published as: Linckels S.,Christoph Meinel: Resolving Ambiguities in the Semantic Interpretation of Natural Language Questions;
Proceedings of Intelligent Data Engineering Automated Learning (IDEAL 2006);
Burgos, Spain, pp. 612-619, ISBN 978-3-540-45485-4.

Resolving Ambiguities in the Semantic Interpretation of Natural Language Questions

Serge Linckels[1,2] and Christoph Meinel[1]

[1] Hasso-Plattner-Institut (HPI), Potsdam University, P.O. Box 900460, D-14440 Potsdam, Gremany

[2] Luxembourg International Advanced Studies in Information Technologies (LIASIT), L-1511 Luxembourg, Luxembourg

Abstract. Our project is about an e-librarian service which is able to retrieve multimedia resources from a knowledge base in a more efficient way than by browsing through an index or by using a simple keyword search. The user can formulate a complete question in natural language and submit it to the semantic search engine.

However, natural language is not a formal language and thus can cause ambiguities in the interpretation of the sentence. Normally, the correct interpretation can only be retrieved accurately by putting each word in the context of a complete question.

In this paper we present an algorithm which is able to resolve ambiguities in the semantic interpretation of NL questions. As the required input, it takes a linguistic pre-processed question and translates it into a logical and unambiguous form, i.e. $\mathcal{ALC}$ terminology. The *focus function* resolves ambiguities in the question; it returns the best possible interpretation for a given word in the context of the complete user question. Finally, pertinent documents can be retrieved from the knowledge base.

We report on a benchmark test with a prototype that confirms the reliability of our algorithm. From 229 different user questions, the system returned the right answer for 97% of the questions, and only one answer, i.e. the best one, for nearly half of the questions.

1 Introduction

Our vision is to create an *e-librarian service* which is able to retrieve multimedia resources from a knowledge base in a more efficient way than by browsing through an index or by using a simple keyword search. The user formulates a complete question in natural langauge (NL), then the e-librarian service retrieves the most pertinent document(s) in which the user finds the answer to her/his question. The user's NL question is processed in three steps. Firstly, the linguistic pre-processing (section 3), secondly the translation of the linguistic pre-processed user question into a computer readable and unambiguous form w.r.t. a given ontology (section 4), and thirdly the retrieval of pertinent documents (section 5).

The main contribution of this paper is an algorithm, which is able to resolve ambiguities in the user question. The *focus function* returns the best interpretation for a given word in the context of the complete user question. A benchmark test confirms the reliability of this algorithm (section 6).

E. Corchado et al. (Eds.): IDEAL 2006, LNCS 4224, pp. 612–619, 2006.

2 Related Work

In this section, we present some related work from the fields of "Natural Language Interfaces to Databases" and "Question-Answering Systems". In general, the main difference is that our system's corpus is about a specific and well defined domain (computer history or fractions in mathematics), whereas other related projects deal with larger corpora and/or other domains. Also, some projects focus more on the NLP of the user question; we only use linguistic tools to overcome this step. Furthermore, other projects do not have an "ontological approach" like in our case in order to map a sentence into an logical and unambiguous form, i.e. $\mathcal{ALC}$ terminology.

START [2] is the first question-answering system available on the Web. Several improvements have been made since it came online in 1993. However, the NLP is not always sound, e.g. the question "What did Jodie Foster before she became an actress?" returns "I don't know what Jodie fostered before the actress became an actress". Also, the question "Who invented the transistor?" yields two answers: the inventors of the transistor, but also a description about the transistor (the answer to the question: "What is a transistor").

AquaLog [4] is a portable question-answering system which takes queries expressed in NL and an ontology as input, and returns answers drawn from one or more knowledge bases. User questions are expressed as triples: <subject, predicate, object>. If the several translation mechanisms fail, then the user is asked for disambiguation. The system also uses an interesting learning component to adapt to the user's "jargon". AquaLog has currently a very limited knowledge space. In a benchmark test over 76 different questions, 37 (48.68%) where handled correctly.

The prototype PRECISE [5] uses ontology technologies to map semantically tractable NL questions to the corresponding SQL query. It was tested on several hundred questions drawn from user studies over three benchmark databases. Over 80% of the questions are semantically tractable questions, which PRECISE answered correctly, and recognized the 20% it could not handle, and requests a paraphrase. The problem of finding a mapping from the tokenization to the database requires that all tokens must be distinct; questions with unknown words are not semantically tractable and cannot be handled.

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

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

3 Linguistic Pre-processing

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

4 Ontology Mapping

4.1 Ontology Preliminaries

The e-librarian service masters a domain language L_H over an alphabet Σ^*, which may or may not contain all the possible words L used by the user to formulate his question, so that $L_H \subseteq L \subseteq \Sigma^*$. The semantics are attached to each word by classification in the knowledge source, which is structured in a hierarchical way like *hyperonyms*, *hyponyms*, *synonyms*, and *homonyms*, e.g. *WordNet*.

