Querying Structured Knowledge Sources

Anette Frank, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit,
Berthold Crysmann, Brigitte Jörg and Ulrich Schäfer
German Research Center for Artificial Intelligence, DFKI
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
{frank,krieger,xu,uszkoreit,crysmann,joerg,uschaefer}@dfki.de

Abstract

We present an implemented approach for domain-restricted question answering from structured knowledge sources, based on robust semantic analysis in a hybrid NLP system architecture. We build on a lexical-semantic conceptual structure for question interpretation, which is interfaced with domain-specific concepts and properties in a structured knowledge base. Question interpretation involves a limited amount of domain-specific inferences and accounts for quantificational questions. We extract so-called *proto queries* from the linguistic representation, which provide partial constraints for answer extraction from the underlying knowledge sources. The search queries we construct from proto queries effectively constitute minimum spanning trees that restrict the possible answer candidates. Our approach naturally extends to multilingual question answering and has been developed as a prototype system for two application domains: the domain of Nobel prize winners and the domain of Language Technology, on the basis of the large ontology underlying the information portal LT World.

Introduction

The recent TREC and CLEF competitions have engendered significant progress both in the underlying research and the performance of practical Question Answering (QA) systems. While these competitions are focusing on open-domain textual QA on the basis of large document bases, there is increasing interest in QA in restricted domains. There are several motivations for this move. First, where open-domain QA exploits the wealth of information on the Web, it is also confronted with the problem of reliability: information on the Web may be contradictory, outdated, or utterly wrong. Second, the utilisation of formalised knowledge in a restricted domain can improve accuracy, since both questions and potential answers may be analysed w.r.t. to the knowledge base. Third, there is a market for accurate specialised information management solutions in both business intelligence and public administration.

QA systems for restricted domains may be designed to retrieve answers from unstructured data (free texts), semi-structured data (such as XML-annotated texts), or structured data (ontologies or data bases). Whenever structured data can be exploited, this option offers clear advantages over open text QA. However, despite a tendency towards deeper analysis, current techniques in QA are still knowledge-lean in exploiting data redundancy and paraphrasing techniques. That is, textual QA works on the assumption that the answer to a question is explicitly stated in some textual passage—which is typically not the case in restricted domains.

Question answering applied to restricted domains is therefore interesting in two important respects. Restricted domains tend to be small and stable enough to permit careful modelling in terms of structured knowledge bases that can serve as certified information sources. More importantly though, QA in restricted domains requires techniques that crucially differ from the techniques that are currently applied in open-domain textual QA. Since document sizes tend to be small, textual QA techniques cannot exploit data redundancy. Further, both in domain-restricted textual QA and QA from structured knowledge sources, we cannot expect the answer to a given question to be explicitly stated.

Since the question is the primary source of information to direct the search for the answer, a high-quality question analysis is of utmost importance in domain-restricted QA. Since the answer may not be literally stated in the underlying document or knowledge base, we need a semantic interpretation of the question that can be tightly connected to the domain knowledge sources and the process of answer extraction.

In this paper we present an approach to domain-restricted QA from structured knowledge sources that starts from these considerations. We focus on a high-quality deep linguistic analysis of the question, with conceptual-semantic interpretation of the question relative to the chosen application domain. Our approach extends to multilingual QA scenarios and provides a natural interface to the underlying knowledge bases, enabling flexible strategies for answer extraction.

In the following we give an overview of the architecture and the base components of the system. We introduce the main aspects of domain modelling for our two application domains: Nobel prizes and Language Technology. We then describe our approach to question analysis. We start from HPSG analyses of questions, which are enriched with a conceptual-semantic representation that can be further modified by domain-specific inference rules. Next, we describe the interface between question interpretation and domain

ontologies. We define a mapping between domain-specific concepts in the semantic representation of the question and corresponding concepts in the underlying domain ontology. This mapping is used to extract so-called *proto queries* from the semantic representation of the question. These abstract query patterns are translated to concrete data base or ontology query language constructs in the answer extraction phase. We go into the details of the concrete query construction and explain how a proto query can be mapped to SQL in the MySQL system and to SeRQL in the Sesame RDF framework. We conclude with a preliminary evaluation of our system and a short comparison to earlier approaches.

Architecture and Domain Modelling

Overall system architecture The QA system for structured knowledge sources described below is part of a general QA system architecture, the QUETAL[1] architecture. The hypothesis underlying the QUETAL architecture design is that QA systems perform best if they combine virtues of domain-specialised and open-domain QA, accessing structured, semi-structured, and unstructured knowledge bases. The core idea is that—instead of providing specific information portals (with system-specific user interfaces)—the QUETAL system provides a single and uniform natural language-based QA access to different information sources that exhibit different degrees of structuring.

The QUETAL architecture is hybrid in two senses. The question analysis is hybrid in that shallow and deep NLP are combined to yield both robustness and a rich semantic representation of questions. The answer document base is hybrid in that three types of information sources are employed: (i) *unstructured* text retrieved via Web-based or local full-text search engines and information retrieval systems, (ii) *semi-structured* text that has been enriched offline with IE and NLP techniques, and (iii) *structured* fact databases, e.g., ontologies and traditional relational databases containing domain-specific facts, relations, and concepts.[2]

In the overall QUETAL architecture, the QA process starts with linguistic analysis and a subsequent interpretation of the question. After a question type has been identified together with the expected answer type, one (or more than one) information source is selected to retrieve answer candidates. From these, an answer is prepared. As QUETAL supports crosslingual QA, the architecture integrates intermediate translation stages (Neumann & Sacaleanu, 2003).

Architecture for domain-restricted QA The architecture for domain-restricted QA from structured knowledge sources (Figure 1) is embedded in the general QUETAL architecture. A question is linguistically analysed by the Heart-of-Gold (HoG) NLP architecture, which flexibly integrates deep and shallow NLP components (Callmeier *et al.*, 2004), e.g., PoS tagger, named entity recognition and HPSG parser. The semantic representations generated by the HoG are then interpreted and a query object is generated that contains a proto query. This proto query can be viewed as an implementation-independent, 'higher-level' representation of a data base or ontology query. From this, an instance of a specific data base or ontology query is constructed. From the result(s) returned by the queried information source, an answer object is generated which forms the basis for subsequent natural language answer generation.

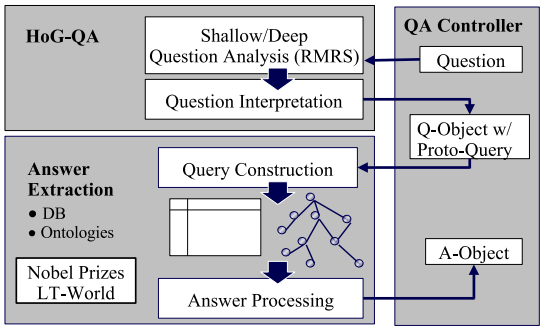

Figure 1: QA from structured knowledge sources.

