Semantic Information Preprocessing for Natural Language Interfaces to Databases

Milan Mosny
Simon Fraser University
Burnaby, BC V5A 1S6,
Canada
mosny@cs.sfu.ca

Abstract

An approach is described for supplying selectional restrictions to parsers in natural language interfaces (NLIs) to databases by extracting the selectional restrictions from semantic descriptions of those NLIs. Automating the process of finding selectional restrictions reduces NLI development time and may avoid errors introduced by hand-coding selectional restrictions.

1 Introduction

An approach is described for supplying selectional restrictions to parsers in natural language interfaces (NLIs) to databases. The work is based on Linguistic Domain Theories (LDTs) (Rayner, 1993). In our approach, we propose *a restricted version of* LDTs (RLDTs), that can be normalized and in normalized form used to construct selectional restrictions. We assume that semantic description of NLIs is described by such an RLDT.

The outline of the paper is as follows. Section 2 provides a brief summary of original LDTs, illustrates how Abductive Equivalential Translation (AET) (Rayner, 1993) can use them at run-time, and describes RLDTs. Sections 3 and 4 describe off-line processes - the normalization process and the extraction of selectional restrictions from *normalized* RLDTs respectively. Section 5 contains discussion, including related and future work.

2 LDT, AET and RLDT

LDT and AET. LDT was introduced for a system, where input is a logical formula, whose predicates approximately correspond to the content words of the input utterance in natural language (lexical predicates). Output is a logical formula, consisting of predicates meaningful to the database engine (database predicates). AET provides a formalism for describing how a formula consisting of lexical predicates can be tranlsated into formula consisting of database predicates. The information used in the translation process is an LDT. A theory Γ contains horn clauses

$$\forall(P_1 \wedge \cdots \wedge P_n \rightarrow Q)$$

or universal conditional equivalences

$$\forall(P_1 \wedge \cdots \wedge P_n \rightarrow (R_1 \wedge \cdots \wedge R_l \equiv F))$$

or existential equivalences

$$\forall((\exists X_1 \cdots X_m.P) \equiv F)$$

where P_i, R_i denote atomic formulas, Q denotes a literal, F denotes a formula and $\forall$ denotes universal closure. The LDT also contains functional relationships that are used for simplifications of the translated formulas and assumption declarations. Given a formula F_{ling} consisting of lexical predicates and an LDT, AET tries to find a set of permissible assumptions A and a formula F_{db} consisting of the database predicates such that

$$\Gamma \cup A \Rightarrow \forall(F_{ling} \equiv F_{db})$$

The translation of F_{ling} is done one predicate at a time. For each predicate in the formula F_{ling}, there is a so-called conjunctive context that consists of conjuncts occurring together with the predicate in F_{ling}, meaning postulates in the theory Γ, and the information stored in the database. Given an LDT, this conjunctive context determines how the predicate will be translated by AET.

As an example, suppose that the lexical representation of the sentence *Is there a student who takes cmpt710 or cmpt720?* is F_{ling}:

$$\exists X, E, Y, Y_1.student(X) \wedge$$
$$(take(E, X, Y) \wedge unknown(Y, cmpt710) \vee$$
$$take(E, X, Y_1) \wedge unknown(Y_1, cmpt720))$$

Suppose that the theory Γ consists of axioms:

$$\forall X.student(X) \equiv db\_student(X) \quad (1)$$
$$\forall X, E, Y, S.db\_course(Y, S) \wedge db\_student(X) \quad (2)$$
$$\rightarrow (take(E, X, Y) \equiv db\_take(E, X, Y))$$
$$\forall X, S.acourse(S) \rightarrow \quad (3)$$
$$(unknown(X, S) \equiv db\_course(X, S))$$
$$\forall E, X, Y.db\_take(E, X, Y) \rightarrow take(E, X, Y) \quad (4)$$

where *student*, *take* and *unknown* are lexical predicates and *db_student*, *db_course*, *db_take* are database predicates[1]. Also suppose, that the LDT declares as an assumption $acourse(X)$, which can be read as "X denotes a course".

