CSE6339 3.0 Introduction to Computational Linguistics
Instructor: Nick Cercone – 3050 LAS – nick@cse.yorku.ca
Mondays, Wednesdays 10:00-11:20 – North Ross 836A
Winter Semester, 2014

Introduction to the course

Thoughts about problems about thoughts

◂ Abstract



## Less Abstract

" Language creates special worlds. Most languages are inaccessible to most of us most of the time. Add computers to the mix and ask: have we any hope of automating language understanding? Is the problem one of representation, organization, processing, or what?

Find out!"



## A Few (not so) Random Thoughts

- Language creates special worlds. It does not matter that the language of the eskimo and inuit describing the many states of snow is inaccessible, we do not feel a sense of loss because what holds for languages holds for many of our life experiences most generally also.

- It is not so much why something happens, or how it occurs, as it is why we perceive things to be or how we plan activities to occur or even what we ponder in between all other thoughts that hold our interest.



- However, for most of us, the world is a world of matter - wysiwyg. The superiority of physics to, say, interpersonal communication, massage, etc. derives from the assumption that if we are able to explain the physical, we may be in a position of explaining everything.

- From a dog's point of view, the world, I suppose, resembles largely a malign kennel deliberately arranged to thwart his pleasure - one damn rule after another: do not urinate on the sidewalk, do not drink from the toilet, fetch, sit. No wonder the poor beasts slink so much and are so sneaky



## Domain Knowledge is required to interpret queries

Many places require domain knowledge to interpret queries. The place where the need is most evident is in resolving references made in the query. For example, the following sequence of sentences could easily arise in the academic advice domain:

- Could I have a report on the students who are registered in Computer Science? [2]
- Would you summarise it for me? [3]
- Which students dropped out? [4]
- Which of them are still mentioned in the report? [5]

- resolve the pronouns it in [3] as referring to the concept represented by the report in [2], and them in [5] as referring to the students in [4].
- Note that the reference is actually to the answer produced by the system to the user's queries, rather than to the concept in the query itself.
- A non-pronominal anaphoric reference must be resolved - report in [5] refers to the report produced in response to [2].



- Another place where the need for domain knowledge is evident is in discovering and resolving ambiguities which arise in the interpretation of a query. Ellipsis frequently occurs during database access, especially given the tediousness of typing questions rather than verbalizing them. Ellipsis can often be handled by syntactic mechanisms. Tracking the focus of attention of the user as he/she asks a series of questions is sometimes important in natural language database interfaces, as the following sequence of queries illustrates:

- Who had an honours average last year? [6]
- Which of them had the highest average? [7]
- Was there anybody else close? [8]

To answer [6], the system must search through the set of all students. In [7] the focus is narrowed to the set of students who had an honours average last year. Query [8] also assumes a focus set of students with an honours average last year. Note how differently [8] is interpreted if it were asked after [6] without the intervention of [7].

## Domain Knowledge is required to answer queries

- To reflect *general rules*, rather than things which happen to be accidentally true of current data. Even with general rules, it may not be possible to respond "yes" or "no". Knowledge of any domain will not be *complete*. Facts will be missing, errors will occur in the data, some rules won't be known, and so on.
- Systems can have several kinds of "ignorance", e.g., when the system hasn't got all of the facts. Consider:
  - Have any students ever failed CS 882?

- According to records it may be that no students have failed CS 882, but records may only go back three years. It is important that the system be able to encode knowledge about what it knows and doesn't know.
- Keeping track of *exceptions* to general rules, e.g.,
  - Do all professors in CS have a Ph.D. degree?
- *summary response generation.*
- hidden *time dependencies*, e.g., "Is professor Jones available for advising"-"No, shall I tell when he is?";
- answers related to *hypothetical queries*, e.g., "If I gave Mark an 85% what would the class average be?";

- references to *system generated concepts*
- practical considerations regarding *updates*
- requests concerning the *database structure*, e.g., "Who is teaching which courses this fall?"

