English Querying over Ontologies: E-QuOnto*

Raffaella Bernardi[1], Francesca Bonin[1,2], Diego Calvanese[1], Domenico Carbotta[1], and Camilo Thorne[1]

[1] Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
lastname@inf.unibz.it
[2] Dipartimento di Linguistica - Università di Pisa

Abstract. Relational database (DB) management systems provide the standard means for structuring and querying large amounts of data. However, to access such data the exact structure of the DB must be know, and such a structure might be far from the conceptualization of a human being of the stored information. Ontologies help to bridge this gap, by providing a high level conceptual view of the information stored in a DB in a cognitively more natural way. Even in this setting, casual end users might not be familiar with the formal languages required to query ontologies. In this paper we address this issue and study the problem of ontology-based data access by means of natural language questions instead of queries expressed in some formal language. Specifically, we analyze how complex real life questions are and how far from the query languages accepted by ontology-based data access systems, how we can obtain the formal query representing a given natural language question, and how can we handle those questions which are too complex wrt the accepted query language.

1 Introduction

Relational database management systems (RDBMs) provide the standard means for structuring, modeling, declaring, updating and querying large amounts of structured data. The interfaces of these systems are based on formal query languages, such as SQL, that combine both declarative and imperative features (cf. [1]). Crucially, the expressive power of these formal languages is well-known, and query answering in relational databases (DBs) can be carried out efficiently in the size of the data. More precisely, the *data complexity* (i.e., the complexity measured in the size of the DB) of answering SQL (or First-Order) queries is in LogSpace [1], and it is precisely this property that allows RDBMSs to handle in practice very large amounts of data. However, access to these data requires knowledge of the exact structure of the DB, which might be far from the conceptualization that human beings have of the represented information. Ontologies

* This research has been partially supported by FET project TONES (Thinking ONtologiES), funded within the EU 6th Framework Programme under contract FP6-7603.

help to overcome these limitations by providing a high level conceptual view of the stored information in a cognitively more natural way[12]. Recently, the problem of ontology based access to DBs has been studied. This problem poses particular challenges since on the one hand, it requires to deal with the large amount of information stored in DBs, and on the other hand, it requires to cope with incomplete knowledge[1]. In fact, the ontology encodes constraints on the domain of interest w.r.t. which data in the DB might be incomplete. For instance, the ontology might require that all instances of some class have to participate to a relation, but for some instance the DB does not contain the corresponding facts. For a concrete example, let's assume, that the ontology requires that each student attends at least one course, but the DB happens not to have any information about the course attended by the student John.

Against this background, a family of ontology languages has been proposed recently that is based on Description Logics [3], namely the *DL-Lite* family [9, 8]. The kind of constraints that can be expressed in variants of *DL-Lite* tightly correspond to constraints typically encountered in conceptual data models used in DBs and software engineering. A distinguishing feature of the logics of the *DL-Lite* family is that they allow for efficiently answering queries posed to an ontology with an underlying DB, taking into account that the latter may be incomplete with respect to the constraints expressed by the ontology. The kind of queries supported in *DL-Lite* are *unions of conjunctive queries* (UCQs): *conjunctive queries* (CQs) correspond to the fragment of First-Order Logic (FOL) whose formulas are conjunctions of atoms over constants, existentially quantified variables that may be shared across the atoms, and free variables (also called *distinguished variables*)[2]. An UCQ corresponds to a disjunction of CQs, all with the same number of distinguished variables.

This paper builds on these results and considers a further problem, namely the fact that using formal query languages requires some previous training and can prove counter-intuitive to a casual end user. For such a user the intuitive appeal and understanding of the machine interface can be crucial. It would thus be suitable in such cases to shift to natural language (NL) and to use natural language questions instead of queries expressed in some formal language. In order to reach this goal, in this paper we try to answer the following questions: (*i*) how complex are natural language questions a user would ask to access data in a DB; (*ii*) how far are these real life questions from conjunctive queries; (*iii*) given a natural language question how can we obtain the formal query (in a suitable query language) representing it; (*iv*) how can we handle those questions which cannot be represented by conjunctive queries. Question (*i*) is addressed in Section 3 whereas (*ii*)–(*iv*) are the focus of Section 4. Before presenting our results, we introduce in Section 2 some preliminary technical notions. The paper presents work in progress, the ideas we are planning to further explore are summarised in Section 5.

