SEMANTIC STORAGE II

HOW TO FIND ANALOGUES

Mykola Rabchevskiy
Apr 16 6

In the last chapter, we talked about how Semantic Storage is arranged and how information can be placed there. But for storage to be used helpfully, you need to be able to retrieve the information placed there; otherwise, it will not be storage rather than a small black hole.

All Semantic Storage operations for retrieving information can be grouped according to difficulty levels. The most elementary function is to check whether a specific concept is a sign of a given entity. For example, "***is the ant a mammal?***" In terms of a graph, this means checking if the graph has an edge going from the "***ant***" vertex to the "***mammal***" vertex. The answer will be "***true***" or "***false***," respectively.

Functions of the next level of complexity are obtaining a set of signs of an arbitrary entity (entity's ***syndrome***) and getting a set of entities for which a given concept is a sign (concept's ***explication***). For example, we can get a set of signs of the concept "***ant***," which may look like "***insect arthropod hexapod terrestrial collective animal***." The list of all mammals represented in Semantic Storage is an example of the second function.

The same complexity level can be attributed to a function that constructs a statement that defines a given entity in the knowledge manipulation language

“taboo” syndrome if one is specified). The presence of the mentioned asymmetry makes it worthwhile to be able to interrupt the search for a set of entities by syndrome if the number of those found exceeds a reasonable limit.

This complexity level also includes the search for entities similar to a given entity in the sense that the requested entities must have all the signs of the sample entity (and, possibly, some other signs). For example, if an ant is chosen as a sample (it syndrome mentioned earlier), we can expect termites and bees to be found among them.

A variation of this search is the search for an anonymous entity based on a set of signs. Due to the dominant subset of anonymous entities, this feature requires careful attention to implementation efficiency. The search for all entities for a given set of signs naturally requires a complete graph examination; the requirement of the desired entity's uniqueness allows the quest to be interrupted as soon as a second entity with the must-have syndrome is found.

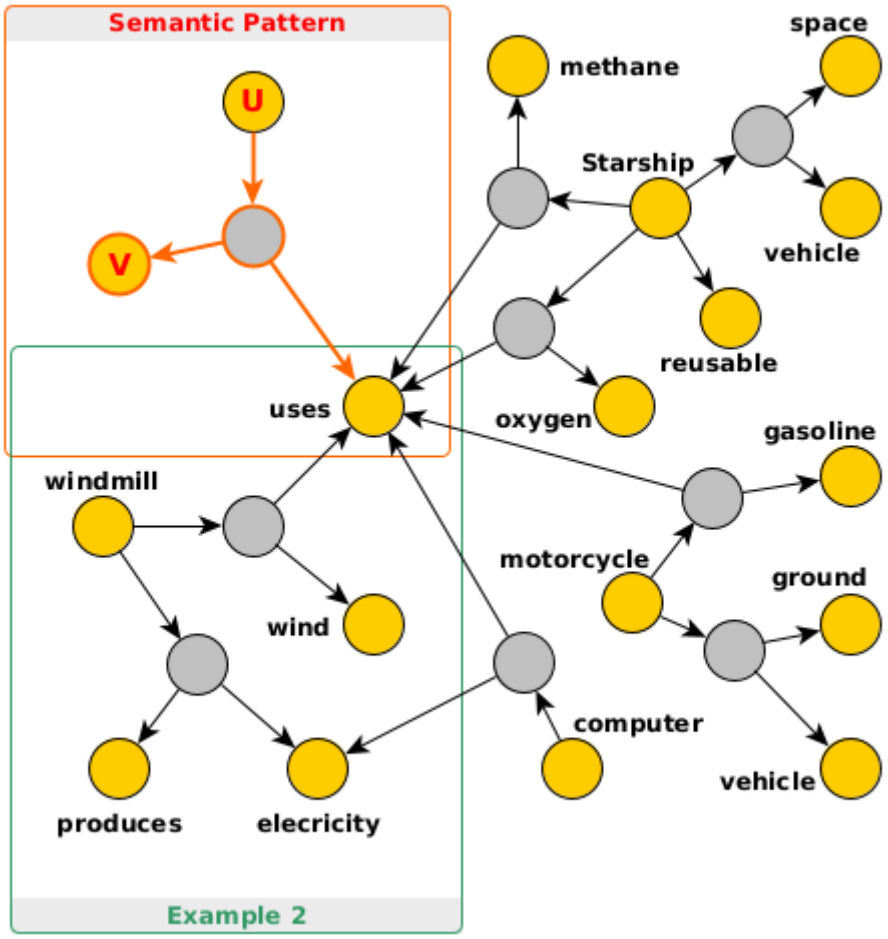
If a search is needed for several different syndromes, the selection is made in one pass.

