Creation and Use of Lexicons and Ontologies for NL Interfaces to Databases

Roberto Bartolini*, Caterina Caracciolo*, Emiliano Giovanetti*, Alessandro Lenci^{++},

Simone Marchi*, Vito Pirrelli*, Chiara Renso, Laura Spinsanti****

*Istituto di Linguistica Computazionale, CNR
v. Moruzzi 1, Pisa, Italy
{roberto.bartolini, caterina.caracciolo, emiliano.giovannetti, simone.marchi, vito.pirrelli}@ilc.cnr.it

**Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", CNR
v. Moruzzi 1, Pisa, Italy
{chiara.renso, laura.spinsanti}@isti.cnr.it

$^{++}$Dipartimento di Linguistica, Università di Pisa
v. Santa Maria 36, Pisa, Italy
alessandro.lenci@ilc.cnr.it

Abstract

In this paper we present an original approach to natural language query interpretation which has been implemented within the FuLL (Fuzzy Logic and Language) Italian project of BC S.r.l. In particular, we discuss here the creation of linguistic and ontological resources, together with the exploitation of existing ones, for natural language-driven database access and retrieval. Both the database and the queries we experiment with are Italian, but the methodology we broach naturally extends to other languages.

1. Introduction

Nowadays, databases (DBs) are the standard technology for storing vast amount of data about a variety of domains and making them available to a larger and larger population of users with scanty, if any, competence of formal query languages (*e.g.* SQL). Natural language, on the other hand, appears to be an optimal substitute for formal query languages in allowing common users to access DBs according to their own familiar concepts and requirements. Yet, it should be appreciated that natural languages and databases impose very different, if not radically opposite, requirements on the way information content is represented and accessed. The explicit and structured way in which information is stored in DBs is in sharp contrast with the inherent vagueness and implicitness of natural language semantics and, more generally, with the wayward way users conceptualise the goal-oriented information they look for. Bridging the gap between these two realms thus presents a formidable challenge for NLP systems, but also promises to offer more flexible and effective ways to access DBs from a user-centred perspective.

In this paper we describe an original approach to NL query interpretation which has been implemented within the FuLL (Fuzzy Logic and Language) project of BC s.r.l. Software Company, funded under PIA INNOVATION measure of MAP (Ministero Attività Produttive). The approach relies on a *domain ontology* as a knowledge representation interface between the logical structure of a DB and natural language semantics. The ontology model is intended to represent an abstract description of the DB structure and provide, at the same time, the "conceptual" level on which natural language queries are interpreted. Specific issues that will be addressed in the paper are i.) the design features of the ontology and ii.) the way it interacts with an automatically derived linguistic representation of natural language queries so as to produce logical forms that are eventually translated into SQL.

A further central element of the present work is the particular domain of its application: *Geographical Information Systems* (GIS). The last several years have witnessed a flourishing development of geographical data bases, manipulated by means of GIS and used in all applications and domains that involve reference to the territory, *e.g.* urban planning and regulation, satellite technologies and route planning among others. The interaction between users and GIS is a well-known critical issue in the specialized literature (Medyckyj-Scott at al 1993, Nyerges et al 1995). GIS user interfaces with natural-language components are found rarely. Current GIS are useful in answering metric-based queries, which involve precise angular and distance measures. On other hand, vague semantic expressions are typical of the way people query spatial information through natural language. For example, people only rarely ask for detailed spatial information, as in "`Give me all the shops within 35 meters from the railway station`", since they rather prefer to use vague expressions such as in "`give me all the grocery shops` *`near`* `the station`". GIS that are flexible enough to accommodate these human requests can reasonably be expected to meet a wider audience and user community than current systems requiring GIS specialist users (Max J. Egenhofer at al 1998). Besides, the way speakers talk about spatial concepts is a challenging case study for investigating genuine theoretical issues in natural language: namely, the way inherently non symbolic concepts such as distance and path are conveyed through a set of basically symbolic units (a word lexicon) which nonetheless combine in a variety of semantically graded, weakly compositional constructions.