[1] A further important issue that may arise in the presence of an ontology is inconsistency w.r.t. the ontology. However, we do not deal with inconsistency here.

[2] In terms of SQL, CQs correspond to the SELECT-PROJECT-JOIN fragment.

2 Preliminaries

We provide now the technical preliminaries underlying languages used in the context of ontology-based data access, both for the specification of the ontology, and for specification of queries over the ontology to access underlying data sources. When asking ourselves which are the formalisms most suited to represent the information about a domain of interest in an ontology, we can draw on the large body of research carried out in the last twenty years in structured knowledge representation, and specifically in the area of Description Logics. Description Logics (DLs) [3] are logics that allow one to structure the domain of interest by means of *concepts*, denoting sets of objects, and *roles*, denoting binary relations between (instances of) concepts. Complex concepts and role expressions are constructed starting from a set of atomic concepts and roles by applying suitable constructs. The domain of interest is then represented by means of a DL knowledge base (KB), consisting of a TBox, storing intensional information, and an ABox, storing assertional information about individual objects of the domain of interest.

We start by defining the ontology language we make use of, which is based on the recently introduced *DL-Lite* family of DLs [9, 8]. Such a family of DLs is specifically tailored for an optimal tradeoff between expressive power and computational complexity of inference in the context of ontology-based data access. In the logics of the *DL-Lite* family, the TBox is constituted by a set of *assertions* of the form

$Cl \sqsubseteq Cr$ *(concept inclusion assertion)* (funct R) *(functionality assertion)*
$R_1 \sqsubseteq R_2$ *(role inclusion assertion)*

In the above assertions, Cl and Cr denote concepts that may occur respectively on the left and right-hand side of inclusion assertions, and R, R_1, R_2 denote roles, constructed according to the following syntax:

$$\begin{array}{rcl} Cl & \longrightarrow & A \mid \exists R \mid Cl_1 \sqcap Cl_2 \\ Cr & \longrightarrow & A \mid \exists R \mid Cr_1 \sqcap Cr_2 \mid \exists R.A \mid \neg A \mid \neg \exists R \end{array} \qquad \begin{array}{rcl} R & \longrightarrow & P \mid P^- \end{array}$$

where A denotes an atomic concept, and P denotes an atomic role.

The $\exists R$ construct is called *unqualified existential quantification*, and intuitively denotes all objects that are connected through role R to some (not further specified) object. The $\exists R.A$ construct, called *qualified existential quantification*, allows one to further qualify the object connected through role R as being an instance of concept A. Also, $\sqcap$ denotes conjunction, and $\neg$ negation (or complement). Finally, P^- denotes the binary relation that is the inverse of the one denoted by P.

We formally specify the semantics of *DL-Lite*, by providing its translation to FOL. Specifically, we map each concept C (we use C to denote an arbitrary concept, constructed applying the rules above) to a FOL formula $\varphi(C, x)$ with one free variable x (i.e., a unary predicate), and each role R to a binary predicate

$\varphi(R, x, y)$ as follows:

$$\varphi(A, x) = A(x) \qquad \varphi(\exists R, x) = \exists y(\varphi(R, x, y))$$
$$\varphi(\neg C, x) = \neg\varphi(C, x) \qquad \varphi(\exists R.C, x) = \exists y(\varphi(R, x, y) \wedge \varphi(C, y))$$
$$\varphi(C_1 \sqcap C_2, x) = \varphi(C_1, x) \wedge \varphi(C_2, x)$$
$$\varphi(P, x, y) = P(x, y) \qquad \varphi(P^-, x, y) = P(y, x)$$

An inclusion assertion $Cl \sqsubseteq Cr$ of the TBox corresponds then to the universally quantified FOL sentence $\forall x(\varphi(Cl, x) \rightarrow \varphi(Cr, x))$. Similarly, $R_1 \sqsubseteq R_2$ corresponds to $\forall x \forall y(\varphi(R_1, x, y) \rightarrow \varphi(R_2, x, y))$. Instead, a functionality assertion (funct R), imposes that the binary predicate R is functional, i.e., $\forall x \forall y \forall z(\varphi(R, x, y) \wedge \varphi(R, x, z) \rightarrow y = z)$.