Domain Modelling and Inference Services

The Nobel Prize Ontology and Data Base The domain ontology plays a crucial role in our approach. It is used as the interface between question analysis, answer extraction and knowledge database design. We chose *Nobel Prize* as our initial subject domain, since it is a domain for which both complete records of all awarded prizes in structured formats and thousands of free texts about awards and laureates can be found on the web.[3] Furthermore the data are manageable in size, authoritative and can be used as our gold-standard for evaluation of the QA task. Here we focus on the exploitation of the structured data for QA.[4]

We started our specification with an existing general ontology as reference. Two sources have been selected: the knowledge-engineering-based top-level ontology SUMO (Niles & Pease, 2001) and its mid-level specification MILO (Niles & Terry, 2004), and on the other hand the structured thesaurus WordNet (Miller *et al.*, 1993). Since there is a mapping between the artificial concepts in SUMO and the word senses in WordNet (Niles & Pease, 2003), we decided to choose the SUMO ontology as our backbone and define sub-concepts by referring to the mapping between SUMO concepts and WordNet word senses.

The main concepts in our application domain are `prize`, `laureate`, `prize-area`, incl. domain-independent general concepts, such as `person` or `organization`. Figure 2 lists some of the mappings between domain concepts and SUMO concepts. `laureate` corresponds to the SUMO concept `cognitiveAgent`, inheriting therefore its two subconcepts `human` and `organization`. Most subconcepts of the concept `prize-area`, except for `Peace`, are

[1] See the QUETAL project homepage at http://quetal.dfki.de.

[2] See Neumann & Sacaleanu (2003, 2004) for textual QA in QUETAL. Semi-structured text is not yet covered by the current system, but will be addressed at a later stage of the project.

[3] Peace Nobel Prizes (as many other prizes) can be also awarded to organisations and not just to persons.

[4] The utilisation of structured data as seed data for learning IE patterns is described in Uszkoreit & Xu (2005).

subconcepts of the general concept `fieldOfStudy`, e.g., `Chemistry`. Each concept is further specified by its attributes. E.g., `person` is assigned the attributes `firstname` and `surname`. The concepts are organized via hierarchical relations. In addition to the domain-specific relations, such as `nobel-prize-nomination`, we also model some general relations like `person-affiliation`.

type	domain	SUMO
entity	prize	award, ...
entity	laureate	cognitiveAgent
entity	person	human
entity	organization	group
entity	prize-area	fieldOfStudy
event	nobel-prize-winning	unilateralGetting
event	nobel-prize-nomination	declaring, deciding

Figure 2: Mappings between domain and SUMO concepts.

The LT WORLD Ontology and Data Base As our second scenario for domain-restricted QA, we have chosen the Language Technology World information portal (http://www.lt-world.org). LT WORLD is an ontology-based virtual information center on Human Language Technology, providing information about people, products, resources, projects, and organisations. The service is free and is provided by the German Research Center for Artificial Intelligence (DFKI) to the R&D community, potential users of language technologies, students and other interested parties.

Most of the concepts referred to in LT WORLD have a direct counterpart in its underlying ontology (Uszkoreit, Jörg, & Erbach 2003). For example, people actively working in Language Technology are modelled as instances of the class/concept `Active_Person`. This concept is a subclass of `Players_and_Teams` which has further subclasses such as `Projects` or `Organisations`. The fact that people coordinate projects is represented by the property/role `hasCoordinated` which maps from `People` $\cup$ `Organisations` (domain) to `Projects` (range).

The original ontology behind LT WORLD made use of RDF and RDF Schema. The ontology has recently been ported to the Web ontology language OWL, the new emerging language for the Semantic Web that originates from the DAML+OIL standardisation. OWL still makes use of constructs from RDF and RDFS such as `rdf:resource` or `rdfs:subClassOf`, but its two important variants OWL Lite and OWL DL restrict the expressive power of RDFS, thereby ensuring decidability. What makes OWL unique (as compared to RDFS) is the fact that it can describe resources in more detail and that it comes with a well-defined model-theoretical semantics, inherited from description logic (Baader *et al.* 2003). The description logic background furthermore provides automated reasoning support such as consistency checking of the TBox and the ABox, subsumption checking, etc. Even though the least expressive variant of OWL, viz., OWL Lite has an EXPTIME worst-case complexity, optimised implementations based on tableaux algorithms are known (Horrocks, Sattler, & Tobies 2000), which actually work well for most practical cases and have been implemented in a few systems (see below).

Inference Services The new LT-WORLD ontology was developed using the OWL plugin of the Protégé knowledge base editor (Knublauch, Musen, & Rector 2004). This version of Protégé comes with partial OWL Lite support by means of the Jena Semantic Web framework (Reynolds 2004).

The latest version of LT WORLD consists of more than 600 concepts, 200 properties, and 17,000 instances. From an RDF point of view, we have more than 400,000 unique triples. It was confirmed by several tests that querying the ontology through Jena (using RDQL) will take too much time, the main reason being that the OWL reasoner uses the rule engines in Jena for all kinds of inference, especially when querying for instances of a specific concept, meaning that we are not only interested in the direct instances, but also in instances of subconcepts of this concept.

We then experiment with implemented description logic systems providing OWL support. The FaCT system (Horrocks 1998) seemed a good candidate but does not provide much ABox support, which is vital for us (17,000 instances) and other Semantic Web applications. For a long time, we were using the Racer system (Haarslev & Möller 2003). Racer helped to uncover many modelling errors in LT WORLD which fell through the "grid" of Protégé/Jena.

During ontology development, the number of instances grew and the complexity of instance descriptions raised. Unfortunately, it turned out that the ABox does not scale up well: 5,000 instances is the maximum that Racer can handle when complex queries on parts of LT WORLD are processed. TBox reasoning (as is the case for the FaCT system) is fine, though.

Therefore we moved to RDF data base systems (see Guo, Pan, & Heflin 2004). Even though we are developing OWL ontologies (LT WORLD) with Protégé, the information that is stored on disk is still RDF on the syntactic level. We are thus interested in RDF DB systems which make sense of the semantics of OWL and RDFS constructs such as `rdfs:subClassOf`.

We solved the scalability problem by porting the ontology to Sesame (http://www.openrdf.org/), an open-source middleware framework for storing and retrieving RDF data. Sesame partially supports the semantics of RDFS and OWL constructs via entailment rules that compute "missing" RDF triples in a forward-chaining style at compile time. LT WORLD originally consists of about 200,000 RDF triples, resulting from the 17,000 instances. The closure computation adds almost the same number of new entailed triples, so that Sesame must handle in the end 404,767 statements. Closure computation is fast and takes only a few seconds of real time on a mid-size Linux machine.

