Erick Stone
11/8/2019
HW09

Analysis of "Conceptual Analysis of Natural Language".

The following outlines the details of the paper:

It is an unsolved problem to map natural language to memory structures.
An approach to approximate this is Conceptual Analysis (CA).
CA seeks to take language and break it into syntactic elements.
The elements are then analyzed via pattern recognition to identify key elements and the assumption is that this process equates to understanding.

One of the key concepts they address is flexibility. ACs prior to this paper, handle information as they recieve it, word by word. Things such as sarcasm and jokes however require knowledge of both circumstance, previous words, and words yet to come.

The requirement to hold information requires flexible memory constructs. Note that inference might be ambiguous until sufficient information can be appended to these memory constructs. As well, groupings of memory constructs in order, proximity, relevance, and importance may all be important in language cognition. So, we can treat these memory blocks as expectations.

Expectations are held until they are met. We gather expectations and resolutions and decide when and how to remove them. One such process described in the paper is:
1. Get the next word or phrase (item).
2. Load the request of the item into a request list.
3. Consider all items in request list.
4. Loop to 1.

Requests are an embodiment of processed knowledge. They either test for the occurence of a word or phrase or test for expectation to be met in another list. These two test kinds are lexical and conceptual respectively. Requests are dictioned beneath words relevant to them. When a request is met, it has four possible actions:
1. Add to the concept list.
2. Fill a slot in a concept structure with another structure.
3. Activate other requests.
4. Deactivate other requests including itself.

The paper argues that syntax is actually inappropriate for defining and using syntactic knowledge. This is because the syntax alone does not mandate the entirety of the situation or domain knowledge required to understand the syntax.

The Recency Rule dictates that the order and proximity of objects and concepts added also matter in conceptual understanding. They use noun groups to identify this as a requirement. A noun after all can be a descriptor if followed by another noun. So, obviously syntax does matter, it is just insufficient by itself.

Use CList to handle multiple simultaneous True requests to avoid ambiguity. It is possible that none of the requests should activate as the ambiguity can't yet be resolved requiring more complex decisions to be made.

And this is some of my thoughts after reading the paper:

I like this paper. It conceptually is a fun step into a new territory for me. I imagine at the time this was ground breaking for this field. I don't think it is enough though. The authors outlined steps seem to do well, but I'm not so sure they are adequate for real biological systems mimicking or true intelligence. Perhaps an automated approach could be taken to describe the process? We use neural networks to build neural networks, perhaps a neural network to build a CA system?

The more I learn, the more amazing our brain seems. I predict that if we do find a means to create true intelligence, it will/must have logic from all of the different machine learning branches. Convolutional networks to break up the memory stored in an RNN. CA generated networks to handle language processing and image processing. RNNs to handle storing and retrieving entire concepts. Weight transfers to move the concepts the CA uses between short term and long term storage, and classification trees to quickly access related concepts quickly. Perhaps even more?