In *DL-Lite*, an ABox is constituted by a set of assertions on *individuals*, of the form $A(c)$ or $P(a, b)$, where A and P denote respectively an atomic concept and an atomic role, and a, and b denote constants. As in FOL, each constant is interpreted as an element of the interpretation domain, and we assume that distinct constants are interpreted as distinct individuals, i.e., we adopt the *unique name assumption* (UNA). The above ABox assertions correspond to the analogous FOL facts, or, by resorting to the above mapping, to $\varphi(A, x)(c)$ and $\varphi(R, x, y)(a, b)$, respectively.

A *model* of a *DL-Lite* KB is a FOL model of the conjunction of FOL formulas representing its semantics.

It is worth noticing that, by means of the constructs present in *DL-Lite*, one can capture almost all features of conceptual data models used in DBs and software engineering, such as the Entity-Relationship model [4] or UML class diagrams[3] Hence, the DLs of the *DL-Lite* family are well suited also to represent data stored in commercial DBMSs.

Indeed, in the setting where such data are accessed through an ontology, the DL ABox is actually represented by the DB. Alternatively, the ABox may be reconstructed from the data present in the DB through suitable mappings, that allow one also to overcome the impedance mismatch between the data values stored in the DB and the objects at the conceptual level [7].

Reasoning has been studied for several variants of logics of the *DL-Lite* family. Specifically, the logic obtained by dropping functionality assertions is called *DL-Lite*$_R$, and has already been adopted in the context of natural language specification of ontologies [6, 5]. Instead, the logic obtained by dropping role inclusion assertions and also the construct for qualified existential quantification is called *DL-Lite*$_\mathcal{F}$, and is quite close to formalisms used in conceptual modeling, such as the Entity-Relationship model. A combination of all constructs and types of assertions considered above, with some restriction on the possible interaction of functionality and role inclusions, has also been studied [7].

It has been shown in [9, 8] that for the above mentioned variants of *DL-Lite* all relevant reasoning services (e.g., KB satifiability, subsumption, ecc.),

[3] Complete (also called covering) hierarchies are an exception, since they require some form of disjunction, which is *not* present in *DL-Lite* (see [8] for motivations).

are polynomial in the size of the knowledge base, and LogSpace in the size of the ABox only, i.e., in *data complexity* (cf. [1]).

Given an *DL-Lite* KB $\mathcal{K}$, queries over $\mathcal{K}$ are *unions of conjunctive queries* (UCQs), which are expressions of the form $\{\boldsymbol{x} \mid \exists \boldsymbol{y}_1 \mathit{conj}_1(\boldsymbol{x}, \boldsymbol{y}_1) \vee \cdots \vee \exists \boldsymbol{y}_n \mathit{conj}_n(\boldsymbol{x}, \boldsymbol{y}_n)\}$, where $\boldsymbol{x}$ is a (possibly empty) finite sequence of *distinguished variables*, each $\boldsymbol{y}_i$ is a finite sequences of (existentially quantified) variables and constants, and each conj_i is a conjunction of atoms whose predicates are the concept and role symbols of the KB. If there is only one disjunct (i.e., $n = 1$), the query is called a *conjunctive query* (CQ). The FOL formula following | is called the *body* of the query. Each distinguished variable must appear also in the body of the query. A *boolean query* is one where sequence of distinguished variables is empty.

As an example, consider the natural language questions *"Which are the red books"?*, corresponding to the CQ $\{x \mid \mathsf{book}(x) \wedge \mathsf{red}(x)\}$, and *"Which are the books read by John?"*, corresponding to the CQ $\{x \mid \mathsf{book}(x) \wedge \mathsf{read}(\mathsf{john}, x)\}$. Also, CQs may allow one to represent complex dependencies coming from relative pronouns, e.g., *"which are the students who attend a course which is taught by their father?"* can be represented by $\{x \mid \exists y_1 \exists y_2(\mathsf{student}(x) \wedge \mathsf{attend}(x, y_1) \wedge \mathsf{teach}(y_2, y_1) \wedge \mathsf{father}(y_2, x))\}$.

Note that UCQs may contain no negation, no universal quantification, and that disjunction may appear only at the outermost level. Hence, CQs and UCQs constitute a *proper* fragment of FOL (in particular, we lack a complete set of boolean operators). In terms of SQL, it is well known that CQs correspond to the SELECT-PROJECT-JOIN fragment of SQL.

As an example, the questions *"What is causing* all *the joint pain?"* and *"What is the chance that aspirating a joint effusion that is* not *red will help the patient?"* cannot be translated into CQs or UCQs.

Given an interpretation $\mathcal{I}$ of $\mathcal{K}$, the *semantics* of a UCQ $q(\boldsymbol{x})$, denoted $q^{\mathcal{I}}$, is given by the set of tuples $\boldsymbol{c}$ of *constants*, such that, if each constant in $\boldsymbol{c}$ is assigned to the corresponding variable in $\boldsymbol{x}$, the formula that constitutes the body of q evaluates to true in $\mathcal{I}$. Notice that for a boolean query q, $q^{\mathcal{I}}$ is either empty or constituted by the empty tuple only.

It is well-known (cf. [1]) that given a finite interpretation $\mathcal{I}$ (that in our setting plays the role of a traditional database), $q^{\mathcal{I}}$ can be computed in LogSpace in the size of $\mathcal{I}$, i.e., in *data complexity*. However, the setting we are considering here is complicated by the fact the we are usually not given a single interpretation $\mathcal{I}$, but rather a DL KB $\mathcal{K}$, and are interested in reasoning with respect to all models of $\mathcal{K}$. In other words, we are interested in the answers to queries in the presence of *incomplete information* in the ABox/database with respect to the constraints specified by the TBox. Formally, given $\mathcal{K}$ and a UCQ $q(\boldsymbol{x})$, we are interested in computing the so-called *certain answers* to q over $\mathcal{K}$, which are defined as the set of tuples $\boldsymbol{c}$ of constants that are in $q^{\mathcal{I}}$ for *every* model $\mathcal{I}$ of $\mathcal{K}$. The problem of computing certain answers has been studied for various variants of *DL-Lite*, and it has been shown that is polynomial in the size of $\mathcal{K}$ and LogSpace in data complexity, i.e., in the size of the data constituted by the ABox (or the

database representing it) [9, 8]. However, this property crucially depends on the constructs in the DL and on the adopted query language, and does not carry over if we e.g., allow for negation or universal quantification in queries.

Recently, the system QuOnto [2] has been developed, which implements (sound and complete) algorithms for computing certain answers to UCQs over *DL-Lite* ontologies, by relying on the underlying DBMS for the actual execution of the queries and retrieval of the data.

3 Analysis of Questions Asked by Users

We are interested in understanding which natural language questions can be expressed by CQs (and UCQs) and which ones cannot. "How" and "why" questions are clearly outscoping CQs, which return only sets of tuples of objects. On the other hand, both boolean (i.e., yes/no) questions and the wh-questions built out of "which", "what", "when", "where", and "who" could be expressed as CQs if their semantic representation does not contain any of the operations not admitted in CQs, specifically disjunction, negation, and universal quantification. The question we try to address in this section is how frequent this happens and how complex are the structures of these questions. To this end, we have analysed several corpora of questions on different domains and asked in different settings:

Clinical Questions: contains users' questions on the clinical domain, mostly asked by doctors to colleagues; 435 questions, vocabulary: 3495, total tokens: 40489 (questions with introduction).

Answer.com: contains questions on different topics (e.g., art, sport, computers) asked by internet users; 444 questions[4], vocabulary: 1639, total tokens: 5791 (without introduction)[5].

TREC: we have used the TREC 2004 corpus that contains 408 questions.

To understand how often FOL constructs outside the CQ fragment occur in real life questions, we have checked the occurrences of universal quantification, negation and disjunction in the corpora listed above, by searching for the frequency of the terms shown in Table 1. We are aware that these terms

[4] `http://clinques.nlm.nih.gov/`

[5] `http://wiki.answers.com/Q/WikiFAQs:Finding_Questions_to_Answer`

Table 1. Searched Terms

Operator	**Linguistic terms**
Universal quantification	all, both, each, every, everybody, everyone, any (in positive context), none, nothing
Disjunction	or
Negation	not (and its abbreviations), without
Existential quantification	any, anything, anyone, anybody, some, something, someone, somebody, there is a, there are a, there was a

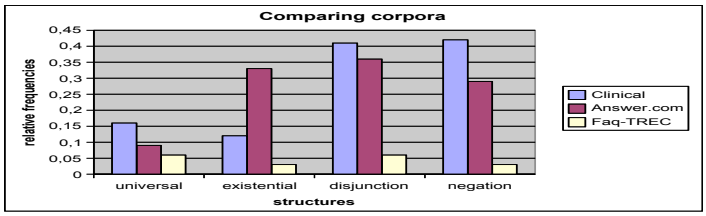

Fig. 1. Summary of the analysis

might not cover all the possible ways of expressing the operations under investigation, however, we believe the results do help revealing features of natural language questions with respect to the problem we are trying to tackle. To have a more general picture of the occurrences of logical operators in questions, we considered also the frequency of existential quantifiers. This was also needed in order to check whether the latter were negated, hence resolving into universal quantification. Similar studies have been conducted on other corpora containing declarative sentences and the results were different than those reported here, which highlights the peculiarity of questions.