The remainder of this paper is organized as follows: section 2 provides some background overview of natural language interfaces to DBs and sketches the logical architecture of FuLL's interpretive components. In Section 3 we describe the ontology modelling phase, while in Section 4 we provide more details about the query interpretation process in a strict sense. Section 5 gives a quick highlight on thecnologhy FuLL's preliminary evaluation results. Section 6 offers some concluding remarks.

2. Background & General Strategy

The issue of providing a natural language interface to database (NLIDB) has been explored since the late sixties and early seventies (Androutsopoulos et al., 1995). Various strategies have been adopted in order to use unconstrained natural language queries to access DBs. Pattern matching approaches rely on a set of rules activated by specific linguistic patterns in the query in order to access the appropriate table(s) in the database. While these techniques proved to give very good results compared with the shallowness of the analysis, cases of bad failures are also reported in the literature (Johnson, 1985). Other systems use some form of syntactic analyses and try to map syntactic trees onto the DB data structure. The approach proved to be fruitful in application-specific database systems (Perrault, 1988), but it also appears to be hardly extensible to general database query languages such as SQL.

More sophisticated strategies rely on some form of semantic analysis of the query. Systems based on "semantic grammars" were quite popular in the past decades, but in more recent times they have been largely replaced by systems using one or more layers of some intermediate representation language. The user query is translated into a set of clauses expressing high level logico-semantic representations, independent of the actual underlying database. In some cases, the module generating the intermediate level also encodes a world model, typically consisting of a *is-a* hierarchy of concepts plus constraints to limit the predicate arguments that can appear in the logical form (Alshawi, 1992). Recent developments include the possibility of inducing transformation rules to map natural language queries into a formal query of command language (Kate, 2005).

The use of an intermediate level of semantic representation raises the issue of its format and structure. Recently, much effort has been devoted to the development of "light" semantic formalisms, an important example being *Minimal Recursion Semantics* (MRS; Copestake et al. 2004). Such shallow semantic representations have the threefold advantage of reducing the amount of potential structural semantic ambiguities (*e.g.* quantifier scope), enhancing the robustness of the overall process of logical form construction, and providing a mapping onto semantic structures even from largely underspecified syntactic analyses. As will be shown in more detail below, in our approach natural language queries are mapped onto a level of logico-semantic representation (or Logical Form, LF) explicitly reminiscent of MRS, generated from a level of linguistic representation (LR) of the query. LF is linked to the domain ontology, which acts as a formal interpretive model of LF predicative constants and expresses the conceptual restrictions constraining the compositional process of building logical forms out of the syntactically analysed natural language inputs (see section 4.2). The logical architecture of this strategy is diagrammed in Figure 1 below.

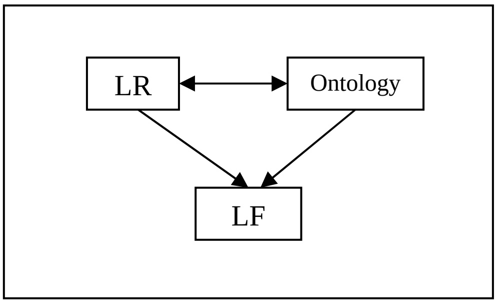

Figure 1: Logical architecture of FuLL's interpretive components

Logical forms are eventually automatically mapped onto SQL queries by other specific components of FuLL's overall architecture, namely an LF post-processor and parser, a Fuzzy Engine dealing with treatment of lexical vagueness, an SQL translation Engine and, where necessary, a Dialogue Manager.

In what follows, we shall focus on the two key-aspects of ontology modelling and query interpretation.

3. Ontology Modelling

Domain ontologies are to be clearly distinguished from so-called linguistic ontologies. While the former aim at representing a particular conceptualization of a domain, the latter include a repertoire of concepts lexicalized by natural language expressions. This opposition is essential to clarify the status and role of the ontology in this work. The domain ontology is designed to model the conceptual constraints of the GIS domain. This also implies that ontology modelling is not guided by linguistic criteria, but only aims at encoding the relevant constraints to be met within the target domain. This section discusses some specific issues concerning domain ontology modelling, while we leave to next section the discussion of how the ontology is linked to natural language interpretation.