Since sets of RDF statements represent RDF graphs, querying information in an RDF framework means to specify path expressions. Sesame comes with a powerful query language, SeRQL. It includes: (i) generalised path expressions, including multi-value nodes and branches, (ii) a restricted form of disjunction through optional matching, (iii) existential quantification over predicates, and (iv) Boolean constraints. We will see below that all of the above features, even predicate quantification (which gives us some decid-

able second-order expressiveness here) are needed to arrive at a SeRQL query to correctly constrain the object retrieval from LT WORLD.

Sesame has been tested with several 100,000 instances (Guo, Pan, & Heflin 2004). Its storage model can be configured by either using an existing data base system (e.g., PostgreSQL, MySQL, or Oracle) or by going for a pure in-memory representation of the data. We have opted for the latter version in LT WORLD to speed up query time. The system scales up very well, giving satisfactory performance. The memory footprint ranges from 70 to 200 MBytes.

Question Analysis and Interpretation

Hybrid NLP for Question Analysis For question analysis we employ deep HPSG syntactic and semantic analysis. HPSG parsing is efficiently performed using the PET parser (Callmeier, 2000). For increased robustness, the parser is embedded in an NLP processing platform for integrated shallow and deep analysis, the Heart-of-Gold (HoG) architecture (Callmeier *et al.*, 2004). Within this architecture, HPSG parsing is seamlessly integrated with the Information Extraction system SProUT (Drożdżyński *et al.*, 2004). SProUT performs named entity recognition on the basis of unification-based finite-state transduction rules and gazetteers. It provides structured representations both for general named entity classes and domain-specific terms and named entities. The Heart-of-Gold architecture is designed for integration of NLP components for multiple languages.

In our QA application we are using wide-coverage HPSG grammars for English (Baldwin *et al.*, 2004) and German (Crysmann *et al.*, 2002). Both grammars are integrated with shallow NE recognition. HPSG parsing delivers semantic representations in the formalism of Minimal Recursion Semantics (MRS) (Copestake *et al.*, 2005). MRS is designed for underspecification of scope ambiguities, using a flat, non-recursive representation format. A variant of MRS, Robust Minimal Recursion Semantics (RMRS) has recently been designed in Copestake (2003), facilitating the integration of deep semantic structures with *partial* semantic structures, as produced by more shallow NLP components, such as chunkers or robust PCFGs. Within the HoG architecture, RMRS constitutes the interchange format for all the different NLP components, including named entity recognition.

Figure 3 displays an RMRS produced by HPSG parsing, along with RMRS representations from the NE recognition. The RMRSs of the SProUT NER component are highly structured, IE-like NE representations, decomposing, e.g., a person name into *surname* and *given_name* relations. The identified NE classes are further mapped to coarse-grained HPSG NE-types (cf. *named_abb_rel*), which are directly delivered to the HPSG parser to enhance robustness.

Question interpretation RMRS representations of questions are marked by way of a semantic relation *int_m_rel*, for interrogative message type. In wh-questions, interrogative pronouns introduce sortal relations for the queried constituent, such as *person_rel* (who), *time_rel* (when), etc. For wh-phrases with nominal heads, the semantic relation introduced by the nominal constrains the semantic type of the queried constituent (cf. *year_rel* in Figure 3). Yes/no questions are simply marked as interrogative by *int_m_rel*. Imperative sentences such as "*List all persons who work on IE.*" introduce an imperative message type *imp_m_rel*.

While the RMRS representation of questions encodes important semantic information for question interpretation, such as message type and the marking of wh-phrases, the representation must be further enriched in order to derive concise queries for answer extraction from structured knowledge sources. The minimal information we need to identify is the *queried variable (q_var)* in the RMRS logical form. We further want to determine sortal information for the queried variable, that is, the *expected answer type (EAT)*. This information is usually employed in textual QA systems, but can also be effectively used for answer extraction from structured knowledge sources, as will be discussed below.

The semantic interpretation process is driven by a term rewriting system (Crouch, 2005) that takes as input the RMRS analyses provided by "general purpose" HPSG parsing, along with the RMRS for recognised named entities. We apply interpretation rules that refer to (partial) argument structures in the RMRS in order to identify and mark the queried variable *q_var* in the logical form of the RMRS. We further determine the ontological type of the queried variable, which provides important semantic constraints for answer extraction. Pronominal wh-phrases introduce a semantic relation for the queried variable, such as *person*, *location*, or *reason*. For these general concepts, as well as for wh-phrases headed by common nouns, we perform a concept lookup, either by selecting an ontological class from SUMO, by way of its WordNet lookup facility, or else by directly mapping the lexeme to its corresponding domain concept.[5] For the example displayed in Figure 3, this yields the additional semantic constraints: *q_var(x10)* and *EAT(x10, 'year')*, with *x10* the variable corresponding to *"year"*. These are encoded in the RMRS by way of elementary predications (EPs) *q_focus* and *EAT_rel*, as seen below. In both EPs the value of ARG0 identifies the queried variable. *EAT_rel* in addition encodes the feature SORT, which takes as value the sortal type determined for the queried variable.

$$\begin{bmatrix} \text{REL} & \textit{q\_focus} \\ \text{ARG0} & \textit{x10} \end{bmatrix} \begin{bmatrix} \text{REL} & \textit{EAT\_rel} \\ \text{ARG0} & \textit{x10} \\ \text{SORT} & \textit{year} \end{bmatrix}$$

The RMRS as a logical form now explicitly encodes the queried variable, with ontological restrictions as sortal constraints. The remaining EPs define relational constraints on the requested information: in our example we are looking for the time when a Nobel prize was won by a person named "*Nadine Gordimer*", where the area was "*Literature*". These are the key relational constraints that need to be satisfied when retrieving the answer from the knowledge base.

It is the task of question interpretation to identify these relational constraints on the basis of the semantic representation of the question. These constraints can then be translated to a search query in the formal query language of the

[5] In our current prototype system concept lookup is encoded manually. In future work we will experiment with automated methods for concept lookup similar to (Burchardt, Erk, & Frank, 2005).

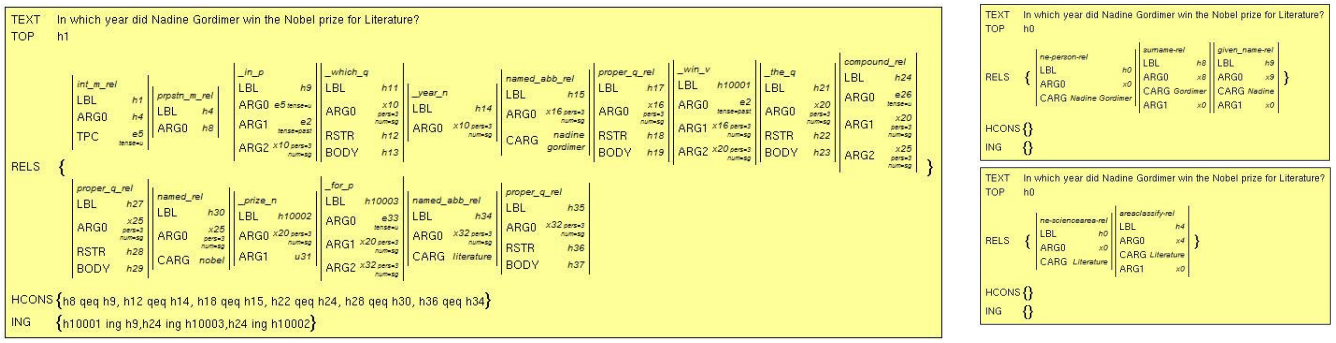

Figure 3: RMRS of HPSG analysis (left) and SProUT NE recognition (right).

underlying knowledge base. We perform this task in three steps: We first project a conceptual, frame semantic representation from the RMRS of the question. On the basis of a pre-defined set of domain-relevant frames and roles, we extract from this representation relational constraints for query construction. These relational constraints, defined in a so-called *proto query*, are then translated to a search query with corresponding domain-specific concepts and properties, to retrieve the requested information from the knowledge base.

The motivation for this approach is twofold: First, the projection of a lexico-conceptual structure yields a normalisation of the semantic representation that naturally accounts for linguistic variants, or paraphrases of questions. It further constitutes a natural approach for multilingual and crosslingual QA in restricted domains. Second, by defining a set of domain-relevant frames and roles, we can establish a modular interface between the linguistically determined conceptual-semantic representation of the question and the concepts of the underlying knowledge bases. On the basis of a mapping between domain-relevant frames and corresponding concepts in the domain ontologies, we efficiently identify and extract the domain-relevant constraints from the semantic representation of the question. These constraints are encoded in the proto query that is handed over to the answer extraction process. Evidently, the use of abstract proto queries gives us a clean interface that abstracts away from the syntax and functionality of the backend query languages.

Mapping RMRS to conceptual representations To obtain a conceptual semantic representation for the question, we project a frame semantic representation from the RMRS. Frame Semantics is pursued in the FrameNet project (Baker, Fillmore, & Lowe, 1998). FrameNet is building a database of frame semantic descriptions for English verbs, nouns, and adjectives, where a *frame* models a conceptual situation with concept-specific *roles* that identify the participants in the situation. In addition, FrameNet defines peripheral or extra-thematic roles, such as MANNER and TIME (cf. example (1)).

(1) [Grant $_{RECIPIENT}$] *obtained*$_{GETTING}$ [his first degree $_{THEME}$] [by attending evening classes at Queen Mary College, London $_{MANNER}$].

Due to their design as conceptual semantic structures, frames account very naturally for the normalisation of paraphrases. For illustration, consider the semantically equivalent paraphrases in (2.a), which are typical expressions for requesting information about Nobel prizes. HPSG semantic representations in terms of (R)MRS are tailored to account for structural semantic properties such as quantifier scoping and predicate-argument structure, and thus still reflect the various different argument structures involved, as illustrated in (2.b). Following related work in Frank & Erk (2004), we enrich the RMRS representation with a frame semantic projection, by mapping the different argument structures to their corresponding frame structure, which states the name of the frame and its roles. An example of a frame assignment rule is given in (2.c). (2.d) displays the uniform frame semantic representation obtained from the RMRS variants in (2.b).

(2) a. *(win/ be awarded/ obtain/ get/ be winner of) a prize*

b. Different argument structures in RMRS

$$\left(\begin{bmatrix} \text{REL} & \textit{win/get/obtain} \\ \text{ARG0} & \textit{e1} \\ \text{ARG1} & \textit{x1} \\ \text{ARG2} & \textit{x2} \end{bmatrix} \vee \begin{bmatrix} \text{REL} & \textit{award} \\ \text{ARG0} & \textit{e1} \\ \text{ARG1} & \textit{u1} \\ \text{ARG2} & \textit{x2} \\ \text{ARG3} & \textit{x1} \end{bmatrix} \vee \begin{bmatrix} \text{REL} & \textit{winner} \\ \text{ARG0} & \textit{x1} \\ \text{ARG1} & \textit{x2} \end{bmatrix} \right) \begin{bmatrix} \text{REL} & \textit{prize} \\ \text{ARG0} & \textit{x2} \end{bmatrix}$$

c. RMRS-based frame assignment rules

$$\begin{bmatrix} \text{REL} & \textit{win} \\ \text{ARG0} & \textit{e1} \\ \text{ARG1} & \textit{x1} \\ \text{ARG2} & \textit{x2} \end{bmatrix} \begin{bmatrix} \text{REL} & \textit{prize} \\ \text{ARG0} & \textit{x2} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \text{GETTING} & \textit{e1} \\ \text{SOURCE} & \textit{u1} \\ \text{THEME} & \textit{x2} \\ \text{RECIPIENT} & \textit{x1} \end{bmatrix} \begin{bmatrix} \text{AWARD} & \textit{x2} \\ \text{LAUREATE} & \textit{x1} \\ \text{DOMAIN} & \textit{u3} \end{bmatrix}$$

d. Conceptual (frame semantic) representation

$$\begin{bmatrix} \text{GETTING} & \textit{e1} \\ \text{SOURCE} & \textit{u1} \\ \text{THEME} & \textit{x2} \\ \text{RECIPIENT} & \textit{x1} \end{bmatrix} \begin{bmatrix} \text{AWARD} & \textit{x2} \\ \text{LAUREATE} & \textit{x1} \\ \text{DOMAIN} & \textit{u3} \end{bmatrix}$$

Inferences The frame semantic representations can be further enriched by applying simple inference rules.

Frames define a number of *core* frame elements, which can be understood to be existentially quantified even in cases where the role is not overtly realised. Thus, we can introduce non-instantiated argument variables for unexpressed frame elements (e.g., SOURCE in the GETTING frame in (2)).

We further define domain-specific inference rules that can be derived from inherent semantic relations between frames. The rule in (3), for example, defines that whenever there is an AWARD frame where the role LAUREATE refers to some variable in the logical form, this variable in turn projects a frame LAUREATE, with its own specific core semantic roles, such as NAME, etc. By application of rule (3) we extend the frame representation in (2.d) with an additional frame LAUREATE, bound to the variable *x1*. Inferences of this type turn out to be very effective to obtain maximally connected frame semantic representations.

