Rule-based computational semantics

Analysing the meaning of sentences in S. Bird, E. Klein, and E. Loper. Natural language processing with Python. O'Reilly, 2009.

Questions to discuss

- What are challenges of translating natural language to logic (in general)?
- Different logics are formal languages with specific properties: what features of natural language they can represent and what are their limitations?
- How is underspecification of quantifier scope implemented in Cooper storage?
- How about other forms of underspecification in natural language, e.g. lexical ambiguity?
- Why do we need lambda calculus?
- How can we use model builders and theorem provers (computational tools) to check validity of arguments?
- Do humans also reason this way?
- Overall, what aspects of natural language semantics are treated well with these methods and what aspects are not captured?
- What NLP applications benefit from this approach?

Simon's notes

- A lot is hard-wired into the grammar; composing semantic meaning parallel to composition of syntactic constituents

- Formal language and natural language: "and" vs `AND`, "or" vs `OR`

- Logical equivalence

- Pragmatic implicature: `all x.(dog(x) -> disappear(x))`; also true if there is no dog in the model

- More pragmatic constraints required are demonstrated when demonstrating model building: e.g. to conclude from the first premise that Adam loves Eve: sets of men and women are disjoint; Eve is the only woman

  ```
  exists y. (woman(y) & all x. (man(x) -> love(x,y)))
  man(adam)
  woman(eve)
  all x. (man(x) -> -woman(x))
  exists y. all x. (woman(x) -> (x = y))
  Goal: love(adam,eve)
  ```

- Quantifier scope ambiguity: Everyone likes someone

  - `all x.(person(x) -> exists y.(person(y) & admire(x,y)))`

- exists y.(person(y) & all x.(person(x) -> admire(x,y)))

- Consistent vs inconsistent propositions (i.e. those that are not contradictory); can build a model of consistent descriptions
- Can computer understand language? Turing test: natural language understanding and generation at the level of *observable behaviour*
- Theorem proving: can a prove (a proof goal) be derived by a finite sequence of inference steps from a list of assumed formulas?; *a valid argument*
- Model building: model building tries to create a new model, given some set of sentences
- Lambda calculus and transitive verbs
  - The verb phrases are: \y.exists x.(dog(x) & chase(y, x))
  - Take out the verb: \P.exists x.(dog(x) & P(x))(\z.chase(y, z))
  - Replace the NP part with a variable X: \P.exists x.(dog(x) & P(x))
  - X(\z.chase(y, z))
  - Turn this into a function that will take an NP: \X y.X(\x.chase(y, x)) <<<e,t>,t>,<e,t>>
- Inference
  - Model building: negate the conclusion and add it to the premises; if the we can build a model with a negated conclusion then it means that there exists a model where premises are true and conclusion is false; hence the conclusion cannot follow from the premises; if we cannot build such a model then the conclusion is true; *hence fast when the conclusion does not follow*
  - Theorem proving: negate the conclusion and add it to the premises, if we find a contradiction then it must be the cases that the conclusion follows; if there is no contradiction among premises the negated conclusion is consistent with premises; *hence fast when the conclusion follows*
- Cooper storage
  - Make the semantic representations underspecifed by removing quantified expressions and replacing them with variables of type e
  - core = <chase(z1,z2)>
  - store = (bo(\P.all x.(girl(x) -> P(x)),z1), bo(\P.exists x.(dog(x) & P(x)),z2))
- Extensions to FOL required for
  - events, tense and aspect;
  - semantic roles;

- generalized quantifiers such as *most*;
- intensional constructions involving, for example, verbs like *may* and *believe*.
- (i) and (ii) can be expressed in FOL; (iii) and (iV) require extensions

From the class, VT23

- Program complexity for rule-based systems and processing time;
  - but training ANNs is also time consuming
  - humans writing grammars also take time
  - https://en.wikiquote.org/wiki/Fred_Jelinek
- Granulairty of representations required
- Interpretability
  - sensitive applications
- The use of formal representations
  - context specific: sensitivity of appliations
  - low-resource scenarios where linguistic inference is required
  - can use rules to verify what has been learned
- Reliance on syntactic parsing and syntax
  - inefficient?
  - what happenes if we encode everything in lambda calculus, the grammaer would likely be more complex
  - allows us to map different sentences to canonical semantic representations
  - disambiguation: syntactic and semantic explosion of readings
- Non-compositional expressions
  - idioms
  - can decide the granualrity of lambda applications

From the class, VT22

- Efficiency of databases and queries
  - we need a complete database, understand the problem
  - separation of data and representations and language; SQL language is connected to the storage; portability of systems
  - are relational databases enough?
- Translating NLP to logic, limitations of logic to cover linguistic constructions
  - I think Gothenburg is... I think that P where P is a proposition
  - Natural language is not "grammatical": sarcasm, figurative language, idioms (sequence of words or a single word); lexical semantics is not dealt with; deictic representations, spatial relations
  - Why FOL? Tools available.
  - Logic and AI: SHRDLU, Winograd, https://en.wikipedia.org/wiki/SHRDLU

  - NLU and NLG
  - We don't say everything we mean? Language in context
- Ambiguities: lexical ambiguities, can they be solved by logic; pragmatics and slang
  - resolving lexical ambiguities: resolving through word contexts, e.g. rock (music), rock (stone); also depending on the context: in the Spotify app
  - words in different social contexts: changing language; updated contexts in the database?
- Lambda calculus: how to do it for non-European languages and dialects
  - ambiguity
  - apply to corpora and study semantic relations in spoken language
  - cross-linguistic semantic representations
- Computational tools for reasoning and how do they relate to human reasoning?
- Cooper storage?
- From logic to applications:
  - a QA system for family members
  - solving logic riddles, games, https://www.uni-bamberg.de/en/sme/teaching/bambirds/
  - tools and representations: what is good for different tasks? Where do we find resources?

From the class, VT21

girl(x) & walk (x)

all x exits y.girl(x) & sleepy(y) -> walk(x) & likes(x,y)

T

F

->

T F F

F T T

F T F

T T T