SEMINAR 1 SUMMARY REPORT:
RULE-BASED COMPUTATIONAL SEMANTICS

Group 4

SEMINAR 1 SUMMARY REPORT:
RULE-BASED COMPUTATIONAL SEMANTICS

This report has been compiled based on collective notes from our group discussion sessions.

1) Main Points (Thoughts and Conclusions)

Challenges of translating natural language to logic

One of the primary challenges in translating natural language into logic lies in addressing its ambiguity. One example is the issue of structural ambiguity, particularly in relation to quantifier scopes, where a single sentence may produce multiple possible logical/valid interpretations. Similarly, another issue is lexical ambiguity, where words that have various meanings (homonyms) can complicate logical representation. Pragmatic elements that cannot be captured with logical operators include, for example, the plural "some" (which implies "more than one"), which cannot be precisely captured with either existential quantification (∃) or negated universal quantification (¬∀). An additional point that was lightly touched upon during the seminar was about how non-literal language such as idioms, metaphors, tense, deixis, and probabilistic elements can pose challenges for logical translation.

A natural language feature that logic struggles to capture well are presuppositions. An example of this is the sentence "Every dog disappeared" which implies that dogs exist, but in logic ∀X. (dog(x)-> disappear(x)) is vacuously true even if there are no dogs. Another issue that was brought up was about context-dependent meanings and how discourse context can affect interpretation which poses the same challenges as metaphorical and figurative expressions or vagueness.

Contrasts between human reasoning and computational tools (model builders, theorem provers) and argument proving

A significant portion of the discussion focused on understanding and comparing the how human reasoning differed from theorem provers and model builders. A concluding understanding of these two was that theorem provers work strictly from the facts and rules provided to it. Additionally, they don't add anything more than what is explicitly stated, always produce valid conclusions, and make syntactic proofs. On the other hand, model builders assign meaning by looking for possible world/model where the assignment doesn't hold. It also tries to find counter examples where premises are true but the conclusion is false. In contrast, human reasoning incorporates additional contexts and personal experience and is this more flexible and more likely to be less consistent. As a result, it is prone to accommodate arguments that are "true" (accepting as true for the sake of reasoning) without them being factually provable. Eg: matters of personal opinion when deciding between taking the longer route because it is scenic versus a shorter route. What is inferred as correct is completely dependant on what the deciders hold as truth now (scenic view vs saving time). Logic that can encode this will need to keep track of what is being accepted as true in each state of the discourse.

Hence concluding that human inferences are often based on individual priorities (example of scenic route) and subjective values. For example:

Rule: $\forall x \forall y.(loves(x,y) \rightarrow loves(y,x))$ (love is mutual)

Fact: loves(Cristina, John)

Since we use rule and fact for the theorem proves it concludes loves (John, Cristina). On the other hand, humans infer with more information, emotions and life experience that lead to different conclusions.

The aim when using theorem provers is to show that the conclusion logically follows from the premises. The prover return a "True" result means the argument is valid because the conclusion can be derived from the premises – there's no possible situation where the premises are true and conclusion is false. Contrastingly, model builders try to find a counterexample – a situation or "model" where all the premises are true but the conclusion is false. If such case is found, it means the argument is not valid, because the conclusion doesn't necessarily follow from the premises.

Cooper Storage for Quantifier scope and Underspecification

Quite some time in the seminar meeting was spent on understanding cooper storage. Cooper storage address quantifier scope ambiguity, it does this by: separating the core (predicate) from the noun phrases and then depending on the order in which we plugin the noun phrases to the core, we can get different meanings. For example: the sentence "Every man loves a woman". The core would be: λx.λy.loves(x,y) ; the storage would contain: λP.∀x.(man(x) -> P(x)) and λQ.∃y(woman(y) ∧ Q(y)). Retrieving these in different orders can produce two distinct interpretations:

i) $\exists y(\text{woman}(y) \wedge \forall x(\text{man}(x) \rightarrow \text{loves}(x, y)))$

(meaning "There is a woman whom every man loves".)

ii) $\forall x(\text{man}(x) \rightarrow \exists y(\text{woman}(y) \wedge \text{loves}(x, y)))$

(meaning "Every man loves some woman (possibly different women)")

Lastly, it was suggested that for forms of underspecification such as lexical ambiguity, additional NLP techniques such as word sense disambiguation via WordNet and probabilistic modelling of most likely readings given in contexts would be needed (eg: Veterbi algorithm where emanations are encodings into some logic.