In Figure 1, we report the results of such analyses. The chart shows the frequency of each class of terms in the three corpora. Frequency values refer to the normalized relative frequency (number of tokens/total word count multiplied by 100). As the figure shows, the use of the "forbidden" constructs is particularly frequent in natural questions even in a free text access to data. The details about the number of questions in which these operators occur are given in Table 2.

More specifically, the comparison among the corpora shows that: (*i*) universal quantifiers are rare in all the analysed questions; they are slightly more frequent in the Clinical Questions corpus where, anyway, they mostly occur in the declarative sentences preceding the question; (*ii*) existentials do occur in questions but never negated; (*iii*) negation and disjunction are more frequent particularly in real free text access, but still rare in questions like those in TREC.

Though rare, still there is the need of handling those questions that outscope the expressivity of CQs. Therefore, we now turn to analyse them in detail.

Universal quantification. Universal quantifiers have been found to occur in any syntactic position: as subject (1a) or object (1b) as well as in prepositional

Table 2. Number of questions

	Clinical Questions Tot: 435	**Answer.com** Tot: 444	**TREC** Tot: 408
universal	12	6	2
existential	111	22	1
disjunction	132	15	2
negation	52	13	1

phrases (1c), as exemplified by the following questions taken from the Clinical Questions corpus:

(1a) *Should* all *pregnant women take a test for human immunodeficiency virus?*
(1b) *What is causing* all *the joint pain?*
(1c) *Can we use Energix for* all *five doses of DPT?*

Negation. In the analysed corpora, "negation" is used only to negate the verb of an embedded clause (2a) (never the main verb of the question or the auxiliary), or to negate adjectives (2b) or in the form of *without* (2c), as exemplified again by the Clinical Questions corpus:

(2a) *Should you give a full series of tetanus shots to an adult who* does not *know their immunization history?*
(2b) *What is the chance that aspirating a joint effusion that is* not *red or tender will yield anything diagnostically?*
(2c) *Is it possible for someone to get recurrent pelvic disease* without *a new exposure?*

Disjunction. Finally, disjunction occurs only coordinating nouns and really rarely between propositions. See the examples below taken from Answer.com.

(3a) *What is the word for the fear if viewing sports* or *playing sports?*
(3b) *Is Liberia considered a rural* or *a urban country?*

In such rare cases, we would need to resort to a UCQ, rather than a CQ, obtained by splitting up the query at each occurrence of a disjunction.

The analysed corpora contain questions whose structure is rather simple and not too far away from CQs' constructs. This could be claimed even strongly if we focus on questions asked by users to DBs. For English, we have looked into Geoquery[6] (304 questions on the geographical domain) where there is only one question of the "forbidden" class, namely one containing negation –though the corpus contains several (75) questions falling outside CQs, viz. aggregation questions (how high, how many, etc.). Similar studies have been carried out for Italian obtaining similar results.

In the remainder of the paper, we show how to build the CQ representation of a given question, and propose a method, based on *semantic weakening*, that allows us to extend the kinds of natural language questions that query answering tools over ontologies can handle.

4 Answering Natural Language Questions over Ontologies

Our third question, namely "how can we obtain a formal query (in a suitable query language) representing a given natural language question?", can be rephrased in how well current Computational Semantics tools will perform when