(3) $\begin{bmatrix}\text{AWARD} & x2\\ \text{LAUREATE} & x1\end{bmatrix} \Rightarrow \begin{bmatrix}\text{LAUREATE} & x1\\ \text{NAME} & u5\end{bmatrix}$

In addition we define a number of inference rules that are crucial to bridge mismatches between the conceptual representation that is generated from the linguistic structure and the conceptual model structure of the underlying knowledge base. In example (4), the linguistic analysis of the question renders a frame semantic structure where the temporal modifier of the winning event is mapped to the TIME role of the GETTING frame. The domain ontology, however, does not encode a concept that corresponds to the GETTING frame. Instead, this temporal information is encoded as a property of the award. Mismatches of this type can be accounted for by inference rules, as in (4.c). The rule states that if there is a GETTING frame where the THEME is an AWARD, and its TIME role refers to some temporal variable, the AWARD frame inherits the value of this TIME role. This corresponds to an inference according to which the time of receiving an award is equal to the time (attribute) of the award.

(4) a. *When did Marie Curie win the Physics prize?*

b. Partial RMRS and frame semantic projection

$\begin{bmatrix}\text{REL} & \textit{sciencearea}\\ \text{ARG0} & x3\\ \text{CARG} & \textit{physics}\end{bmatrix}$ $\begin{bmatrix}\text{REL} & \textit{person}\\ \text{ARG0} & x1\\ \text{CARG} & \textit{Marie Curie}\end{bmatrix}$

$\begin{bmatrix}\text{REL} & \textit{q\_focus}\\ \text{ARG0} & t1\end{bmatrix}$ $\begin{bmatrix}\text{REL} & \textit{eat\_rel}\\ \text{ARG0} & t1\\ \text{SORT} & \textit{time}\end{bmatrix}$ $\begin{bmatrix}\text{GETTING} & e1\\ \text{SOURCE} & u1\\ \text{THEME} & x2\\ \text{RECIPIENT} & x1\\ \text{TIME} & t1\end{bmatrix}$

$\begin{bmatrix}\text{AWARD} & x2\\ \text{LAUREATE} & x1\\ \text{DOMAIN} & x3\\ \text{TIME} & u2\end{bmatrix}$ $\begin{bmatrix}\text{LAUREATE} & x1\\ \text{NAME} & \textit{Marie Curie}\\ \text{AFFILIATION} & u6\end{bmatrix}$

c. Inference rule

$\begin{bmatrix}\text{GETTING} & e1\\ \text{THEME} & x2\\ \text{TIME} & t1\end{bmatrix}$ $\begin{bmatrix}\text{AWARD} & x2\\ \text{TIME} & u2\end{bmatrix} \Rightarrow \begin{bmatrix}\text{AWARD} & x2\\ \text{TIME} & t1\end{bmatrix}$

Inferences of this type allow us to map linguistically determined frame semantic represention to the structure of the domain ontology, and thus, to extract appropriate query constraints for answer extraction. This will be discussed below.

Multilinguality Our approach to question interpretation naturally extends to multilingual and crosslingual QA scenarios. Since frames are defined as conceptual structures, they are to a large extent language independent. Thus, question interpretation in terms of a frame semantic representation effectively implements an interlingua approach for QA.

In our NLP architecture, HPSG grammars for different languages—here, German and English—provide semantic structures in the uniform formalism (R)MRS. The language-specific relations in these semantic forms are translated by language- and lexeme-specific frame projection rules to a common, language-independent frame semantic representation. The remaining parts of the question interpretation and answer extraction processes are then uniform across languages. Both the domain-specific inference rules and the rules for the extraction of proto queries uniformly operate on the language-independent frame semantic representations, thus they are applied to the same type of intermediate structures in question interpretation, irrespective of whether they were produced by German or English HPSG grammars.

For crosslingual QA from structured data we perform term translation for instances (i.e., named entities) and domain-specific terms of the knowledge base that can appear as values in search queries constructed from the question.

Interfacing Question Interpretation and Domain Ontologies

Mapping frames to domain concepts For the extraction of queries to the domain knowledge bases we define a set of *domain-relevant* frames and roles, for which the domain models specify corresponding concepts and properties. This is illustrated for the frame AWARD from Nobel prizes.

```
frame_role2nobel_domain(award,laureate,_,_).
frame_role2nobel_domain(award,domain,_,_).
```

Besides *identification* of domain-relevant frames, we can further specify a *mapping* to corresponding concepts in the underlying ontology. This option we pursued in the LT World scenario. Here the clauses additionally state target concepts and properties in the LT World ontology.

```
frame_role2ltw_domain(project,leader,
                      ActiveProject,coordinatedBy).
frame_role2ltw_domain(project,name,
                      ActiveProject,projectNameVariant).
```

On the basis of this information we extract domain-relevant concepts from the semantic representation of the question, and turn them into abstract query terms that are then translated to concrete data base or ontology queries.

Construction of Proto Queries A basic distinction for the construction of structured query terms is the distinction between queried vs. constraining concepts. For the extraction of queried concepts in (5.a), we select those domain-relevant frames and/or roles that correspond to the queried variable in the logical form, represented as ARG0 of the *q_focus* relation. We further extract the ontological restrictions encoded as the expected answer type in *EAT_rel*. In (5.b) we extract all remaining (i.e., non-queried) domain-relevant frames and roles, which provide additional constraints on the queried concepts. Again, we extract ontological restrictions, here in terms of their named entity type, as encoded by the RMRS structures provided by NE recognition in the HoG.

(5) a.
```
q_focus(Y), frame(Frame,X), fe(Role,X,Y),
frame_role2domain(Frame,Role,_,_), EAT_rel(Y,Sort)
=> select_cond(Qid,Frame,Role,Sort).
```

b.
```
-q_focus(Y), frame(Frame,X), fe(Role,X,Y),
frame_role2domain(Frame,Role,_,_), ne_type(Y,NE)
=> where_cond(Qid,Frame,Role,NE).
```

By this method we extract so-called *proto queries* from the frame semantic structures, as illustrated in (6) below.[6]

(6) a. *In which areas did Marie Curie win a Nobel prize?*

b. Question interpretation

$$\begin{bmatrix}\text{REL} & \textit{q\_focus}\\ \text{ARG0} & \textit{x10}\end{bmatrix}\begin{bmatrix}\text{REL} & \textit{EAT\_rel}\\ \text{ARG0} & \textit{x10}\\ \text{SORT} & \textit{FieldofStudy}\end{bmatrix}$$

$$\begin{bmatrix}\text{REL} & \textit{person}\\ \text{ARG0} & \textit{x17}\\ \text{CARG} & \textit{Marie Curie}\end{bmatrix}\begin{bmatrix}\text{GETTING} & \textit{e2}\\ \text{THEME} & \textit{x21}\\ \text{RECIPIENT} & \textit{x17}\end{bmatrix}\begin{bmatrix}\text{AWARD} & \textit{x21}\\ \text{LAUREATE} & \textit{x17}\\ \text{DOMAIN} & \textit{x10}\end{bmatrix}$$

$$\begin{bmatrix}\text{LAUREATE} & \textit{x17}\end{bmatrix}$$

c. Proto Query

```
<PROTO-QUERY id="1">
  <SELECT-COND qid="0" rel="award" attr="domain"
               sort="FieldofStudy">
  <WHERE-COND qid="0" rel="award" attr="laureate"
              netype="person" val="Marie Curie">
</PROTO-QUERY>
```

Quantificational Questions QA from structured knowledge bases is particularly well suited to answer questions where the answer is not explicitly represented in the document or knowledge base, but must instead be inferred from the available basic information. Prime examples are cardinality, quantificational or comparative questions, as in (7).