The classic ontology building life-cycle can be synthesized as a top-down process, starting with the identification of the purpose of the ontology, going through the knowledge acquisition task and ending with the encoding of the obtained conceptualisation of the domain using the chosen knowledge representation language. In this work we have a combined *top-down* and *bottom-up* approach. More precisely, we have adopted a sort of "reverse-engineering" bottom-up strategy that, starting from a database schema structure, is enriched with the introduction of domain concepts and properties not directly encoded inside the DB, but needed to provide an exhaustive representation of the domain on the basis of the user needs.

As a test bed for this GIS-oriented ontology modelling approach, within the FuLL project the "urban planning" domain was selected. In particular, we have based the bottom-up side of the construction upon a set of tables defining a specific but representative class of "urban" entities. Regarding the "bottom up" aspect, in many cases several ontology classes and properties have been defined starting from the very same table. For example the table "GAS_STATION" has been used to define different classes and properties such as the concept of *gas station* itself, the *owner company* and the *company flag*. Many properties in the ontology them have been extracted from the same table and used to link the core class *gas station* to other classes such as *road* as showed in Fig. 2.

On the other hand, concerning the "top down" aspect of the ontology modelling process, some concepts contained in the database appeared fragmented over many different tables. This is, for example, the case of the "road" concept in a Urban Planning domain: here the *road* concept (denoting streets, highways, main roads, etc.) is obviously central. However in an urban planning DB "road" information is typically split into a series of more detailed tables, such as road junctions, road elements, road names, highways, regional roads and so on. It is therefore extremely important to define a high level ontology concept of "road" that abstracts away from DB specific data.

In the ontology definition particular emphasis was laid on *spatial* aspects of the domain. Indeed, when dealing with a spatial domain, besides descriptive attributes of concepts, also *geometrical* and *positional* information about them need to be represented. A geometrical attribute defines the kind of spatial object represented, such as polygon, point or line. Positional attributes define the geographical position of the object in space. Objects with positional attributes are often denoted as "georeferenced".

A set of spatial features has been introduced in the ontology for the correct representation of the chosen domain. To do this, two alternative directions can be taken. On the one hand, a specific formalism for spatial (or spatio-temporal) ontologies could be adopted. However, this kind of formalisms are at a very early stage of development. The other possible direction, adopted in the approach presented here, is to enrich a standard ontology representation formalism with *meta information* with specific spatial semantics. As an example, meta information can be used to denote, for each concept, which are the "location" attributes. This also allows us to deal with queries containing complex spatial operators like "distance", that are applied to georeferenced concepts of the ontology.

4. Query Interpretation

In FuLL, the ontology is designed so as to provide a conceptual abstraction from the target DBs. One of the main challenges of this work is the use of this knowledge structure for the purposes of query interpretation. The solution we present here sees query interpretation as a process of mapping natural language structures onto non-recursive logical form (LF) structures. Within this process, domain ontology plays a twofold role: i.) it provides an interpretation of non-logical constants in LF, ii.) it drives the process of LF construction by using the domain structure as semantic constraints on the composition of atomic predicates into complex LF expressions and on variable linking.

To be more concrete, a typical example of the range of requests we would like FuLL system to successfully deal with is the following: *Posso fare benzina nei pressi dell'ospedale S. Chiara a Pisa?* 'Can I get gasoline nearby S. Chiara Hospital in Pisa?'. In order to correctly translate such a request into an SQL query, one needs to recognize that the user wants to know the address of a gas station, that the gas station must be "close" to S. Chiara Hospital, and that both the gas station and the hospital must be located in the city of Pisa (*i.e.*, Pisa is a city name). In other words, the following pieces of ontological information must be available to the system: (i) gas stations are geographically located entities, whose position is uniquely identifiable through an address, at a certain distance from other locations of a compatible ontological type; (ii) when asking if it is possible to get gasoline, a user is implicitly asking about the address of a gas station (its identifiable location), and (iii) Pisa is a city.