Part of the conjunctive context associated with formula $take(E, X, Y)$ in F_{ling} is a formula (5).

$$student(X) \wedge unknown(Y, cmpt710) \quad (5)$$

From (1) and (3) of the theory Γ it follows that (5) implies the formula (6):

$$db\_student(X) \wedge db\_course(Y, cmpt710) \quad (6)$$

According to the translation rules of AET, axiom (2), and a logical consequence of a conjunctive context (6), the formula $take(E, X, Y)$ can be translated into formula (7)

$$db\_take(E, X, Y) \quad (7)$$

Formulas $student(X)$, $take(E, X, Y_1)$, $unknown(Y, cmpt710)$ and $unknown(Y_1, cmpt720)$ are translated similarly. Assuming *cmpt710* and *cmpt720* are courses, the input F_{ling} can be rewritten into F_{db} shown below.

$$\exists X, E, Y, Y_1 . db\_student(X) \wedge$$
$$(db\_take(E, X, Y) \wedge db\_course(Y, cmpt710) \vee$$
$$db\_take(E, X, Y_1) \wedge db\_course(Y_1, cmpt720))$$

So we can claim that F_{db} and F_{ling} are equivalent in the theory Γ under an assumption that *cmpt710* and *cmpt720* are courses.

RLDT. We shall constrain the expressive power of the LDT to suit tractability and efficiency requirements.

We assume that the input is a logical formula, whose predicates are input predicates. We assume that input predicates are not only lexical predicates, but also unresolved predicates used for, e.g., compound nominals (Alshawi, 1992), or for unknown words, as was demonstrated in the example above, or synonymous predicates that allow us to represent two or more different words with only one symbol.

The output will be a logical formula consisting of output predicates. We do not suppose that the output formula contains pure database predicates. However, we allow further translation of the output formula into database formulae using only existential conditional equivalences. The process can be implemented very efficiently, and does not affect selectional restrictions of the input language.

We assume that each atomic formula with input predicates can be translated into an atomic formula with output predicates. An RLDT therefore also contains a dictionary of atomic formulas that specifies which input atomic formulas can be translated into which output atomic formulas.

Existential equivalences in RLDT's logic will not be allowed. We also assume that F in the universal conditional equivalences is a conjunction of atomic formulas rather than arbitrary formula.

We demand that an RLDT be nonrecursive. Informally RLDT nonrecursivness means that for any set of facts A, if there is a Prolog-like derivation of an atomic formula F in the theory $\Gamma \cup A$, then there is a Prolog-like derivation of F without recursive calls.

3 The Normalization Process

Our basic idea is to preprocess the semantic information of RLDT to create patterns of possible conjunctive contexts for each lexical predicate. The result of the preprocessing is a *normalized* RLDT: the collection of the lexical predicates, their meanings in terms of the database, and the patterns of the conjunctive contexts.

First we introduce the term *(Nontrivial) Normal Conditional Equivalence with respect to an* RLDT T ((N)NCE(T)).

Definition: Let T be an RLDT and Γ be a logical part of T. The quadruple $(A, C, F_{input}, F_{output})$ is NCE(T) iff C is a conjunction of input atomic formulas of T, A is a conjunction of assumptions of T, and formulas

$$\forall(A \wedge C \rightarrow (F_{input} \equiv F_{output}))$$
$$\forall(A \wedge F_{output} \rightarrow F_{input})$$

are logical consequences of the theory Γ (we shall refer to the last condition as soundness of the NCE(T)). We shall call the quadruple $(A, C, F_{input}, F_{output})$ *nontrivial* NCE(T) (NNCE(T)) iff formula $C \wedge A$ does not imply truth of F_{output} in the theory Γ.

Informally it means that F_{input} can be rewritten to F_{output} if its conjunctive context implies A and does not imply the negation of C. (A, C) thus can be viewed as a pattern of conjunctive contexts, that justifies translation of F_{input} to F_{output}.

We allow RLDTs to form theory hierarchies, where parent theories can use results of their children's normalization process as their own logical part.

Given an RLDT T, for each pair consisting of the ground lexical atomic formula F_{input} and the ground database atomic formula F_{output} from the dictionary of T, we find the set S of conditions (A, C) such that $(A, C, F_{input}, F_{output})$ is NCE(T). We shall call the set of all such NCE(T)s *a normalized* RLDT.

If F_{input} and F_{output} contain constants that do not occur in the logic of RLDT, the generalization rule of FOL can be used to derive more general results by replacing the constants by unique variables.

[1]The predicate *unknown* will be discussed in the next section.