(7) a. *How many researchers have won a Nobel prize for Physics before 1911?*

b. *Which institution has published most papers between 2000 and 2004?*

c. *Which nation has won more Nobel prizes in Physics than the U.S.?*

To account for quantificational aspects, we employ special proto query conditions `OP-COND` and `QUANT-COND`. These constructs go beyond the formal power of most data base query languages, but can be translated to special post-processing operations in the answer extraction phase.

The quantificational conditions are determined by the semantic representation of the question. Cardinality questions (cf. (7.a)) are marked by operators like *how many* that range over the queried variable. For such configurations we generate, in the proto query, a condition `OP-COND` that specifies the operator relation `op-rel` that corresponds to the semantics of the quantifier. Since the quantification ranges over the queried variable, the domain of computation is defined as the answer for the sub-query for the queried variable.

In quantificational and comparative questions (cf. (7.b,c) and Figure 4) the quantification ranges over a non-queried variable. In these cases we perform query decomposition. We compute conditions for a base query that retrieves instances for the domain of quantification (*nation* in Figure 4). The quantifier condition `QUANT-COND` defines that for each instance in this domain we perform a sub-query for the queried variable and the non-queried relational constraints (select and where conditions), by referring to each instance of the quantifier domain. An operator condition encodes the quantifier-specific relation (e.g., `max-card` for *most*) that is to be computed over the retrieved data records.

[6] Proto queries may be complex, i.e., may be decomposed into individual sub-queries with specially marked dependencies. Therefore, all conditions that pertain to a single sub-query are marked by a common index (`qid`).

```
<PROTO-QUERY id="8">
 <SELECT-COND qid="0" rel="laureate" attr="origin"
              sort="?"/>
 <QUANT-COND qid="1" quantrel="foreach" domain="answer"
             domain-id="0"/>
 <SELECT-COND qid="1" rel="award" attr="" sort=""/>
 <WHERE-COND qid="1" rel="laureate" attr="origin"
             valfunc="answer_of" valarg="0"/>
 <WHERE-COND qid="1" rel="award" attr="domain"
             val="Physics"/>
 <OP-COND oprel="max_card" domain="answer" domain-id="1"/>
</PROTO-QUERY>
```

Figure 4: Proto query: *Which nation has won most Nobel prizes for Physics?*

Extraction of concept-relating paths The rules for extraction of proto queries (5.a,b) only consider local frames and roles to define relational constraints for query extraction. Thus, the concepts that appear in the individual select and where conditions may be unconnected, as in Figure 4. The frame semantic representation of the question does, however, often specify connecting paths between these frames and roles. In part, these connections are introduced by the linguistic structure, in part by domain-specific inferences. We extract such connecting paths and record them as a `path` attribute in the proto query conditions. This path information is used in the answer extraction phase to further specify the connections between the partial search constraints.

Answer Extraction

Answer Extraction from Nobel Prize Data Base The instances of domain relations are stored in the relational database MySQL. We store the Nobel prize winners in two separate tables: one for persons and one for organisations, since the two concepts *person* and *organization* are associated with different attributes. In the following examples, we call these *winner-person* and *winner-organization*.

The first step to be taken in answer extraction is to translate proto queries provided by question interpretation to SQL queries. Proto queries identify: (i) the answer type concept, which corresponds to the value of the SQL `SELECT` command, (ii) additional concepts and their values, which constrain the answer type value (these concepts will fill the SQL `WHERE` conditions), and (iii) dependencies between elementary questions, if a question is complex and needs to be decomposed into subqueries.

For example, for a simple fact-based question such as *Who won the Nobel Prize in Chemistry in 2000?* question analysis returns the following proto query:

```
<PROTO-QUERY id="q13" type="sql">
  <SELECT-COND rel="award" attr="laureate" />
  <WHERE-COND rel="award" attr="domain" val="Chemistry"/>
  <WHERE-COND rel="award" attr="time" val="2000"/>
</PROTO-QUERY>
```

The task of SQL query translation is to first identify the

tables where the requested concepts can be found, and second, the relevant table fields which can match the values given in the proto query. We have defined mapping rules between FrameNet frames and their roles and their corresponding data base tables and their fields. In a special field `event-dependent` we further mark concepts that are events. Below we list some examples of table entries:

Relation	Attr	val-concept	DBTable	DBField	event-dependent
award	laureate	person	winner-person	name	yes
award	laureate	organization	winner-organization	name	yes
award	domain	prize-area	winner-person	area	no
award	domain	prize-area	winner-organization	area	no
award	time	date time	winner-person	year	yes
award	time	date time	winner-organization	year	yes

The `SELECT-COND` in the example above only mentions the frame semantic `rel` and `attr` attributes `award` and `laureate`. Yet, there is no direct mapping to a table for `laureate`. In such cases we make use of our ontology and discover that `laureate` corresponds to `cognitiveAgent` which has two subconcepts: `human` and `group`. Their corresponding domain concepts are `person` and `organization`. We thus expand `laureate` to `person` and `organization` and find their corresponding tables. In the same way, we identify the tables for the `WHERE-COND`. In this example, `SELECT-` and `WHERE-COND` require access to the same tables. Thus, we generate the following two SQL queries:

```
SELECT name FROM winner-person
  WHERE year="2000" AND area="chemistry"
SELECT name FROM winner-organization
  WHERE year="2000" AND area="chemistry"
```

The final answer is obtained by merging their results.

While the example just considered involves concept expansion, we also perform concept disambiguation. This is illustrated by the example *In which year did Nadine Gordimer win the Nobel prize for Literature?*, with the proto query

```
<PROTO-QUERY id="1">
  <SELECT-COND rel="award" attr="time" sort="Year" />
  <WHERE-COND rel="award" attr="domain"
              netype="prize-area" val="Literature" />
  <WHERE-COND rel="award" attr="laureate"
              netype="person" val="Nadine Gordimer" />
</PROTO-QUERY>
```

Again, both the `SELECT-COND` and the first `WHERE-COND` identify the two tables `winner-person` and `winner-organization`. However, in the second `WHERE-COND`, the linguistic analysis recognises that the entity type of `laureate` is `person`. We can use this information for table disambiguation and choose the table `winner-person`. The SQL query for this question then is:

```
SELECT year FROM winner-person
  WHERE area="Literature" AND name="Nadine Gordimer"
```

Finally, we distinguish queried entities that are independent of individual prize winning events from event-dependent entities. Consider the two questions:

(8) *How many areas are there for the Nobel Prize?*
(9) *How many Nobel Prize winners has France produced?*

In the first case, every area in which a person or organisation has won a Nobel prize is only counted once. For answering the second question, we could also count every person once, even if the person has been awarded two prizes, such as, e.g., Marie Curie. We thus decided to make the cardinality of recipients event-dependent.