Consider in greater detail tracking exceptions to general rules, e.g., to answer

Do all profs in the CS department have a Ph.D.?

the system may find a general rule

$$(\forall x)(\text{professor}(x) \Rightarrow \text{has-PhD}(x))$$

which would allow it to respond "Yes". Such universal rules are fine, in theory, but in knowledge bases representing real world domains, there are exceptions. Some exceptions will represent special cases; others may represent errors in data. In either case, the system must discover exceptions and produce extensive responses based on them.

For example, All professors do except Professor Jones. It may even be useful to try to explain why the exception exists in order to help the user understand the subtleties of the general rule. Thus, an answer like All professors do except Professor Jones. She was hired before the requirement that all professors have Ph.D. degrees came into effect. might be more appropriate.

Another place where a knowledge base may be needed rather than simply a database in order to successfully answer a query is in the area of summary response generation. Instead of a long enumerative response to the query Which courses may a CS student take in addition to the courses in his or her major? (for example, "Classics 100, Classics 101, Biology 110, ...."), a summarized version of the response (for example, Other courses in Arts and Science) would be more appropriate.



It is fairly easy to imagine how this might be done without recourse to anything beyond a relational database. In this case the database could be searched for all courses taken by computer science students, and then the faculty which offers any non-computer courses read. If the entry for the faculty attribute for each such course is "Arts and Science", then the summary response can be given.

Other considerations requiring domain knowledge to answer queries include hidden time dependencies, e.g., "Is professor Jones available for advising" may require a monitored response such as "No, shall I tell you when he is?"; answers related to hypothetical queries, e.g., "If I gave Mark an 85% what would the class average be?"; references to system generated concepts, e.g., the concept "members of the graduate faculty" may be a system generated concept, subsequently referenced, which answers the query "Who is capable of supervising graduate students"; practical considerations regarding updates; requests concerning the database structure, for example, "Who is teaching which courses this fall?"; and the generation of extended responses.

## Modeling the User is Important

Domain knowledge is not enough; we also need to be able to model the user.

The database community has recognized the need for different user views corresponding to different users. User views allow the database system to restrict access available to a given class of users. A similar use should be made of user models. Moreover, a user model would be helpful in disambiguating the user's utterances in several ways: it would restrict the number of possible interpretations of the user's query; it would generally cut

down on the combinatorics involved in accessing the knowledge base; and it is far easier to decide how to phrase the response for a class of users.

Grice elaborates four basic tenets of the cooperative principle which a speaker should obey: (i) the maxim of quantity: be as informative as required, but no more so; (ii) the maxim of quality: do not make a contribution which one believes to be false or for which adequate evidence is lacking; (iii) the maxim of relation: be relevant; and (iv) the maxim of manner: avoid obscurity of expression, avoid ambiguity, be brief.

One obvious place the need for user modeling arises is how to recognize presuppositions, and correct any false presuppositions if necessary. To illustrate, consider the following:

How many undergraduates passed CS 859 last year?

There is a presupposition that undergraduates can take CS 859. The user believes this presupposition, or s/he wouldn't have asked the question in this fashion. An answer of "None" does nothing to correct this false presupposition. The system violates the maxim of quality. A much better answer would be Undergraduates are not allowed to take Cmpt. 859.

The "standard" procedure for handling presuppositions divides into two steps: find the presuppositions, and recognize any that are false so they can be corrected. When correcting false presuppositions other false presuppositions should not be established (e.g., if CS 859 were not offered, the answer above would still contain a false presupposition). An answer generation/ presupposition correction cycle must be carried out before producing the response.

Alternative methods whereby the presuppositions of the answer, rather than those of the question, are computed would allow the system to behave more reliably when confronted with presuppositions that it cannot prove true or false in the database.

False presupposition correction needs domain knowledge to determine if a presupposition is consistent with current data.

Developing automatic procedures which generate concept hierarchies to help respond appropriately to null database responses resulting from false user presuppositions would be useful. For example, depending on which part of the query What grade did John receive in Math 101? resulted in the null response (grade not yet posted, Math 101 not a course or not offered, John not a student, etc.) an appropriate response is generated ("Grades have not yet been posted. Would you like me to inform you when they are?", etc.).

