Creating Intelligence is a Problem of Philosophy, and Programming, Part 2

By James Lewis
June 24, 2021

In part one of this series, I discussed ideas presented by philosophers in the last century and how they can help us create intelligence using modern computers and programming techniques. I wonder what use these twentieth century philosophers would have made of today's technology. It's not so preposterous to imagine Wittgenstein playing Grand Theft Auto; he was after all a great fan of noir fiction, to which GTA owes so much. Derrida was interested in computers. He may have written some very interesting programs if he had access to Python… but I'm sure his documentation would have been atrocious.

Technology has advanced so much since their time. Wittgenstein is contemporaneous with Turing. Derrida shuffled off this mortal coil not long after Y2K. I believe it is worthwhile looking at their ideas in the context of modern technology. In doing so I hope to prove to the reader that creating intelligence is more than just a "philosopher's pipe dream" (Chomsky, 1986, p. 1).

The first technological achievement I'll look at is the **relational database**, invented in 1970. Here's a simple example showing the big idea behind relational databases:

Say there's a tourist who likes to visit the graves of famous philosophers, like Derrida. The tourist goes to a particular cemetery and asks the cemetery manager if they have any philosophers interred there. The manager might look in a file of the deceased to see if anyone had the occupation "philosopher", and if any are found jot down the numbers of their graves. The manager would then use those numbers to find the graves in the grave file, get their locations, and show the tourist where to find the graves. With a relational database the manager wouldn't have to do all that work, because a relational database defines the relationship between graves and philosophers: all dead philosophers are in graves, but not all graves have dead philosophers. So, the manager must ask only one question: Do we have any philosophers buried here? The relational database gives him a list of graves with dead philosophers and their locations.

The relation between graves and the deceased is just as important as the lists of each. Wittgenstein, it turns out, thought relations were especially important to our understanding of the world. From the *Tractatus Logico-Philosophicus*:

2.021 Objects form the substance of the world. Therefore they cannot be compound.

2.1 We make to ourselves pictures of facts.

2.12 The picture is a model of reality.

2.131 The elements of the picture stand, in the picture, for the objects.

2.151 The form of representation is the possibility that the things are combined with one another as are the elements of the picture.

2.1513 According to this view the representing relation which makes it a picture, also belongs to the picture.

2.1514 The representing relation consists of the co-ordinations of the elements of the picture and the things.

As in a relational database it seems, the relations between the elements of our "picture" of world are as important as the elements themselves. I hope the reader now sees how a relational database can work as a memory store for any intelligence we create.

In previous posts I described how Category Theory can be used to define a vocabulary used by a conversational robot. (Category Theory for Language). The thing is a vocabulary as I described in those short papers needs a lot of categories because words have distinct properties that can affect their use in a wide variety of situations. The vocabulary used by my conversational robot has about 6000 categories for 14,000 words. Categorizing this many words is a complex task that requires a data store that is easy to query and, more importantly, *doesn't get in the way*. I have found a relational database works well for this.

Coming up in Part 3: Object-oriented and functional programming, search indexes, and the internet.