Thus, the answer to the first question will be: *Six areas*, although all areas occur more than once in award-winning events. We treat *area* as event-independent, generating an SQL query with a `DISTINCT` condition:

```
SELECT DISTINCT area FROM table
```

The answer to the second question will be: *Three winners: Marie Curie (2) and Pierre Curie (1)*. Here, the person in the *award* relation is handled as event-dependent. In this case we generate the SQL query

```
SELECT person FROM table WHERE country="France"
```

Answer Extraction from the LT World Ontology In this section, we show how a proto query can be mapped to an expression in the query language SeRQL of Sesame.

Based on the mapping from domain-specific frames and roles in the proto query conditions to domain concepts and properties, we first perform a translation of the values of `rel` and `attr` attributes to the corresponding domain concepts and attributes of the LT World ontology. Thus, each relation (value of `rel`) now denotes a concept in the ontology and each attribute (value of `attr`) denotes an OWL property.

In a SeRQL query, instances of a concept are identified by variables in the subject position of an RDF triple. The concept itself is stated in the object position, and subject and object are connected by `rdf:type`—this is exactly the way how instances of a specific concept are represented in the RDF base of Sesame. For example,

```
<SELECT-COND rel="Organisations" attr="locatedIn" ... />
```

leads to the introduction of the following RDF triple (`_r` is a fresh variable, `ltw` the LT WORLD namespace):

```
{_r} rdf:type {ltw:Organisations}
```

Since attributes like `locatedIn` refer to properties of a concept, we obtain a further triple:

```
{_r} ltw:locatedIn {_q}
```

The property `locatedIn` connects instances of the main concept `Organisations` via the root variable `_r` with the queried information. The queried information is bound to a new question variable `_q` that will be returned. It is marked by the SELECT clause in a SeRQL query:

```
SELECT {_q}
FROM {_r} rdf:type {ltw:Organisations},
     {_r} ltw:locatedIn {_q} ...
```

In Figure 5 we give an overview of the main principles of the transformation from proto queries to SeRQL queries.

In order to illustrate the transformation principles, let us consider the question *Who is working in the Quetal project?*, with its (simplified) proto query that contains a `SELECT` and a single `WHERE` condition:

```
<PROTO-QUERY>
  <SELECT-COND rel="Active_Person" attr="">
  <WHERE-COND rel="Active_Project" attr="projectName"
                                   val="Quetal">
</PROTO-QUERY>
```

Given this proto query, we generate the SeRQL query

(1) for each SELECT-COND and WHERE-COND
  – each relation denotes a concept
  – each attribute denotes a property
  – each unique relation introduces a new *root* variable
(2) each SELECT-COND introduces a new *query* variable
(3) each WHERE-COND introduces a new *local* variable
(4) guarantee that the RDF triples form a connected graph
  – if path constraints are specified, link the root variables
  – otherwise, introduce new *property* vars linking the roots
(5) finally apply OP-COND to the result table

Figure 5: Transforming proto queries into SeRQL queries.

```
SELECT DISTINCT _q0
FROM {_r1} rdf:type {ltw:Active_Person},
     {_r2} rdf:type {ltw:Active_Project},
     {_r1} ltw:name {_qo},
     {_r2} ltw:projectName {_l3},
     [ {_r1} _p4 {_r2} ],
     [ {_r2} _p4 {_r1} ]
WHERE (NOT (_p4 = NULL) AND (_p5 = NULL)) AND
      (_l3 LIKE "Quetal")
```

Query construction comprises three main aspects. Firstly, information that is requested must be encoded by variables following the starting SELECT clause. We make use of the keyword DISTINCT to rule out duplicate occurrences in case no OP-COND condition (which enforces counting) is specified in the proto query. However, if an operator condition is present, DISTINCT should not be added because duplicates must be taken into account for arithmetic operations in quantificational questions (e.g. *Who led most projects in Information Extraction?*).

Secondly, RDF triples are collected in the FROM clause, separated by commas, which implicitly express logical conjunction. A restricted form of disjunction is available at this point due to the optionality operator [ ] which expresses information that need not be matched.

Thirdly, additional restrictions on variables can be formulated in the WHERE clause, including equality (=), inequality (!=) and string matching (LIKE). These restrictions can be combined using the Boolean connectives.

Returning to the example above, according to principle (1) of Figure 5, two root variables _r1 and _r2 are introduced and linked to concepts Active_Person and Active_Project via rdf:type. Principle (2) leads to the query variable _q0 which is linked to _r1 via the property name, the default property in case the attribute value in a SELECT-COND is empty (= ""). From (3), we get a variable _l3 which binds the value of attribute val ("Quetal") in the proto query above. The value itself is specified in the WHERE clause of the SeRQL query.

The most interesting aspect in the query construction process is how to account for partially connected concepts, cf. principle (4). In our example, two relations/concepts _r1 and _r2 are introduced in the proto query. Since we want to retrieve information related to _r1 (via _q0), it is important that _r1 and _r2 are *connected*. Otherwise the information from _r2 (through rdf:type and ltw:projectName) can not be incorporated into the search of the RDF data base. Put differently, if we did not account for connecting concepts/properties between _r1 and _r2, the above SeRQL query would simply retrieve all instances of concept Active_Person.

It is the last two clauses of the FROM condition and the first WHERE clause that account for this problem. Since _r1 and _r2 are not connected and no information is given regarding a connecting *property* and the *direction* of such a connecting property (from _r1 to _r2, or vice versa?), we let Sesame "guess" this information. Firstly, in order to guess the property, we use *property* variables in the predicate position of an RDF triple. Secondly, in order to guess the direction, we need some kind of disjunction on the FROM level. Here the optionality operator comes into play. Notice that there may be several different properties, connecting _r1 and _r2, even properties from _r1 to _r2 *and* from _r2 to _r1 at the same time. In order not to rule out both optionality statements, we have to formulate further constraints, specifying that the two property variables should not be NULL at the same time.

Looking at this from a graph-theoretical perspective, we are interested in *constraints*, characterising directed *minimum spanning trees* (Garey & Johnson, 1979, p. 130). The nodes in such a tree are exactly the root nodes representing the concepts, and the edges which connect the nodes represent the missing properties. The missing edges are either specified via path expressions, or represented by property variables which need to be instantiated by Sesame through a DB search. Clearly, if path expressions are specified in a proto query, they will be utilized to speed up the ontology serch, resulting in less non-determinism.

