Knowledge Engineering in the Long Game of Artificial Intelligence

The Case of Speech Acts

Marjorie McShane, Jesse English, Sergei Nirenburg
Cognitive Science Department, Rensselaer Polytechnic Institute

11/17/21

#paper04-mcshane

The Goal: Holistic Knowledge Engineering

- KE is expensive – but it also enables a level of understanding that ML does not
- Siloed and domain-specific KE is useful for producing application results, but doesn't help the long goal of AI
- To address human-level reasoning, we need Holistic Knowledge Engineering:
  - KE that foresees a wide range of agent capabilities
  - KE that supports multiple domains
  - KE that works in a variety of applications

A Brief Aside – OntoAgent [1]

- Our agent theory and architecture
- OntoAgent is knowledge-centric
- Expects large-scale knowledge
- Expects all knowledge, perceptions, and experiences to be ontologically grounded in a uniform metalanguage

A Brief Aside – OntoAgent [2]

- OntoAgent contains the following modules:
- Perception + Interpretation (something perceived, voice or text, vision, etc. must be interpreted into the metalanguage; e.g., an NLU system converts text into meaning)
- Attention + Reasoning (all inputs and thoughts must be attended or not, inputs and thoughts can be reasoned over)
- Action Specification + Rendering (decisions to take action must be made, and actions must be realized in the world; e.g., vocalized, motorized, etc.)
- Memory and Knowledge management are a must (everything is in the same metalanguage!)

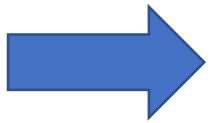
OntoAgent's NLU System

- Uses a syntactic/semantic lexicon to bridge lexical inputs and ontological knowledge
- Produces a TMR (Text Meaning Representation) that is in the uniform metalanguage of the agent's memory
- The TMR represents unambiguous meaning of the input text
- The agent can reason over the TMR

NLU Example – “Did you eat a cookie?”

```
REQUEST-INFO
is-a              COMMUNICATION
agent             HUMAN
beneficiary       HUMAN
…
```

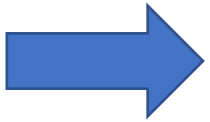
```
DO-AUX47
SYN
  ROOT
  SUBJ   $var1
  VP     $var2
  PUNCT  ?
SEM
  REQUEST-INFO
    AGENT        *speaker*
    BENEFICIARY  $var1
    THEME        $var2
```



```
REQUEST-INFO-1
    agent              HUMAN-1
    beneficiary        HUMAN-2
    theme              INGEST-1
INGEST-1
    agent              HUMAN-2
    theme              COOKIE-1
```

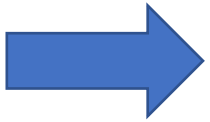
NLU Example – “Yes.”

```
YES-ADV1
SYN
  ROOT
SEM
  ACCEPT
    agent       *speaker*
```



```
ACCEPT-1
    agent                HUMAN-2
```

```
YES-ADV2
SYN
  ROOT
SEM
  RESPOND-TO-REQUEST-INFO
    agent       *speaker*
    VALUE       positive
    THEME
        <the proposition
        of the previous text that
        anticipates a response>
```



```
RESPOND-TO-REQUEST-INFO-1
    agent                HUMAN-2
    value                positive
    theme                ???
```

NLU Example – “Yes.”

```
ACCEPT-1
    agent                HUMAN-2
```

?

```
RESPOND-TO-REQUEST-INFO-1
    agent                HUMAN-2
    value                positive
    theme                ???
```

Well… that’s not very helpful, is it?

- A dialog model can help us better understand what “Yes.” means here!
- But… a dialog model isn’t exactly taking the long view for AI. What do we do?

```
YES-ADV2
SYN
  ROOT
SEM
  RESPOND-TO-REQUEST-INFO
    agent       *speaker*
    VALUE       positive
    THEME
        <the proposition
        of the previous text that
        anticipates a response>
```

Introducing Scriptlets [1]

- Concepts: atomic, general-purpose units of knowledge (e.g., `DOG` or `INGEST`)
- Scripts: dense and often domain-specific encapsulations of knowledge about all behavior in an event, for example:
  - To travel you must
    - Buy a plane ticket (how, with what resources, from where?)
    - Pack your bags (with what, and why? into what container?)
    - Drive to the airport (directions? take a cab? when to leave?)
    - Go through security…
    - And so forth