Finally, the most challenging way to extract information is to ***search by the semantic pattern***. A semantic pattern is a group of entities connected by relationships within a group (we will call them ***inner relationships***) and graph entities outside this group (***outer relationships***). The search task is to find all such groups of entities with similar relationships, both inner and outer.

In terms of graph theory, this means the isomorphism of the pattern subgraph and the looked-up patterns under the matching outer connections' additional condition.

In terms of semantics, this means a ***search for analogies***, when part of the relationships (outer) is fixed, and the rest (inner) correspond to the sample's relationships.

In each found group, every element of the original pattern corresponds to an analogical element. Let us explain this with a specific example:



The graph is described by the following set of statements (anonymous entities marked in gray):
***Starship: reusable** (**space vehicle**) (**uses methane**) (**uses oxygen**).*
***motorcycle:** (**ground vehicle**) (**uses gasoline**).*
***windmill:** (**produces electricity**) (**uses wind**).*
***computer:** (**uses electricity**).*

Highlighted Semantic Pattern:

{ ***U V*** | ***U*** : (***uses V***) }

where ***U*** and ***V*** - pattern variables, "***uses***" - specific entity.

Since the combination "(***uses V***)" is an anonymous entity, the pattern graph consists of three vertices, one of which is an implicit (anonymous) variable; denoting it as ***#1***, we get the following set of pattern edges:

U --> ***#1***
#1 --> ***V***
#1 --> ***uses***

The search result, obviously, will be the following combinations:

U V #1

Starship oxygen (uses oxygen)
Starship methane (uses methane)
motorcycle gasoline (uses gasoline)
computer electricity (uses electricity)
windmill wind (uses wind)

Omitting implicit entities, we get the desired set of pattern analogs:

U V

Starship oxygen
Starship methane
motorcycle gasoline
computer electricity
windmill wind

More complex pattern:

{ ***U V W*** | ***U***: (***uses V***) (***produces W***) }

A set of pattern edges:

U --> ***#1***
U --> ***#2***
#1 --> ***uses***
#1 --> ***V***
#2 --> ***produces***
#2 --> ***W***

As a result of the search, we get a single subgraph marked in the figure as "*Example 2*":

U V W #1 #2
windmill wind electricity (uses wind) (produces electricity)

Semantic pattern search is implemented in three steps: adding a subgraph pattern to the semantic graph, actual searching, and deleting a pattern subgraph. Alternatively, a group of nodes already presented in the graph can play the role of a pattern.

The names of the pattern variables that appear in the textual definition of the pattern are technically not labels of the graph vertices rather ***attributes***; this prevents name collisions (the uniqueness of the variable names within the pattern is sufficient).

The search for analogies is an essential part of intellectual activity; therefore, the ability to search by the semantic pattern is a beneficial function of Semantic Storage. Since Semantic Storage data structures use manipulation of sets, and not just single entities, search by semantic pattern demonstrates acceptable efficiency for a small number of variables. The search time naturally grows rapidly with the increase in the size of the pattern. The increase in the number of external links, on the contrary, reduces the search time.

SUMMATION

- *Elementary operations are checking for the presence of an entity sign, obtaining a set of signs (entity's syndrome), and obtaining a collection of items having a given sign (concept's explication).*
- *Simple search operations include searching multiple entities by feature set and searching for the unique anonymous entity by the syndrome.*
- *The most challenging thing is to search for a set of groups of entities according to a semantic pattern.*
- *Semantic Pattern Search implements detection of analogies.*

← Previous Next →

Eugene Mayevski Aug 23

There's a game, aimed to train imagination, where a player should say, which attributes cannot be applied to an object. E.g. a crocodile - can it be "glassy"? When building relationships between entities, it makes sense to also define the probability of the

relationship, which can be "always", "never" and "maybe".

♥ Reply

> **5 replies by Mykola Rabchevskiy and others**

5 more comments…

Ready for more?

Subscribe