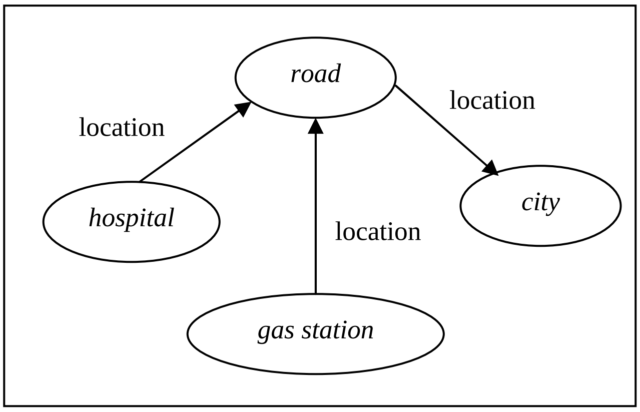

Figure 2: A fragment of the domain ontology

The ontologically interpreted LF is used as a bridge between the linguistic content of a natural language query and its intended SQL rendering. Formally, the LF is defined as a conjunctive set of simple (non recursive) first order predicates, (under)specified over (possibly unbound) argument variables~~.~~

To ensure that the various aspects of the query interpretation are effectively integrated, we must provide formal means to map natural language expressions onto the various components of the full LF structure. This in turn implies: i) assigning proper names to appropriate ontological categories; ii) associating lexical terms with ontologically interpreted concepts; iii) projecting complex events against their intended ontological background; iv) identifying the pragmatically-relevant focus of the question at hand, *i.e.* the topic which the user wants to know something about. We have dealt with these issues by adopting a general strategy where interpretation is split into two phases: i.) *lexicon-to-ontology mapping*, in which an application specific lexicon is used to map simple and complex terms onto the domain ontology nodes; ii.) *compositional mapping*, in which ontology-derived semantic constraints and linguistically derived syntactic constraints are used to build the complete LF structure. While ii.) requires an advanced level of on-line processing of the syntactic structure of the sentence, i.) can profitably be addressed by developing a lexicon-to-ontology mapping function. In the next subsections we deal with both phases in more detail.

4.1. Lexicon-to-Ontology Mapping

Even if we are dealing with one natural language only, the need for keeping lexicon and domain ontology, in spite of their tight relationship, as two fundamentally independent repositories rests on two major reasons. First, a simple concept can be expressed linguistically by means of a complex construction whose interpretation is only weakly compositional. By this we mean that it would be exceedingly difficult, in most cases, to interpret such constructions through semantic functions defined on ontologically interpreted simpler constituents. What we seem to be in need of is a loose notion of “phraseological” lexicon whose entries may consist of entire text chunks or even long-distance dependency constructions. By doing so, we are picturing the relationship between a linguistic lexicon and an ontology as complex dynamic mapping, rather than a straightforward, one-to-one static relationship. Later on, we shall give illustrative examples of this mapping.

The second major reason has to do with the treatment of lexical vagueness and relational operators. For instance, the intrinsic vagueness of adjectives like *close* or *large* is not possibly represented in the domain ontology proper, but rather as a property of lexical items. Moreover, some of these lexical entries do not have a corresponding interpreted node inside the domain ontology, as they do not refer to any inherent property of an ontological class, but rather to a varying relationship between instances of a class. For example, there exists an entire class of lexical entries (associated with different grammatical categories) that are mapped onto the spatial operator `distance`. Words such as *close, far, distant, distance* etc. are used in remarkably different constructions for different purposes:

i. as constraints over classes in the user query, by means of restrictive adjectives like *close, far* etc.: “Is there any gas station close to the hospital?”

ii. as the “focus” of the user query, through terms like “distance”, “distant”, and the like: “How distant is the shopping centre from the railway station?”

The `distance` operator is constrained to apply to pairs of geo-referenced ontology classes only and can, in its use i), be assigned different graded values, depending on the lexical entry used in the query. Graded values are, once more, defined within the lexicon, not in the ontology, and eventually processed by FuLL’s Fuzzy Engine.

We now move on to consider, in more detail, a range of increasingly difficult cases of lexicon-to-ontology mapping.

Probably the linguistically simplest case of such mapping is represented by *proper names*, which can be tokenized with no or very little recourse to parsing and must be recognized as being associated with a specific ontological category (address, company name, city name etc.). At the moment, named-entity classification of this kind is carried out on the basis of a closed list of domain-specific names (a gazetteer) contained in the DB of interest, and makes provision for a minimum of surface variants (*e.g.* abbreviations for address names). More flexible, machine-learning categorization strategies can be envisaged to allow the system to provide more informative feedback to the user in cases of gaps in the gazetteers.