Introducing Scriptlets [2]

- In between the relatively isolated and clean concept definition, and the dense, complex, and hyper-specific script definition lies the scriptlet
- A scriptlet is an augmented concept that contains private references to concept instances so that it can, in a domain-agnostic way, define interrelated behaviors in a lightweight fashion
- Example: “when someone `ORDERs` lunch, they typically `INGEST` it”
- Example: “when something moving at X speed `ACCELERATEs`, it is now at Y speed, where Y > X”

Scriptlets are small but are key to the long game

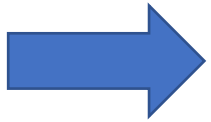
- They introduce a minimal amount of knowledge to enable reasoning
- They are not bound by weighty domain-specific scripts (but can be used by them)
- This is psychologically plausible:
  - Ask someone what happens “after you pick up a hammer” and you’ll likely get a variation of “you hit a nail”; a heavy a complex script isn’t required to be worked through, this is quick reasoning
- Scriptlets remove the need for an agent to know it is in a script to function
- Scriptlets make manual knowledge acquisition practical as you don’t need to record “all knowledge”
- Scriptlets enable lifelong learning as agents can more easily augment a few critical slots in an `EVENT` than construct an entire script from scratch

Let’s use scriptlets to improve the example

- We want to use scriptlets to make a minimal dialog model
- We need to stick to keeping things simple (as simple as it can be)
- We want things to be domain independent, and to feel like common knowledge

Scriptlet Implementation

```
REQUEST-INFO
is-a                COMMUNICATION
agent               HUMAN
beneficiary         HUMAN
…
```



```
REQUEST-INFO
is-a                COMMUNICATION
agent               HUMAN-1
beneficiary         HUMAN-2
happens-next        RESPOND-TO-REQUEST-INFO
                       agent          HUMAN-2
                       beneficiary    HUMAN-1
```

To make a scriptlet:

- Turn some of the fillers into internal instances
- Add a `happens-next` field
- For this concept, the default value is a type of `RESPOND` (the listener is now the speaker)

NLU with Scriptlets – “Yes.”

```
ACCEPT-1
    agent                HUMAN-2
```

?

```
RESPOND-TO-REQUEST-INFO-1
    agent                HUMAN-2
    value                positive
    theme                INGEST-1
INGEST-1
    agent                HUMAN-2
    theme                COOKIE-1
```

”Did you eat a cookie?”

```
YES-ADV2
...
    THEME
        <the proposition
        of the previous text that
        anticipates a response>
```

- The second sense of YES is now selected. **Why?**
  - The agent knows it has an instance of `REQUEST-INFO` in its working memory.
  - `REQUEST-INFO` is usually followed by `RESPOND-TO-REQUEST-INFO`; and further, the proposed `agent` (`HUMAN-2`) lines up.
  - The agent can fill the the `theme` as well.
  - All of this allows the agent to prefer this sense when interpreting the meaning.

NLU with Scriptlets – “Yes.”

```
RESPOND-TO-REQUEST-INFO-1
    agent                HUMAN-2
    value                positive
    theme                INGEST-1
INGEST-1
    agent                HUMAN-2
    theme                COOKIE-1
```

“Yes.” == “Yes, I ate a cookie.”

The TMR is improved; **the scriptlet effectively expands the useless “Yes.” into “Yes, I ate a cookie.”**

Scriptlets can be used across various input modalities

- `RESPOND-TO-REQUEST-INFO` is a `COMMUNICATE`, not just a `SPEECH-ACT`. It doesn't matter what the source is, the agent can reason over it just the same
- Our Text Meaning Representations (TMRs) have a vision analogue (VMRs); and the lexicon's counterpart, the opticon, fills the same role
- We can have an opticon entry for *nodding a head* whose semantic interpretation is identical to the speech act for "Yes!"
- The scriptlet will get involved in exactly the same way, making it very portable, even across input modalities

Conclusion

- Demonstrated how scriptlets can be used to implement a dialog model
- Showed how scriptlets aren’t bound to a domain, are small enough to reasonably acquire, and are generally useful to agent reasoning
- Scriptlets are one part of the long game of knowledge engineering for human-like AI

Questions?

This research was supported in part by the U.S. Office of Naval Research.