If the T does not contain negative horn clauses of the form $P \rightarrow notQ$ then the following completeness property can be proven:

> If $(A_1, C_1, F_{input}, F_{output})$ is NNCE(T) and S is a resulting set for the pair F_{input}, F_{output} then there are conditions (A, C) in S, such that $A \wedge C$ is weaker or equivalent to $A_1 \wedge C_1$.

The normalization process itself is based on SLD-resolution(Lloyd, 1987) which we have chosen because it is fast, sound and complete but still provides enough reasoning power.

Using the example from the previous section, the normalization algorithm when given the pairs ($student(a)$, $db\_student(a)$), ($unknown(a, b)$, $db\_course(a, b)$) and ($take(e, a, b)$, $db\_take(e, a, b)$) will produce the results $\{(true, true)\}$, $\{(acourse(b), true)\}$ and $\{(acourse(X), student(a) \wedge unknown(b, X)\}$ respectively.

4 The Construction of Selectional Restrictions

The *normalized* RLDT is used to construct selectional restrictions.

We assign the tags "thing" or "attribute" to argument positions of the lexical predicates according to what kind of restriction the predicate imposes on the referent at its argument position. If the predicate is a noun or the referent refers to an event, we assign the tag "thing". If the predicate explicitly specifies that the referent has some attribute - e.g. predicate $big(X)$ specifies the size of the thing referenced by X and predicate $take(\_, X, \_)$ specifies that the person referenced by X takes something - then we tag the argument position with "attribute".

The *normalized* RLDT allows us to compute which "things" can be combined with which "attributes". That is, we can determine which words can be modified or complemented by which other words.

We assume that the *normalized* RLDT has certain properties. Every NCE(T) that describes a translation of an "attribute" must also define a "thing" that constrains the same referent, e.g. the NCE(T) ($true$, $person(X) \wedge drives(E, X, Y)$, $big(Y)$, $db\_big\_car(Y)$) for translation of the predicate $big(Y)$ does not fulfil the requirement but NCE(T) ($true, car(Y), big(Y), db\_big\_car(Y)$) does.

We also assume that if a certain "thing" does not occur in any of the NCE(T)s that translates an "attribute" then the "thing" cannot be combined with the "attribute".

Using the example above and the assignments

$student(X)$	X is a "thing"
$unknown(X, S)$	X is a "thing"
$take(E, X, Y)$	E is a "thing", X and Y are "attributes"

we can infer that $student(X)$ can be combined with attribute $take(\_, X, \_)$ but cannot have an attribute $take(\_, \_, X)$.

To simplify results, we divide "attributes" into equivalence classes where two "attributes" are equivalent if both attributes are associated with the same set of "things" that the attributes can be combined with. We then assign a set of representatives from these classes to "things".

To be able to produce more precise results, we distinguish between two "attributes" that describe the same argument position of the same predicate according to the "thing" in the other "attribute" position of the predicate, when needed. Consider for example the preposition "on" as used in the phrases "on the table" or "on Monday". We handle the first argument position of a predicate $on(X, Y)$ associated with the condition $table(Y)$ as a different "attribute" as compared to the condition $monday(Y)$.

5 Discussion

Automating the process of finding selectional restrictions reduces NLI development time and may avoid errors introduced by hand-coding selectional restrictions. Althcugh the preprocessing is computationally intensive, it is done off-line during the delevopment of the NLI.

A similar approach was proposed in (Alshawi, 1992) but a different method was suggested. (Alshawi, 1992) derives selectional restrictions from the types associated with the database predicates, whereas our approach uses only the constraints that the RLDT imposes on the input language.

Future work will explore other uses of *normalized* RLDTs: to construct a sophisticated help system, to lexicalize some small database domains, and to develop more complex lexical entries. We shall also consider the possible uses of our work in general NLP.

Acknowledgments

The author would like to thank Fred Popowich and Dan Fass for their valuable discussion and suggestions. This work was partially supported by the Natural Sciences and Engineering Research Council of Canada under research grant OGP0041910, by the Institute for Robotics and Intelligent Systems, and by Faculty of Applied Sciences Graduate Fellowship at Simon Fraser University.

References

Alshawi, Hiyan, ed. 1992. *The Core Language Engine.* Cambridge, Massachusetts: The MIT Press.

Lloyd, John W., 1987. *Foundations of Logic Programming,* Second, Extended Edition, Springer-Verlag, New York.

Rayner, Manny, 1993. *Abductive Equivalential Translation and its application to Natural Language Database Interfacing.* Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden.