Lecture 13

Natural Language Interfaces

Parsing controlled English, NL disambiguation

Ras Bodik
Shaon Barman
Thibaud Hottelier

Hack Your Language!

CS164: Introduction to Programming Languages and Compilers, Spring 2012

UC Berkeley

Announcements

Midterm exam 1: in a week, in this room, usual time

- tomorrow's section is a review session

Final project proposal assigned today

- you will need to analyze the problem solved, propose your language, provide a few code examples

HW5 post mortem. What are the lessons?

Why the recommended hash-value solution failed.

We have two versions of the parser: Local; remote.

1) Staff solution tested on one. Lesson: test, test, test.
2) Client code (SDT) never to rely on internal semantics of semantics the implementation.

```
x = symVal()
hash(x)
y = symVal()
hash(y)
```

```
x = symVal()
hash(x)
x = symVal()
hash(x)
```

Applications of Natural Language Queries

past, current and future

Sample interaction with Loqui (research,1995)

> Who works on 3 projects?

B. Vandecapelle, C. Willems, D. Sedlock, J.L. Binot, L.Debille, ...

> Which of them are project leaders?

D. Sedlock, J.L. Binot

> Documents describing their projects?

Bim Loqui:"The LoquiNlidb", "Bim Loqui"

Mmi2:"TechnicalAnnex"

> How many of these projects do not finish before 1994?

Bim Loqui, Mmi2

> Are they led by JLB or DS?

The former.

Wolfram Alpha (commercial, today)

Deep dialogue systems (future phones)

You: *Which CS courses are offered in the fall?*

Siri: CS164, CS162, CS188, CS194.

How many do I need to graduate?

Two.

Tell me about the first one.

You don't want to know.

More future applications

Recall the puzzle from Lecture 1. We wrote a Prolog program that computed the solution.

It would be nice to have it automatically translated to Prolog from English.

9 • A New "Colored Hats" Problem

Three subjects—A, B, and C—were all perfect logicians. Each could instantly deduce all consequences of any set of premises. Also, each was aware that each of the others was a perfect logician. The three were shown seven stamps: two

A problem you'll see in PA6

Translate this puzzle to a Prolog program:

Facts: *Someone who lives in Dreadbury Mansion killed Aunt Agatha. Agatha, the butler, and Charles live in Dreadbury Mansion, and are the only people who live therein. A killer always hates his victim, and is never richer than his victim. Charles hates no one that Aunt Agatha hates. Agatha hates everyone except the butler. The butler hates everyone not richer than Aunt Agatha. The butler hates everyone Aunt Agatha hates. No one hates everyone. Agatha is not the butler.*

Question: *Who killed Aunt Agatha?*

Natural Language Interfaces to Databases

asking queries in informal English

First, what are formal query languages?

Relational database:

employees table

employee	**department**	**phone**
Thompson	sales	2317
Richardson	accounting	2554
. . .	. . .	. . .

departments table:

department	**manager**	**city**
sales	Ferguson	London
accounting	Richardson	Bristol
. . .	. . .	. . .

a SQL query:

```
SELECT employees table.employee, departments table.manager
FROM employees table, departments table
WHERE employees table.department = departments table.department
```

SQL vs. Prolog/Datalog

Find the manufactures of the beers that Joe sells:

SQL:

```
Beers(name, manf)
Sells(bar, beer, price)

SELECT manf
FROM Beers
WHERE name IN(
    SELECT beer
    FROM Sells
    WHERE bar = 'Joe''s Bar'
);
```

Datalog/Prolog:

```
joeSells(B) :- sells("Joe's Bar", B, P)
answer(M) :- joeSells(B), beers(B,M)
```

Formal query languages (Datalog)

Facts define a relational database:

employees table

employee	**department**	**phone**
Thompson	sales	2317
Richardson	accounting	2554
. . .	. . .	. . .

departments table:

department	**manager**	**city**
sales	Ferguson	London
accounting	Richardson	Bristol
. . .	. . .	. . .

Why natural language interfaces

No artificial (computer) language

- Use natural language and natural domain concepts.

Better for questions with negation and quantification

- *Which department has **no** programmers?*
- *Which company supplies **all** departments?*
- Corresponding SQL queries might be complex or tedious.

More advantages

Use of *anaphora* in discourse

An expression referring to another, in previous question:

> Is there a ship whose destination is unknown?

yes

> What is it?

What is [the ship whose destination is unknown]?

Saratoga

Disadvantages

Linguistic coverage not obvious

It may be hard to understand what subset of natural language the system understands than it is to learn a formal query language.

Ex: the ability to answer

"What are the capitals of countries bordering the Baltic and bordering Sweden?"

may suggest that the system can answer

"What are the capitals of countries bordering the Baltic and Sweden?"

Compare with formal query languages

It is usually clear which programs you can write. Any syntactically valid query can usually be answered.

More disadvantages

Linguistic vs. conceptual failures

On failure, the user may try to rephrase a question in different *linguistic* terms (eg, synonyms, tense, syntax) while the failure may be that the system does not understand a particular concept (eg, multi-city trip)

A solution:

Need for diagnostics: unknown word, syntax too complex, unknown concept, etc.

A user study: SQL vs Natural Language Queries

NL was found better when

- multiple tables had to be combined in the query and
- queries included negation
- the query did not resemble one encountered in training

Technical Challenges

Modifier attachment

Consider

List all employees in the company with a driver's license

Difficult for a system to distinguish between

List all (employees (in the company) (with a driving license))

and

List all (employees (in the company (with a driving license)))

A human can answer given his background

But replace "driver's license" with "export license"

Such can be belong to both a person and the company

A solution

Choose rightmost association

List an employee

who was hired by a recruiter

whose salary is greater than $3,000

But some ambiguity is truly ambiguous

List all employees in the division making shoes

Quantifier scoping

Consider

Has every student taken some course?

Two readings:

1. Check that ∀*student* ∃*course taken(student, course)*
2. Check that ∃*course* ∀*student taken(student, course)*

Usually prefer left-to-right ordering of quantifiers

We could also give precedence to the quantifiers

Interesting question is whether we can write a %dprec grammar that parses the sentence into reading 2.

Conjunction and Disjunction

Consider

List all applicants who live in California and Arizona

"*And*" often means disjunction (*or*)

This ambiguity is hard to resolve. Consider

Which minority and female applicants know Fortran?

and could mean both *and* or *or*. Both readings are meaningful.

A solution to ambiguity: system answers both queries

The nominal compound problem

city department

a department located in a city, or a department responsible for a city.

research department

is probably a department carrying out research.

research system

is probably a system used in research, it is not a system carrying out research.

Solution

Declare the meaning during configuration phase.

This is when the domain knowledge for the particular application is provided.

This problem illustrates that portability (from domain to domain) is a big open problem of NL systems.

Elliptical sentences

Does the highest paid female manager have any degrees from Harvard?

Yes, 1.

How about MIT?

No, none.

Who is the manager of the largest department?

Name	Dept.	Count
Patterson	045	40

The smallest department?

Name	Dept.	Count
Saavedra	011	2

Continued …

"How about MIT?" and "The smallest department?" are elliptical sentences. A shopping example:

What is the price of the three largest single port fixed media disks?

Speed?

Two smallest?

How about the price of the two smallest?

Also the smallest with two ports?

Speed with two ports?

Parsing Natural Language

Grammars for NL

Context free grammars have been shown not a great fit for arbitrary text in a natural language.

- Semantic and probabilistic grammars are used instead

We'll get by with context free grammars

- By restricting which sentences we can handle

NL Parsing example

S -> NP VP

VP -> V PP

| V NP

PP -> Prep NP

NP -> N

| Det N

| N N

N -> arrow | banana | fruit | flies | time

V -> flies | like

Det -> a | an

Prep -> like

time flies like an arrow

fruit flies like a banana

NL Parsing example: Find parse trees

time flies like an arrow

fruit flies like a banana

Translating NL to Prolog queries

We want to translate NL sentence to Prolog

Translate

What is the capital of each country bordering Greece?

to

```
answer(Capital, Country):-
        is_country(Country),
        borders(Country, greece),
        capital_of(Capital, Country).
```

Let's pick a specific problem

S → NP VP

NP → Det N

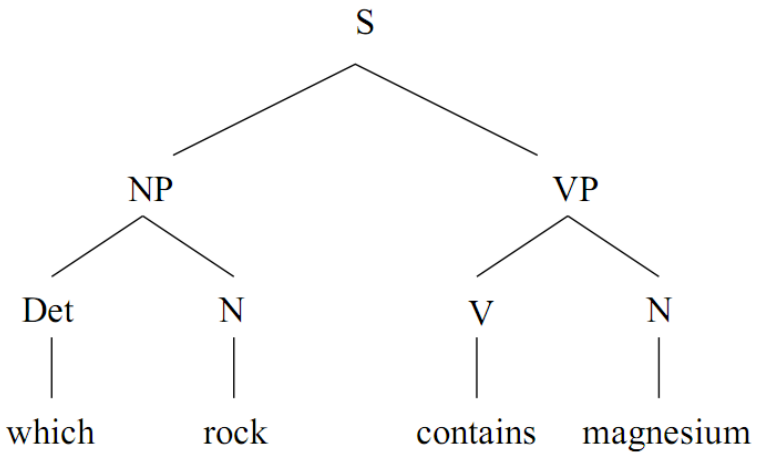
Det → "what" | "which"

N → "rock" | "specimen" | "magnesium" | "radiation" | "light"

VP → V N

V → "contains" | "emits"

Translate parse to Prolog-like query



Prolog query:

Restricted NL Syntax

Facilitates disambiguation. One restricted form:

what

conjoined noun phrases

nested relative clauses

conjoined relative clauses

Example:

what are
the names, ids, and categories of the employees (1)
who are assigned schedules (2)
that include appointments (3)
that are executions of orders (4)
whose addresses contain 'maple' and (5)
whose dates are later than 12/15/83 and (6)
whose statuses are other than 'comp' (7)

Restricted NL Syntax

Facilitates disambiguation. One restricted form:

what

conjoined noun phrases

nested relative clauses

conjoined relative clauses

Not in this restricted form:

what are
the addresses of the appointments (1)
that are included in schedules (2)
whose call times are before 11:30 and (3)
that are executions of orders (4)
whose statuses are other than 'comp' (5)

Reading

Required:

Natural Language Interfaces to Databases – An Introduction, I. Androutsopoulos, G.D. Ritchie, P. Thanisch