Lambda Calculus

It is essential as it provides abstraction for creating reuseable expressions and functions, and composition for building complex meanings from simpler part. It allows for nexting and higher-order structures, for example:

Very tall:

tall=λx.tall(x)

very=λA.λx.very(A(x))

very(tall) = (λA.λx.very(A(x))) (λx.tall(x))

very tall=λx.very(tall(x))

Resuability of the function – rather than having separate expressions for "John walks" and "Simon walks", lambda calculus lets us use a single function λx.walks(x) and apply it to different subjects through β-reduction.

NLP Applications of logical semantics

The approach can be applied in NLP areas such as question answering systems, natural language understanding (NLU) for intent recognition, dialogue systems (especially

for tracking logical relationships and dialogue state), and analysing complex linguistic structures with multiple negations or conditions etc.

When statistical approaches require more data than available, it is more efficient to combine rule-based methods. For example, for Denis' chess game application (for Dialogue Systems course) the NLU service could not distinguish between targetSquare and originSquare entities, so he merged them into one, and instead applied post processing to the result that would infer the specific square in the utterance depending on the number of square entities within the utterance, on the presence of a chessPiece entity and on precedence of words such as "from", "on" and "to". Likewise in order to distinguish the intent to castle, there wasn't enough variability in natural language: long castle, queenside castle, short castle, kingside castle (in reverse these are 8 possible ways to express an intent to castle, – still too little and too noisy to distinguish even one type of castling move from the other). Consequently, Sharon has encountered similar problems with the application of NLU services during her work in the Dialogue System course.

2) Clarification points

In section 4.3 "Quantified NPs," there is a need for clarification on how Type-Raising of quantified NPs work, as this concept was not fully understood. Additionally, regarding Cooper storage, further clarification is needed on what exactly constitutes a "core" semantic representation along with a list of binding operators.

Models like GPT-4 sometimes generates fluent but logically incorrect answers because they lack explicit semantic reasoning. How can computational semantics, and tools from this chapter such as lambda calculus etc. help solve this problem?

3) Other details of discussion

We had examined the cooper storage approach in detail during the meeting with the following example: "Every man loves a woman".

Core: the main argument
Storage: quantifiers
"Every man loves a woman"
Storage
λP.∀x.(man(x) ->P(x))
λQ.∃y(woman(y) ∧ Q(y))
Core
λx.λy.loves(x,y
store = [
(λP.∀x(man(x) → P(x)), x), # "Every man"
(λQ.∃y(woman(y) ∧ Q(y)), y) # "a woman"
]
Retrieve quantifiers from the store in different orders to generate both readings:
Reading 1: Universal-Existential (∀∃)
Apply "every man" first:
Substitute P in λP.∀x(...) with λx.loves(x, y):

∀x(man(x) → loves(x, y))
Apply "a woman" next:
Substitute Q in λQ.∃y(...) with λy.∀x(man(x) → loves(x, y)):
∃y(woman(y) ∧ ∀x(man(x) → loves(x, y))
Final: ∃y(woman(y) ∧ ∀x(man(x) → loves(x, y)))
(There is a woman whom every man loves.)
Reading 2: Existential-Universal (∃∀)
Apply "a woman" first:
Substitute Q in λQ.∃y(...) with λy.loves(x, y):
∃y(woman(y) ∧ loves(x, y))
Apply "every man" next:
Substitute P in λP.∀x(...) with λx.∃y(woman(y) ∧ loves(x, y)):
∀x(man(x) → ∃y(woman(y) ∧ loves(x, y)))
Final: ∀x(man(x) → ∃y(woman(y) ∧ loves(x, y)))

Overall, the reading was highly informative and practical, offering hands-on approaches to learning with the NLTK library. The “further reading” sections at the end of the chapters helpfully acknowledged the limitations of the paper and guide readers towards additional resources.

Contribution description:
Denis: contributed extensively to large portion of the discussion, particularly given his knowledge about logic. Contributed many examples regarding Cooper storage and challenges about ambiguity. Answered all the discussion questions
Cristina: contributed extensively to large portion of the discussion, focusing particularly on comparison between theorem provers and model builders and how these tools contrast with human reasoning. Answered all the discussion questions
Huoyuan: limited discussion due to different educational background and new exposure to linguistic material. Brought forward clarification questions and how she engaged with the text.
Sharon: made some contributions but limited due to finding the text a bit challenging and struggle with understanding certain concepts. Brought forward certain Answered 3-4 out of all 8 questions. Responsible for writing the summary report.