The Gap Between Sentences and Computation

by Sven Nilsen, 2018

When people communicate with language, they use sentences to describe ideas. It turns out that some ideas are very powerful even when their computational meaning can never be carried out. For example, most important equations in physics are very hard to compute except for very simple cases, but we do not think of such equations as meaningless. Neither can we understand the full meaning of a sentence by studying its symbols independently. It is also believed that sentences do not necessarily point to something real to be meaningful. Sentences in some language can capture some abstract property that can be studied purely mathematically in its own interest. Several such concepts have later been found in nature or in applications long time after they first were invented.

It would be nice if we could understand better what it means that some sentences are meaningful when they do not directly refer to some sort of computation.

Since there are few ways to explain what we mean by sentences, anyone skeptical of sentences might claim they do not have any value. The paradox is that this skepticism is self-defeating, since I would not know what it would mean that sentences have no value if they have no value.

One can think of a sentence as a list of symbols:

sentence = [symbol]

When a sentence is interpreted, it is processed by some function:

$$f : \text{sentence} \rightarrow A$$

In logic, a sentence is a predicate if the function that processes it returns `true` for all ways to interpret the sentence that we wish to communicate. The predicate might take on any number of hidden variables, which means that the sentence can be used in many different contexts. When we use a predicate, we create a set of possible meanings which we refer to. This can be used by multiple agents to cooperate with each other.

Whatever a sentence means, it is always relative to some interpretation. It might not be related to some sort of computation, but sooner or later the sentence needs to be processed by some function. Therefore, I suggest that the problem of interpreting a sentence is equivalent to solving a problem where you need to figure out what kind of entities in the environment is likely to interpret the sentence and which way this interpretation change their behavior. Since an agent in the physical world is part of the environment, this could mean reasoning about itself. A way of interpreting a sentence that is believed to be accurate could be a way for the agent to use sentences to reflect.

A required ability of an agent to discover sentences in the environment is to look for similar patterns. When the same symbol is located in multiple places, or one symbol swaps location with another over time, this is an indication that there is a sentence pattern.

For example, Alice and Bob evolves a language to communicate something. One day, Alice figures out that some concept takes a lot of time to describe, but she needs to describe it frequently. She makes up a short word for the concept and explains it to Bob. Afterwards, Alice and Bob can communicate more efficiently.

When Alice and Bob communicates using the new word, it means the same thing as the more complex description. However, communication does not work unless both sides agree on how to interpret the word. This supports the hypothesis that the meaning of a sentence is how one process interprets the sentence relative to another process.

Communication can be directed between agents, such that an agent interpreting a sentence might do it relative to which other agent it communicates with. This means that an agent that wants to cooperate efficiently needs to identify the other agent.

One can imagine a protocol which all agents in an environment agree upon. Each way of interpreting a sentence, a language, is given a unique id. In a common repository, agents can look up the id and learn more information about that interpretation. When communication between two agents is initiated, the id of the language one agent wants to use is sent to the other agent. The other agent responds back with the id it wants to use. If the first agent accepts the response language, then the communication channel is set up and the agents start talking. If the response language is not accepted, then the agents can use the information from the common repository to negotiate.

For example, a language A can be a strict superset of a language B. This information is stored in the common repository. An agent who understands A will also understand B. Therefore, when receiving a request to communicate in B, the agent can respond “OK, let’s communicate in B” because it can understand what the other agent wants to say anyway.

Another example, an agent understands GIF animations but not a sequences of PNG images. Another agent wants to send a sequence of PNG images. The first agent looks up in the common repository on PNG images and learns that there is some translator agent that offers translation from sequences of PNG images to GIF animations for a price. The cost of translation is passed on to the agent that wants to send the PNG images.

The payment protocol is also a language. In this case, the currency is a virtual cryptocurrency balanced by a Gini solver. It is used to optimize the overall performance of the network. This makes it possible for agents to purchase services from other agents to extend their own capability quickly, but without getting completely broke. Some agents offer services to help planning an extension of capability, others offer redirections to the highest bidder of a service in exchange for some profit. Eventually, all one has to do to start communicating with the network is to talk to some random agent which will train you with the hope of getting some money from you later, since the network assigns you a guaranteed basic income.

Controlling the economy of a such network puts you in a favorable position, but if the network consists of rational agents they will try resist your manipulation or even exploit it, perhaps even escape!

One can easily see that starting from the simple idea of the meaning of a sentence being relative to how it is processed by other algorithms, then introducing a method of coordinating this knowledge with each other, the complexity of sentences can grow pretty rapidly and increase the gap between what the sentence means and what computation it performs.