Both simple (*e.g. gasoline*) and complex terms (e.g. *gas station*) are mapped onto appropriate ontological categories by means of a lexico-semantic repertoire, whereby inflected forms are normalised into lexical exponents (lemmas) and paired with their corresponding concept (be it a class or a property). While proper names require no or little processing, identification of simple and multi-lexical terms must take place at a level of linguistic analysis where words have already been lemmatised, to account for their inflectional variability. Moreover, Italian complex terms typically exhibit an *NP prep NP* structure (*e.g. stazione di rifornimento, stazione per il rifornimento*, both meaning ‘gast station’), with some variability in preposition selection and optional intervening determiners and modifiers. To account for such a variability, complex term identification must abstract away from optional determiners and modifiers, and focus on the bare relationship between noun heads only.

We attain this level of abstraction by *chunking* natural language queries, *i.e.* by segmenting the text into non recursive basic constituents (or *chunks*, refs. here). Each chunk contains an indication of its (obligatory) lexical head, optional modifiers and determiners and one or more introducing prepositions. Chunking complex terms allows normalisation of their surface syntactic variability, thus simplifying the term-concept mapping in the lexico-semantic repository (see example below).

```
1. [[CC: FV_C] [AGR: @S1] [MOD: POTERE#V@S1IP]
   [POTGOV: FARE#V@F]]
2. [[CC: N_C] [AGR: @NN] [POTGOV: BENZINA #S@FS]]
3. [[CC: P_C] [AGR: @NS] [PREP: NEI_PRESSI_DI#E]
   [DET: LO#RD@FS@MS] [POTGOV:OSPEDALE_S._CHIARA
   #SP@NN]]
4. [[CC: P_C] [AGR: @NN] [PREP: A#E] [POTGOV:
   PISA #SP@NN]]
5. [[CC: PUNC_C] [PUNCTYPE: ?#@]]
```

In the lexico-semantic repository, complex predicative constructions such as *get gasoline* can be used as pointers to non trivial ontological properties (*e.g. it supplies gasoline*, predicated of gas stations). We adopt this strategy to constrain the role of the ontology model in filling in an underspecified LF representation.

It should be appreciated that mapping predicates and their arguments onto ontological concepts requires a non trivial level of text processing. The predicative construction *get gasoline* can be phrased in a number of ways, ranging from declarative to interrogative, personal to impersonal constructions. If we want to use *get gasoline* as an abstract pointer to an ontological concept, then we need to be able to abstract away from such a wide variety of constructions. Annotating the query text with an explicit indication of the pair-wise dependency relations holding between lexical heads goes a long way in this direction. A functional analysis of the query is represented below, where lexical heads are associated with dependency relations such as subject, direct-object, complement, modifier and argument:

```
SUBJ(FARE[1],PRO[0]<agr = S1>)
MODIF(FARE[1],POTERE[1]<role = MODAL>)
OBJD(FARE[1],BENZINA%EROGACARBURANTE[2])
COMP(FARE[1],OSPEDALE_S._CHIARA%NOMEPOLO[3]
    <intro = NEI_PRESSI_DI>)
ARG(FARE[1],PISA%NOMECOMUNE[4]<intro = A>)
```

4.2. Compostitional Mapping

Building LF requires the dynamic merging of linguistically structured information (made accessible, as we saw, through intermediate stages of increasingly abstract parsing) and a "world model", represented as a domain-specific ontology (see section 3). The purpose of assigning linguistic structure to a natural language query is to single out text-to-ontology "anchors", that is word sequences and constructions, such as proper names, simple and complex terms, event designators etc., that play the role of linguistic pointers (of varying structural complexity) to ontological concepts. Anchors are used as constraints over LF structures and variable binding. As we saw in the previous section, they are dealt with at the level of lexicon-to-ontology mapping, whereby natural language makes contact with the domain ontology.

A free natural language question, however, is likely to be dramatically underspecified as to the database content the question is intended to tap into. In the relatively simple example mentioned above, asking about the location of a gas station requires no mentioning of a location, let alone a gas station. This radical form of presupposition, typical of domain specific, natural language questions, calls for massive recourse to background knowledge and inference, under suitable linguistic constraints.

Under the closed world assumption that the ontology faithfully represents all those aspects of the domain the database says something about, it makes eminent sense to let the ontology fill in pieces of information that are missing and presupposed in the natural language query. For example, one can reasonably link the event *get gasoline* (overtly conveyed in the query) to the *property it dispenses gasoline*, predicated of the class *gas station* in the ontology. Moreover, since the event *get gasoline* is in the question focus, it takes a small inferential step to transfer the focus to the specific ontological class *it dispenses gasoline* is a property of.

This form of LF expansion is attained by navigating the ontology at the time LF is produced, looking for all possible paths linking overtly mentioned ontological classes, and for the ontological classes which are only presupposed (but not explicitly mentioned) in the natural language query.

This is a good example of the strategy of building LF as a result of multiple constraint satisfaction. While the common sense fact that gas stations dispense gasoline is stored in the domain ontology, indirect reference to this knowledge is mediated through a complex event designation in the text. It is important that both pieces of knowledge (ontological and linguistic) are checked simultaneously so as to rule out unconstrained interpretation of conceptually different but apparently similar event designators (e.g. burn gasoline).

To sum up, production of LF boils down to a process of hybrid constraint satisfaction. Constraints are hybrid for two reasons. First, they are derived from different levels of linguistic analysis of the query text. Secondly, they are derived "bottom-up", through propagation from the query content, and "top-down", through percolation from the ontology model.

5. Preliminary Evaluation

The evaluation we report here focuses on the translation of natural language queries into LF. The interpretive components have been assessed on a sample of 211 Italian queries, intentionally phrased so as to represent a challenging test bed and be instrumental in spotting potential errors and weaknesses in the analysis and translation process. All queries were processed automatically and then manually checked for errors. Of them, 29 queries turned out to contain spelling errors or to address an irrelevant domain and were thus discarded for evaluation. Of the remaining 182 queries, 126 were concerned with entities fully covered by the ontology. 95 of them were fully translated into well-formed FL, thus scoring an encouraging 80% recall when ontology provides the required information. Most of the remaining ill-formed LFs (22 out of 31) present a mistake in the query focus (see Table 1 below). As focus errors have an immediate repercussion on the LF as a whole, we expect that considerably better results can be obtained by refining the pragmatic component in the parsing assembly line. Other potential sources of errors are conceptually complex and strongly elliptical queries, lexical ambiguity and indirect questions.

# original queries	211		
# with no spelling errors	182		▨
# fully-covered by ontology	126		▨
# well-formed LF	95		▨
# ill-formed LF	▨	31	▨

Table 1: Evaluation results of interpretive components

First results of the FuLL project as a whole (including fuzzy processing of LF and LF to SQL translation) have been applied to a specialized SW prototype, focused on the following domains: "territorial and settlement systems", "territorial and urbanistic cartography", "transport and mobility". The evaluation is actively supported by the Provincia of Bologna and Provincia of Catania, which have kindly provided their territorial data bases and tested the prototype, acting as final users.

6. Discussion and Conclusion

One of the main research issues addressed in this paper concerns the place and role of ontologies in the process of interpreting natural language queries. Ontologies represent a way of structuring our knowledge on the basis of a certain application domain, and can thus open a door onto the use of natural language to query and search goal-oriented information. This requires, however, that we are aware of the existing divide between ontologies and natural language and that we develop principled strategies to bridge the gap.

We presented here an architecture that addresses this issue in the particular setting of a NL interface to GIS databases. The backbone of FuLL's approach is to achieve a sort of "*division of interpretive labour*" between a domain ontology and NLP query analysis. In fact, while the ontology provides domain dependent knowledge constraints, advanced NLP query analysis allows us to extract from the query the linguistic information necessary for its mapping onto a LF, which can then be translated into a SQL query.