Recognizing the implication of multiple user goals, keeping track of user's knowledge, answering why questions, and making scalar implicatures all demand some degree of user modeling and extended database knowledge. For example, the question Is Smedley a student? could be answered "Yes", but more informatively "Yes. A grad student". This extended response is a scalar implicature, an inference made based on some scale, i.e., "grad student" is a subtype of "student". The scalar implicature augments the answer with a reference to the higher value.



## Representation and Organization

Is representation important and why?

What should be represented?

Why should what is represented be represented?

How do (should) we view the universe?

What about organization?



## A good representation engenders a good solution

Data structure

Difficult integral

What is the inverse of a cognitive transformation?

## States, events, actions, cases, causes, and intentions

State attribution – ascribing a modifier to an object or set of objects at some time

Events – ascribing a change of state to an object or set of objects at some time

Actions – proper subclass of events?? Or existence of a situation which tends to produce change, and all actual changes must be inferred??

Cases – semantic primitive or not?

Causes – where do we draw the line between what an "agent" does and what an "agent" causes?

Intentions – add supplementary information to "what was said" and confuses representation.

## Example problem 1

John was hurting Mary by pulling her hair.

John was dragging Mary by pulling her hair.

Representing hurting as a state precludes John as an agent whereas dragging represented as an action has John as an agent. Just as we are compelled to regard certain ongoing activities as instigated by somebody or something, we are denied the option of regarding certain actions as having an agent.

## Example problem 2

Phrases ostensibly expressing instrumental actions often express no more than causation. An example is the "by" clause in The effluents were killing the fish by raising the temperature of the water.

When there is a difference, it lies in the intimation of purposive causation. In John woke Mary by blowing his trumpet purposive causation is expressed, while in Mary woke up because John was blowing his trumpet it is not.

This example clearly shows that instrumental relations amount to causal relations supplemented by intentional states.



## Adjectives and relative terms

John is a big man.

Certainly we are establishing:

John is a man

John's size is greater than {a typical man's size, a typical value of size for men, something else} – which is it and why?

John is the perfect man.

(∀P) {[(∀x)[man(x) & P(x) nec=> y-approves [P(x)]]] => P(John)}



Where y is the speaker. The logical formula reads "John has all properties such that y would approve of any man's having them." We can then easily formulate an expression for "someone is not a perfect man" by utilizing the logical formula with the existential quantifier (∃)~ and replacing P(John) with P(z). Clearly the method of handling comparatives adjectives such as big, tall, and so on does not work in this situation.

## Adverbials

John is running quickly on his hands and knees
John is running quickly on the moon
John is running quickly in Chile
The cheetah is running quickly in the dense forest
The cheetah is running quickly on the plain

It appears that the context which determines the meaning of an adverbial modifier cannot be circumscribed once and for all. Apply the typical value functor to the lambda abstracted predication, thus

## (λx) [cheetah(x) & running(x) & in-dense-forests(x)]

Then we show the explicit relationship between the speed of the cheetah's running as compared to the typical value of speed for something that is running, a cheetah, and in dense forests.

The typical value functor does not presume the existence of a reference set.

## Referential Opacity

John wants to marry the prettiest blonde.

## Vagaries of Reference

The President is elected every four years

## Time

Johnny caught a tadpole and named him Fred. Fred was released into the pond a year later. What was Fred?

## Some NLP Applications

Natural language interfaces to databases

Natural language interfaces to search engines

Generate and Repair Machine Translation (GRMT)



## Conclusions

- All those who believe in telekinesis, raise my hand.
- OK, so what's the speed of dark?
- I intend to live forever - so far, so good.
- 24 hours in a day … 24 beers in a case; coincidence?
- What happens if you get scared half to death twice?
- Plan to be spontaneous tomorrow.
- 42.7 percent of all statistics are made up on the spot.