[6] `http://www.cs.utexas.edu/users/ml/geo.html`

their output is used as input for query answering tools, as e.g., QuOnto, which accepts UCQs. Hence, we start from off-the-shelf wide-coverage parsing tools producing FOL meaning representation of the parsed questions. In this paper we are considering a CCG parser (paired with Boxer) [11] which outputs questions of the form[7]:

$$\{x \mid \exists \boldsymbol{y}(\varphi(\boldsymbol{y}) \wedge D(x) \wedge \exists \boldsymbol{z}(\psi(x, \boldsymbol{y}, \boldsymbol{z})))\} \tag{1}$$

where x represents the answer to the question, D is the domain of the question, ψ represents the body of the question, and φ represents the knowledge that is presupposed by the user posing the question. We consider wh-questions and boolean questions. In the first case, if the wh-phrase is *where*, *who/whom*, *how*, *when*, *what*, or *why*, then D refers to *location*, *person*, *manner*, *unit_of_time*, *thing* and *reason*, respectively. If the wh-phrase is *which*, then the domain is the head noun of the noun phrase. For example, in the question *"Which services are offered by the library?"* the domain is *services*. If the question is boolean, then the domain of the question is empty. In both cases the body of the question contains a conjunction of conditions.

For instance, the question *"Who may use the Interlibrary Loan service?"* is represented as follows:

$$\{x \mid \exists y_1 \exists y_2(\mathsf{loan}(y_1) \wedge \mathsf{interlibrary}(y_2) \wedge \mathsf{service}(y_2) \wedge \mathsf{nn}(y_1, y_2) \wedge \mathsf{person}(x) \wedge \exists z(\mathsf{use}(z) \wedge \mathsf{event}(z) \wedge \mathsf{agent}(z, x) \wedge \mathsf{patient}(z, y_2)))\}$$

As the example shows, besides the predicates introduced by the words in the question, the meaning representation produced by Boxer contains information about thematic roles too, namely agent, patient, theme, whose first argument appears also as argument of the unary predicate event, which in itself is introduced by a verb. Also, the formula in the example contains the predicate nn, which relates nouns that in the question occur as multi words (e.g., interlibrary loan).

In short, Boxer produces formulas of the fragment of FOL with equality and without function symbols, where predicate symbols are either unary or binary. This output must be translated into queries suitable for query answering tools. Here we focus on the structure of the logical form rather than on the predicates to be used for properly matching the ontology vocabulary.

4.1 Translation

The translation to an UCQ starts from the FOL representation of the question, as calculated by Boxer, and hence in form (1). We deal first with the easy case where both φ and ψ in that formula contain conjunctions and disjunctions only. Let $\mathsf{dnf}(\varphi) = \alpha_1 \vee \cdots \vee \alpha_n$ and $\mathsf{dnf}(\psi) = \beta_1 \vee \cdots \vee \beta_m$ be the disjunctive normal form (DNF) expansions of φ and ψ, respectively. Query (1) is then expanded into an UCQ as follows:

[7] Boxer actually outputs such open FOL formulas as closed formulas in which the free variable is universally quantified and the second conjunction is represented by an implication.

$$
\begin{aligned}
q(x) &= \{x \mid \exists \boldsymbol{y} \exists \boldsymbol{z}(\varphi(\boldsymbol{y}) \wedge D(x) \wedge \psi(x, \boldsymbol{y}, \boldsymbol{z}))\} \\
&= \{x \mid \exists \boldsymbol{y} \exists \boldsymbol{z}((\alpha_1(\boldsymbol{y}) \vee \cdots \vee \alpha_n(\boldsymbol{y})) \wedge D(x) \wedge (\beta_1(x, \boldsymbol{y}, \boldsymbol{z}) \vee \cdots \vee \beta_m(x, \boldsymbol{y}, \boldsymbol{z})))\} \\
&= \{x \mid \exists \boldsymbol{y} \exists \boldsymbol{z}((\alpha_1(\boldsymbol{y}) \wedge D(x) \wedge \beta_1(x, \boldsymbol{y}, \boldsymbol{z})) \vee \cdots \vee (\alpha_1(\boldsymbol{y}) \wedge D(x) \wedge \beta_m(x, \boldsymbol{y}, \boldsymbol{z})) \vee \\
&\qquad \cdots \\
&\qquad (\alpha_n(\boldsymbol{y}) \wedge D(x) \wedge \beta_1(x, \boldsymbol{y}, \boldsymbol{z})) \vee \cdots \vee (\alpha_n(\boldsymbol{y}) \wedge D(x) \wedge \beta_m(x, \boldsymbol{y}, \boldsymbol{z})))\} \\
&= \{x \mid \exists \boldsymbol{y} \exists \boldsymbol{z}(\alpha_1(\boldsymbol{y}) \wedge D(x) \wedge \beta_1(x, \boldsymbol{y}, \boldsymbol{z})) \vee \cdots \vee \exists \boldsymbol{y} \exists \boldsymbol{z}(\alpha_n(\boldsymbol{y}) \wedge D(x) \wedge \beta_m(x, \boldsymbol{y}, \boldsymbol{z}))\}
\end{aligned}
$$