The main assumption underpinning this approach is that neither a complete NLP analysis nor a fully-developed ontology is singly sufficient to achieve a correct query-to-LF mapping. On the one hand, an ontology is typically silent about the particular, context-sensitive and goal-oriented perspective entertained by a NL query user in addressing a domain specific content. Moreover, the information structure of a typical ontology grudgingly lends itself to accommodating the inherent ambiguity and vagueness of NL. On the other hand, a NL query is often, at the same time, both underspecified and redundant with respect to the particular pieces of information the user query is about, as they can be concealed behind idiomatic expressions, indirect questions, ellipsis, etc. To cope with these problems in a principled way, we designed an architecture where linguistic and ontological constraints work in step so as to maximize their respective contribution. A domain ontology is used and navigated to drive the interpretation process by filling in missing information and solving potential ambiguities in the query. Linguistic analysis, in its turn, effectively deals with the perspectivizing factors and focusing hints NL is very good at conveying, thus suggesting unique ways of recombining ontological knowledge to get a unique well-formed LF.

The approach is demonstrably able to reach a fairly satisfactory level of robustness and fault tolerance, an important pre-requisite for realistic and effective NL interfaces to databases. The negative impact of parsing failures in the query analysis is mitigated by the recourse to ontological constraints even in those cases where the linguistic analysis is far from perfect. This does not mean that we only rely on shallow linguistic processing. To the contrary, automatically identified deep grammatical dependencies are often used to help the system to select the proper interpretation. The upshot is that, in any case, completeness of NLP analysis is not a necessary condition for the system to home in on a correct LF interpretation.

Many open problems remain that deserve further work, the major one probably being the strong reliance of this approach on an existing, well-developed domain ontology. This has also an impact on issues of scalability, as the size of the ontology may exponentially increase the number of potential ambiguities in the query, and make the process of integrating ontology and linguistic information potentially combinatorial.

References

Alshawi, H. (1992). The Core Language Engine. MIT Press, Cambridge, Massachussets.

Androutsopoulos, I. & Ritchie, G.D. & Thanisch P. (1995). Natural Language Interfaces to Databases – An Introduction. Journal of Natural Language Engineering, Cambridge University Press. 1(1), (pp. 29-81).

Copestake, A. Flickinger, D., Pollard, D., Sag. I. (2004). Minimal recursion semantics: An introduction. *Journal of Language and Computation*.

Egenhofer, M. & Shariff, A. (1998) Metric details for natural-language spatial relations. ACM Transactions on Information Systems (TOIS) Volume 16 , Issue 4 Pages: 295 - 321

Epstein, S.S. (1985). Transportable Natural Language Processing Through Simplicity – the PRE System. ACM Transactions on Office Information Systems, 3(2) (pp. 107–120).

Johnson, T. (1985). Natural Languages Computing: The Commercial Applications. Ovum Ltd., London, 1985.

Kate, R.J. & Yuk Wah Wong, Mooney, R.J. (2005). Learning to Transform Natural to Formal Languages. Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05), (pp. 1062—1068), Pittsburgh, PA, July 2005.

Lenci, A & Montemagni, S & Pirrelli, V. (2001),.CHUNK-IT. An Italian Shallow Parser for Robust Syntactic Annotation, Linguistica Computazionale, Istituti Editoriali e Poligrafici Internazionali, Pisa-Roma.

Medyckyj-Scott, D. & Hearnshaw, H. 1993. Human Factors in Geographical Information Systems. Belhaven Press, London.

Miller, G. (1995). WordNet: A Lexical database for English. CACM. 38(11), (pp. 39—41).

Nyerges, T. & Mark, D. & Laurini, R. & Egenhofer, M. (1995) Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems. Kluwer, Dordrecht, 1–8.

Perrault, C.R. & Grosz, B.J. (1988). Natural Language Interfaces. In H.E.Shrobe (ed.), Exploring Artifical Intelligence (pp. 133—172). Morgan Kauffmann Publishers Inc., San Mateo, California.

Popescu, A-M. & Armanasu, A. & Etzioni, O. & Ko, D. & Yates, A. (2004). Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability. Proc. of COLING 2004.

Tennant, H.R. & Ross, K.M. & Thompson, C.W. (1983). Usable Natural Language Interfaces through Menu-Based Natural Language Understanding. In Proceedings of CHI'83, Conference on Human Factors in Computer Systems, Boston. ACM.

Waltz, D.L. (1978). An English Language Question Answering for a Large Relational Database. Communications of the ACM, 21(7), (pp. 526—539).