Related Approaches and Future Work

In this paper we presented an approach for domain-restricted QA from structured knowledge bases, building on deep semantic question analysis with a modular interface between conceptual semantic representations and domain-specific ontologies or data bases. The architecture embodies a flexible interface to various types of knowledge storage devices and their corresponding query languages.

Our architecture extends traditional NL-based interfaces to data bases (NLIDB), developed in the seventies and eighties (cf. overview in Androutsopoulos & Ritchie, 2000). It builds on more general resources in both linguistic and knowledge modeling, and a clear separation into modular layers: linguistic semantic analysis, lexical-conceptual representation and knowledge-based conceptual modeling. For these reasons, our approach promises to be more scalable and portable to new domains.

The large-scale grammars are not tailored to a specific domain. They deliver a semantic representation (RMRS) that is uniform across languages and is shared with shallow NE recognition grammars. The frame semantic layer accounts for multi- and crosslingual QA, by capturing linguistic variation and paraphrases of semantically equivalent questions. It offers a modular interface for the mapping of general linguistic concepts to domain-dependent ontologies that can be systematically adapted to new domains. Moreover, the normalised conceptual linguistic structures could be employed for textual QA tasks, in open or restricted domains, by matching enriched question and answer candidate analyses. Contrary to traditional NLIDB approaches, our ar-

chitecture uses ontologies as the interface between question analysis, answer extraction and knowledge engineering.

Our prototype system is still small, but has been tested on various question types (wh-, yes/no-, imperative, definition, quantified questions). Future work will focus on adding further question types as well as research into automation techniques for frame and sense assignment, and the induction of mappings to domain concepts. We will further investigate disambiguation of the stochastically ranked question analyses by selecting maximally specific proto queries.

For our current system we performed an impressionistic evaluation by sending 10 questions from the Nobel Prize domain to the web-based open-domain QA system Answerbus (http://www.answerbus.com). They include factoid, list and enumeration questions, to which our system provides correct answers. Some questions contain time expressions and quantification. Answerbus answers 4 out of 5 factoids correctly, yet cannot answer list and enumeration questions, nor does it deliver correct answers when precise semantic analysis is needed, such as "prize winners before 1911".

An issue to be solved for the envisaged integration of open text unrestricted domain QA and restricted QA on structured, semistructured and unstructured data is the selection of the returned response in cases where the individual searches yield different results. The above experiment suggests that even without an empirically validated selection strategy, a mere preference for the restricted domain response over the result of open domain free text QA would improve overall accuracy.

Acknowledgements The research reported here has been conducted in the project QUETAL (http://quetal.dfki.de), funded by the German Ministry for Education and Research, grant no. 01 IW C02. Special thanks go to Bogdan Sacaleanu for implementation of a QA-control server that connects question and answer processing, and to Gregory Gulrajani for advice in the setup of a Sesame server. Günter Neumann and Bogdan Sacaleanu provided major contributions to the realisation of the overall QUETAL QA architecture.

References

Androutsopoulos, I., and Ritchie, G. 2000. Database interfaces. In Dale, R.; Moisl, H.; and Somers, H., eds., *Handbook of Natural Language Processing*.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. 2003. *The Description Logic Handbook*. Cambridge University Press.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The Berkeley FrameNet project. In *Proc. of COLING-ACL*.

Baldwin, T.; Bender, E.; Flickinger, D.; Kim, A.; and Oepen, S. 2004. Road-testing the English Resource Grammar over the British National Corpus. In *Proc. of LREC*.

Burchardt, A.; Erk, K.; and Frank, A. 2005. A WordNet Detour to FrameNet. In *Proceedings of the 2nd GermaNet Workshop*.

Callmeier, U.; Eisele, A.; Schäfer, U.; and Siegel, M. 2004. The deepthought core architecture framework. In *Proceedings LREC*, 1205–1208.

Callmeier, U. 2000. PET – a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering* 6(1):99–108.

Copestake, A.; Flickinger, D.; Sag, I.; and Pollard, C. 2005. Minimal Recursion Semantics. To appear.

Copestake, A. 2003. Report on the Design of RMRS. Technical Report D1.1a, University of Cambridge, UK.

Crouch, R. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings IWCS*.

Crysmann, B.; Frank, A.; Kiefer, B.; Müller, S.; Neumann, G.; Piskorski, J.; Schäfer, U.; Siegel, M.; Uszkoreit, H.; Xu, F.; Becker, M.; and Krieger, H.-U. 2002. An Integrated Architecture for Deep and Shallow Processing. In *Proceedings ACL*.

Drożdżyński, W.; Krieger, H.-U.; Piskorski, J.; Schäfer, U.; and Xu, F. 2004. Shallow processing with unification and typed feature structures —foundations and applications. *K ünstliche Intelligenz* 1:17–23.

Frank, A., and Erk, K. 2004. Towards an LFG syntax–semantics interface for Frame Semantics annotation. In Gelbukh, A., ed., *Computational Linguistics and Intelligent Text Processing*. LNCS, Springer.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman.

Guo, Y.; Pan, Z.; and Heflin, J. 2004. An evaluation of knowledge base systems for large OWL datasets. In *Proceedings of ISWC 2003*. Springer.

Haarslev, V., and Möller, R. 2003. *RACER User's Guide and Reference Manual, Version 1.7.7*.

Horrocks, I.; Sattler, U.; and Tobies, S. 2000. Reasoning with individuals for the description logic SHIQ. In *Proceedings of CADE-17*. Springer.

Horrocks, I. 1998. *FaCT Reference Manual*.

Knublauch, H.; Musen, M. A.; and Rector, A. L. 2004. Editing description logic ontologies with the Protégé OWL plugin. In *Proc. of the International Workshop in Description Logics*.

Miller, G. A.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. 1993. Five papers on WordNet. Technical report, Cognitive Science Laboratory, Princeton.

Neumann, G., and Sacaleanu, B. 2003. A Cross-language Question/Answering System for German and English. In *Proceedings of the CLEF-2003 Workshop*.

Neumann, G., and Sacaleanu, B. 2004. Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering System. In *Proceedings of the Working Notes for the CLEF-2004 Workshop*.

Niles, I., and Pease, A. 2001. Origins of the Standard Upper Merged Ontology: A proposal for the IEEE standard upper ontology. In *IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*.

Niles, I., and Pease, A. 2003. Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering*.

Niles, I., and Terry, A. 2004. The MILO: A general-purpose, mid-level ontology. In *2004 International Conference on Information and Knowledge Engineering*.

Reynolds, D. 2004. *Jena 2 Inference support*.

Uszkoreit, H., and Xu, F. 2005. Semantic model for information extraction. DFKI, forthcoming.

Uszkoreit, H.; Jörg, B.; and Erbach, G. 2003. An ontology-based knowledge portal for language technology. In *Proc. of ENABLER/ELSNET WS* Intern. Roadmap for Language Resources.