4.2 Semantic Weakening

Though rarely, universal quantifiers do occur in questions asked by users. Here we discuss a way to deal with this kind of questions. Techniques (and systems for) query answering in the setting of incomplete information provide the *certain answers* to a query – i.e., given a query in the form $q(\boldsymbol{x}) = \{\boldsymbol{x} \mid \exists \boldsymbol{y}(\varphi(\boldsymbol{x}, \boldsymbol{y}))$, it returns the tuples of constants $\boldsymbol{c}$ that are guaranteed to satisfy the formula $\exists \boldsymbol{y}(\varphi(\boldsymbol{c}, \boldsymbol{y}))$ in all models of the ontology. Certain answers can also be characterized by resorting to the minimal knowledge operator $\mathbf{K}$ [14, 15], where $\mathbf{K}\varphi$ denotes that "φ is known to hold by the ontology". In this way, the answers to the query q above can be defined as the tuples $\boldsymbol{c}$ of constants that satisfy the epistemic formula $\mathbf{K}\exists \boldsymbol{y}(\varphi(\boldsymbol{c}, \boldsymbol{y})))$.

A question containing universal quantification results in a query of the form:

$$\varphi(\boldsymbol{x}) \wedge \forall \boldsymbol{y}(D(\boldsymbol{y}) \rightarrow \exists \boldsymbol{z}(\psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})))$$

Its intended semantics are captured by the equivalent epistemic query:

$$\mathbf{K}(\varphi(\boldsymbol{x}) \wedge \forall \boldsymbol{y}(D(\boldsymbol{y}) \rightarrow \exists \boldsymbol{z}(\psi(\boldsymbol{c}, \boldsymbol{y}, \boldsymbol{z})))) \equiv \mathbf{K}\varphi(\boldsymbol{x}) \wedge \mathbf{K}\,\forall \boldsymbol{y}(D(\boldsymbol{y}) \rightarrow \exists \boldsymbol{z}(\psi(\boldsymbol{c}, \boldsymbol{y}, \boldsymbol{z})))$$

The occurrence of a universal quantification in the scope of the $\mathbf{K}$ operator leads to a query that cannot be handled directly by systems like QuOnto. In order to make the query answerable, we can apply to it some kind of *semantic weakening*. The intuition behind the proposed weakening follows from the fact that all the system can do is try to find a counterexample to the given implication: if it fails, then the implication can be assumed to hold. Hence, the key step consists in substituting the knowledge operator $\mathbf{K}$ enclosing the universal quantification with the belief operator $\mathbf{B}$, i.e., $\mathbf{K}\varphi(\boldsymbol{x}) \wedge \mathbf{B}\,\forall \boldsymbol{y}(D(\boldsymbol{y}) \rightarrow \exists \boldsymbol{z}(\psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})))$.

The derived formula expresses the fact that all the system can do is test whether the universally quantified subformula is consistent with the knowledge base – i.e., the knowledge base does not entail the existence of a counterexample. Making use of the standard equivalence $\mathbf{B}\varphi \equiv \neg\mathbf{K}\neg\varphi$, the whole query can be rewritten as follows (in NNF), i.e., $\mathbf{K}\varphi(\boldsymbol{x}) \wedge \neg\mathbf{K}\,\exists \boldsymbol{y}(D(\boldsymbol{y}) \wedge \forall \boldsymbol{z}(\neg\psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})))$.

This newly introduced universal quantification poses the same problem with respect to query answering. We apply the same kind of weakening, replacing again the knowledge operator with the belief operator, i.e., $\mathbf{K}\varphi(\boldsymbol{x}) \wedge \neg\mathbf{K}\,\exists \boldsymbol{y}(D(\boldsymbol{y}) \wedge \mathbf{B}\,\forall \boldsymbol{z}(\neg\psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})))$, which is equivalent to the final query $\mathbf{K}\varphi(\boldsymbol{x}) \wedge \neg\mathbf{K}\,\exists \boldsymbol{y}(D(\boldsymbol{y}) \wedge \neg\mathbf{K}\,\exists \boldsymbol{z}(\psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})))$.

