 (Interpretation). *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 13 (Knowledge Base). *A knowledge base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a TBox, and $\mathcal{A}$ is an Abox.*

Definition 14 (Terminology). *Let A be a concept name and C a concept definition. Then $A \doteq C$ and $A \sqsubseteq C$ are terminological axioms. The first is a complete definition, the second an incomplete one. A terminology $\mathcal{T}$ is a finite set of terminological axioms such that no concept name appears more than once in the left-hand side of a definition. If a concept A occurs in the left-hand side of a definition, it is called defined concept. The other concepts are called primitive concepts.*

Definition 15 (Model of Assertions). *The interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies the concept assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and it satisfies the role assertion $R(a, b)$ if $(a, b)^{\mathcal{I}} \in R^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies the ABox $\mathcal{A}$ if it satisfies each assertion in $\mathcal{A}$. In this case we say that $\mathcal{I}$ is a model of the ABox.*

Definition 16 (Subsumption). *Let C and D be concept names, D subsumes C with respect to $\mathcal{T}$ (noted $\mathcal{T} \models C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretation $\mathcal{I}$ that satisfies $\mathcal{T}$.*

Definition 17 (Satisfiability). *A concept C is satisfiable with respect to $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is nonempty. In this case we say also that $\mathcal{I}$ is a model of C.*

Definition 18 (Instance Checking). *An assertion α (a concept assertion or a role assertion) is entailed by $\mathcal{A}$ (written $\mathcal{A} \models \alpha$) if every interpretation that satisfies $\mathcal{A}$, that is, every model of $\mathcal{A}$, also satisfies α.*

B Probabilistic context-free grammars

Probabilistic Context-Free Grammars PCFGs are an extension of context-free grammars in that they can model structural preferences (as for noun phrase structure), and degrees of acceptability (such as case assignment). But PCFGs fail when it comes to lexically sensitive phenomena such as PP-attachment, or selectional preferences of individual verbs, since they are based purely on structural factors.

Definition 19 (Probabilistic context-free grammar). *A probabilistic context-free grammar (PCFG) is a quintuple $\langle N, T, R, p, S \rangle$ with*

N	*finite set of non-terminal symbols*
T	*finite set of terminal symbols, $T \cap N = \emptyset$*
R	*finite set of rules $C \to \gamma$,* *$C \in N$ and $\gamma \in (N \cup T)^*$*
p	*corresponding finite set of probabilities on rules,* *$(\forall r \in R) : 0 \leq p(r) \leq 1$ and* *$(\forall C \in N) : \sum_\gamma p(C \to \gamma) = 1$*
S	*distinguished start symbol, $S \in N$*

Head-Lexicalised Probabilistic Context-Free Grammars Like other approaches, head-lexicalised probabilistic context-free grammars (H-L PCFGs) extend the idea of PCFGs by incorporating the lexical head of each rule into the grammar parameters. The lexical incorporation is realized by marking the head category on the right hand side of each context-free grammar rule, e.g. VP $\rightarrow$ V NP. Each category in the rule bears a lexical head, and the lexical head from the head child category is propagated to the parent category. The lexical head of a terminal category is the respective full or lemmatised word form. The lexical head marking in the grammar rules enables the H-L PCFG to instantiate the following grammar parameters:

- $p_{start}(s)$ is the probability that s is the category of the root node of a parse tree.
- $p_{start}(h|s)$ is the probability that a root node of category s bears the lexical head h.
- $p_{rule}(r|C,h)$ is the probability that a (parent) node of category C with lexical head h is expanded by the grammar rule r.
- $p_{choice}(h_C|C_P, h_P, C_C)$ is the probability that a (non-head) child node of category C_C bears the lexical head h_C, the parent category is C_P and the parent head is h_P .

In case a H-L PCFG does not include lemmatisation of its terminal symbols, either the lexical head h of a terminal node and the full word form $w \in T$ are identical and $p_{rule}(C \rightarrow w|C,h)$ is 1 (e.g. $p_{rule}(C \rightarrow runs|C, runs) = 1$), or the lexical head differs from the word form and $p_{rule}(C \rightarrow w|C,h)$ is 0 (e.g. $p_{rule}(C \rightarrow runs|C, ran)$). In case a grammar does include lemmatisation of its terminal symbols, the probability $p_{rule}(C \rightarrow w|C,h)$ is distributed over the different word forms w with common lemmatised lexical head h (e.g. $p_{rule}(C \rightarrow runs|C, run) = 0.3$, $p_{rule}(C \rightarrow run|C, run) = 0.2$, $p_{rule}(C \rightarrow ran|C, run) = 0.5$).

The probability of a syntactic tree analysis $p(t)$ for a sentence is defined as the product of the probabilities for the start category s, the rules r, and the relevant lexical heads h which are included in the tree. R refers to the set of rules established by the grammar, N to the set of non-terminal categories, and T to the set of terminal categories. Frequencies in the tree analysis are referred to by $f_t(r, C, h)$ for lexical rule parameters and $f_t(h_C, C_P, h_P, C_C)$ for lexical choice parameters. H-L PCFGs are able to rank syntactic analysis including lexical choices.

$$\begin{aligned} p(t) = \; & p_{start}(s) \cdot \\ & p_{start}(h|s) \cdot \\ & \prod_{r \in R, C \in N, h \in T} p_{rule}(r|C,h)^{f_t(r,C,h)} \cdot \\ & \prod_{C_p, C_C \in N; h_P, h_C \in T} p_{choice}(h_C|C_P, h_P, C_C)^{f_t(C_P, h_P, C_C)} \end{aligned}$$

ISBN 3-937786-89-9
ISBN 978-3-937786-89-6
ISSN 1613-5652