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

A node v_i represents a concept. The words that refer to this concept are regrouped in T_i. We assume that each set of words T_i is semantically related to the concept that the node v_i represents. Of course, a certain word can refer to different concepts, e.g. "Ada" is the name of a programming language but also the name of a person. Not all words in L_H must be associated with a concept. Only words that are semantically relevant are classified. In general, nouns and verbs are best indicators of the sense of a question.

4.2 Semantic Interpretation

The representation of context-independent meaning is called the *logical form*, and the process of mapping a sentence to its logical form is called *semantic interpretation* [1]. The logical form is expressed in a certain knowledge representation language; we use the *Description Logics* (DL) $\mathcal{ALC}$ language,which is sufficiently expressive for our purposes. 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 [6]. Finally, translating the user question into DL allows direct reasoning over the OWL-DL encoded knowledge base (section 5). A DL terminology is composed, firstly, of *concepts* (unary predicates), which are generally nouns, question words (*w-words*) and proper names, and secondly, of *roles* (binary predicates), which are generally verbs, adjectives and adverbs. The core part of the semantic interpretation is a mapping algorithm—commonly called *non-standard inference* [3]—which maps each word from the user question to one or more ontology concepts, and resolves the arguments of each role by analyzing the syntactic structure of the sentence.

Definition 2 (Mapping). *The meaning of each word $w_k \in L$ is made explicit with the mapping function $\varphi : L \rightarrow V$ over an ontology dictionary $L_H \subseteq L \subseteq \Sigma^*$ and an $\mathcal{ALC}$ concept taxonomy $H = (V, E, v_0)$ so that $\varphi(w_k)$ returns a set of interpretations Φ defined as follows,*

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

The function $ft(v_i)$ returns the set of words T_i associated to the node v_i (definition 1), and $w_k \equiv x$ are two equivalent words respecting a given tolerance. This solution gives good results even if the user makes spelling errors. Furthermore, only the best matching is considered for the mapping, e.g. the word "comXmon" will be considered as "common", and not as "uncommon". Both words, "common" and "uncommon", will be considered for the mapping of "comXXmon". The ambiguity will be resolved in a further step (focus function).

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

Definition 3 (Focus). *The focus of a set of interpretations Φ is made explicit by the function f which returns the best interpretation for a given word in the context of the complete question q. The focus is written $f_q(\varphi(w_k \in q)) = v'$.*

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

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

$s_2 = Creator$. In the question $q =$ "Who invented Ada?" the following mappings are computed:

$$\varphi(\text{"Who"}) = \{Creator\}$$
$$\varphi(\text{"invented"}) = \{wasInventedBy(Thing, Creator)\}$$
$$\varphi(\text{"Ada"}) = \{Person, Language\}$$

The system detects an ambiguity for the word "Ada", which is mapped to an instance of the concept *Person*, but also to an instance of the concept *Language*. The focus function computes the following combinations to resolve the ambiguity:

1. Was Ada invented by who?*	*wasInventedBy("Ada","Who")*
2. Was Ada invented by Ada?	*wasInventedBy("Ada","Ada")*
3. Was who invented by Ada?*	*wasInventedBy("Who","Ada")*
4. Was who invented by who?*	*wasInventedBy("Who","Who")*

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

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

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

Definition 4 (Semantic interpretation). *Let q be the user question, which is composed of linguistic clauses, written $q = \{q'_1, ..., q'_m\}$, with $m \geq 1$. The sematic interpretation of a user question q is the translation of each linguistic clause into an $\mathcal{ALC}$ terminology w.r.t. a given ontology H written,*

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

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

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.

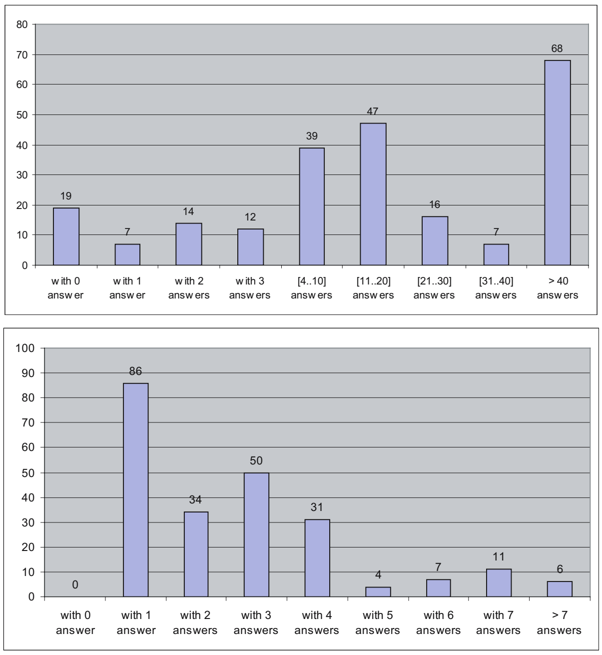

Fig. 1. Number of results yielded by a (1) keyword and by a (2) semantic search engine with a set of 229 questions