Queries in this form correspond to the `SELECT/FROM/WHERE/NOT_IN` fragment of SQL, which can be efficiently answered by systems for query answering over ontologies such as QuOnto, by combining the computation of certain answers with the computation of a set difference (cf. [10])

Going back to natural language, this process would correspond to answer a question like *"What is causing all the joint pain?"* with *"I am not aware of any joint pain that is not caused by the following diseases"*, when the system does retrieve some disease.

5 Conclusions and Future Work

The analysed corpora have shown that (*i*) the natural language questions a user would ask in order to access data in a DB are rather simple (compared to other forms of free text); (*ii*) often these real life questions are actually representable by CQs; (*iii*) given a natural language question, its corresponding query can be obtained by translating FOL representation outputs of state-of-the-art (wide coverage) parsers; (*iv*) questions that cannot be represented by CQs (or UCQs) could be handled via some form of semantic weakening, as illustrated by the case of universal quantifier. A similar method could be used to answer questions containing negation. Also, strategies for handling aggregations could be considered in the described framework. The observed data have also shown that quite often users would use plurals rather that an explicit universal quantifier. Hence, it would be interesting to check how the proposed method could be extended to properly capture their meaning. The work described in [17] could be taken as the starting point.

The output of the parser suggests an improvement to the application of the query answering tool to better meet the user expectations, viz. treat differently the presupposed knowledge in the question representation from the one in the body of the question. In the example considered in Section 4, "Who may use the Interlibrary Loan service?" the existence of an "interlibrary loan service" is presupposed. The system could verify the presupposition before answering the actual questions and give feedback to the user in case the presupposition is actually falsified. Evaluations of the proposed approach should be carried out.

Our work shows that the syntactic constructs present in user questions might in general not constitute a problem, and presents techniques to handle the more problematic cases. However, an important aspect that is largely left open in the present paper, while being addressed in other systems, e.g., Aqualog [16], is the problem of bridging the gap between the terminology of the user and the ontology terms and structure. To address this problem, on the one hand, lexical resources can be used to overcome differences in the terminology, by substituting user terms with appropriate synonyms present in the ontology. Notice also that QuOnto itself may expand a CQ into an UCQ by substituting a term with a collection of terms related to the first one through a (generalization) hierarchy. On the other hand, we are working on enriching our framework with mappings (in the style of local-as-view mappings used in data integration [13]) from the

ontology structures to suitable meaning representations, corresponding to the various ways in which users may query the ontology structures. We aim at constructing the mappings semi-automatically, by exploiting verbalizations of the ontology structures.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995.
2. Andrea Acciarri, Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Mattia Palmieri, and Riccardo Rosati. QuOnto: QUerying ONTOlogies. In *Proc. of AAAI 2005*.
3. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
4. Carlo Batini, Stefano Ceri, and Shamkant B. Navathe. *Conceptual Database Design, an Entity-Relationship Approach*. Benjamin and Cummings Publ. Co., 1992.
5. Raffaella Bernardi, Diego Calvanese, and Camilo Thorne. Expressing DL-Lite ontologies with controlled english. In *Proc. of DL 2007*.
6. Raffaella Bernardi, Diego Calvanese, and Camilo Thorne. Lite natural language. In *Proc. of the 7th Int. Workshop on Computational Semantics (IWCS-7)*, 2007.
7. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. Linking data to ontologies: The description logic DL-Lite$_A$. In *Proc. of OWLED 2006*.
8. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*.
9. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*.
10. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. EQL-Lite: Effective first-order query processing in description logics. In *Proc. of IJCAI 2007*.
11. James R. Curran, Stephen Clark, and Johan Bos. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proc. of the Demonstrations Session of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*.
12. Nicola Guarino. Formal ontology in information systems. In *Proc. of the Int. Conf. on Formal Ontology in Information Systems (FOIS'98)*. IOS Press, 1998.
13. Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*.
14. Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23, 1984.
15. Hector J. Levesque and Gerhard Lakemeyer. *The Logic of Knowledge Bases*. The MIT Press, 2001.
16. Vanessa Lopez, Michele Pasin, and Enrico Motta. AquaLog: An ontology-portable question answering system for the Semantic Web. In *Proc. of ESWC 2005*.
17. U. Schwertel. *Plural Semantics for Natural Language Understanding – A Computational Proof-Theoretic Approach*. PhD thesis, University of Zurich, 2004.