5 Retrieval

Logical inference over the non-empty ABox from the knowledge base $\mathcal{K}$ is possible by using a classical DL reasoner; we use *Pellet* [7]. The returned results are logical consequences of the inference rather than of keyword matchings. 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".

6 Benchmark Tests

Our background theory was implemented prototypically in an educational tool about fractions mathematics. We used an educational knowledge base about fractions in mathematics from the university of Luxembourg. The knowledge base is composed of short multimedia documents (*clips*), which were recorded with tele-TASK (http://www.tele-task.de). All clips were semantically described with OWL-DL metadata, w.r.t. an ontology H. The same ontology H was used to translate the NL questions as explained in section 4. Let us remark that although our algorithm does currently not profit from the full expressivity of OWL-DL, which is $\mathcal{SHOIN}(\mathbf{D})$, it allows to have compatible semantics between the OWL-DL knowledge base, and the less expressive $\mathcal{ALC}$ translated questions.

In a benchmark test we used our prototype to measure the performance of our semantic search engine. A testing set of 229 different questions about fractions in mathematics was created by a mathematics teacher, who was not involved in the development of the prototype. The teacher also indicated manually the best possible clip, as well as a list of further clips, that should be yielded as correct answer. The questions were linguistic correct, and short sentences like students in a secondary school would ask, e.g. "How can I simplify a fraction?", "What is the sum of $\frac{2}{3}$ and $\frac{7}{4}$?", "What are fractions good for?", "Who invented the fractions?", etc. This benchmark test was compared with the performance of a keyword search engine. The keyword search was slightly optimized to filter out stop words (words with no relevance, e.g. articles) from the textual content of the knowledge base and from the questions entered.

The semantic search engine answered 97% of the questions (223 out of 229) correctly, whereas the keyword search engine yielded only a correct answer (i.e. a pertinent clip) in 70% of the questions (161 out of 229). For 86 questions, the semantic search engine yielded just one—the semantically best matching—answer (figure 1). For 75% of the questions (170 out of 229) the semantic search engine yielded just a few results (one, two or three answers), whereas the keyword search yielded for only 14% of the questions less than 4 answers; mostly (138 questions out of 229) more than 10 answers. For example, the semantic interpretation of the question "What is the sum of $\frac{2}{3}$ and $\frac{7}{4}$?" is the following valid $\mathcal{ALC}$ terminology and its corresponding ABox query:

$$Fraction \sqcap \exists\, hasOperation.(Operation \sqcap \exists hasType.Operator)$$

$$Fraction(x1) \wedge hasOperation(x1, x2) \wedge Operation(x2) \wedge hasType(x2, sum)$$

The keyword search engine yields all clips, in which keywords like "sum" are found, e.g. a clip that explains how to represent a complex function in terms of additions, and a clip that explain how to describe situations with simple fractions.

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

7 Conclusion

In this paper we presented an algorithm which is able to resolve ambiguities in the semantic interpretation of NL questions. It takes as input a linguistic preprocessed question and translates it into a logical and unambiguous form, i.e. $\mathcal{ALC}$ terminology. The *focus function* resolves ambiguities in the question; it returns the best interpretation for a given word in the context of the complete user question. Finally, pertinent documents can be retrieved from the knowledge base.

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

References

1. J. Allen. *Natural Language Understanding*. Addison Wesley, 1994.
2. B. Katz. Annotating the world wide web using natural language. In *5th RIAO conference on computer assisted information searching on the internet, Montreal, Canada*, 1997.
3. R. Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001. Ph.D. thesis.
4. V. Lopez, M. Pasin, and E. Motta. Aqualog: An ontology-portable question answering system for the semantic web. In *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece*, pages 546–562, 2005.
5. A.-M. Popescu, O. Etzioni, and H. A. Kautz. Towards a theory of natural language interfaces to databases. In *8th International Conference on Intelligent User Interfaces, January 12-15, Miami, FL, USA*, pages 149–157, 2003.
6. R. A. Schmidt. Terminological representation, natural language & relation algebra. In *16th German AI Conference (GWAI)*, volume 671 of *Lecture Notes in Artificial Intelligence*, pages 357–371, 1993.
7. E. Sirin and B. Parsia. Pellet: An owl dl reasoner. In *International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.