Knowledge base

Tom Rochette <tom.rochette@coreteks.org>

November 2, 2024 — 36c8eb68

0.1 Context

0.2 Learned in this study

0.3 Things to explore

- Every person is a knowledge base. They decide how they want to encode what they experience and how often they need to retrieve that experience (the more it is used, the shorter it should be, similar to how compression works)
  - All humans share the same brain architecture, which means that we most likely encode sensory inputs in the same way, but using different neurons (although their relative position within the brain seems to be constant)
- How should knowledge be structured in order for it to be easily queryable/retrievable when asking various questions?
- Like a "real" person, remembers by activating on all the provided words and searching for related memories (ala find feature of text search)
- How can we say that an (new) experience E is the same as an already experienced experience E?

1 Overview

2 Properties/Requirements

- Must be centralized or distributed with means of updating a node with the latest data

3 Concepts of interest

- Person
- Organization
- Place
  - Real
  - Virtual
- PhysicalItem
  - Product
- Event

4 Complexity layers

? There's some resemblance with relational databases and database normalization, is there an isomorphic relation between the complexity layers and those two?

4.1 L0: Concepts/Entities

The lowest level of “indivisible” knowledge. Anything that can be thought of mentally: people names, locations, sizes, colors, images, videos, audios, topics, etc.

Each concept is presented as a node in a graph. At this layer, the graph is simply a set of vertices without any edges, also known as a null graph.

In a textual format, we can think of the concepts as a list with one concept per line. Similarly named concepts may take multiple lines (e.g. mouse as an animal and as a computer device), although it is suggested to use terms as descriptive as possible (computer mouse vs mouse).

At this level we also accept higher-level constructs such as images (vector of pixels), sounds (vector of amplitudes/frequencies, and videos (vector of images and sounds)).

As the concepts may not be easily identified, once they are inserted into the system they are given a unique textual identifier through which we can refer to instead.

4.1.1 Example

Jack
John
Father
Son
School
Car
Sun
Summer
sound of waterfall
sound of water crashing against the beach
video of firework
image of a cat

4.1.2 Capabilities

- Answer queries such as: “Do I know about Jack/School/Car/Sun/Summer?”

4.1.3 Process

- Scan documents and recognize entities
- Build a list of said entities

4.2 L1: (Anonymous) Relations

At this layer we represent relations between concepts. These relations are however anonymous, in the sense that they contain no information other than the fact that two concepts are related.

Relations are represented as undirected edges between concept nodes.

In a textual format, we can think of the anonymous relations as a list of concepts, two per line, indicating that the first concept is related to the second (and vice versa). One issue with this format is that concepts are duplicated over many lines (as many times as they are part of a relation).

4.2.1 Example

Jack John
Jack Man
John Man
Jack Adult

John Teenager
Jack Father
John Son

We can see in this example that what can be considered relations (such as father/son) are considered as concepts and are associated to other concepts (Jack and John).

4.2.2 Capabilities

- Answer queries such as: "Is *Jack* related to *John*?" (Yes), "Is *Jack* related to (the concept) of *Man* or *Woman*?" (Yes, Man), "Is *John* a *Father*?" (No), "Is *Jack* related to something that is a *Teenager*?" (Yes), "What concepts are related to *Jack* and *Teenager* and *Son*?" (John)
- Query language using the previous examples (note that the order of the arguments does not matter)
  - (Jack, John)? (John, Jack)?
  - (Jack, Man) or (Jack, Woman)? (Man, Jack) or (Woman, Jack)?
  - (John, Father)? (Father, John)?
  - (John, (Teenager, ?))? (John, (?, Teenager))? (John, x) & (Teenager, x)?
  - (?, [Jack & Teenager & Son])? (?, Jack) & (?, Teenager) & (?, Son)?

4.2.3 Process

- Using the results from L1, attempt to build a relation graph between entities
  - This process may be akin to the TSP, where instead of finding the optimal path through cities, we're trying to find the optimal relation set between entities

4.3 L2: Types/(Named) Relations

At this layer relations now have a name that can identify them. For instance, John - Brother - Jack indicates that John and Jack are brothers.

In a textual format, the concepts and their relation is represented as a triplet. However, if you are familiar with databases, relations can be seen as a table with two columns, one for each concept, and for which the table name represents the relation between the two concepts.

4.3.1 Example

John Brother Jack
Jack Friend Jason

Brother John Jack
Friend Jack Jason

Brother John Jack
Brother Jack John
OlderBrother John Jack

Father John Jack
Son Jack John

4.3.2 Questions

- Should relations be undirected(reciprocal) at this level or directed(unidirectional)?
- Should this level already have the ability to create higher level relations (3..n)?
- Should this level specify domain(input)/range(output) constraints?

4.3.3 Capabilities

4.3.4 Process

- From the results obtained through L1, we refine each and every relation by giving them a name

4.3.5 Ontology construction

(Based on http://cgi.csc.liv.ac.uk/~frank/teaching/comp08/lecture12.pdf)

- Enumerate terms
- Define classes
- Organize the concepts
  - Add abstractions where needed
  - Identify relations
  - Identify definable things
- Self-standing things vs modifiers
  - Self-standing things can exist on their own (people, animals, houses)
  - Modifiers "modify" other things (wild/domestic, male/female)
- Arrange concepts/properties into hierarchies
- Identify the domain and range constraints for properties

5 See also

- Knowledge transfer

6 References

- https://en.wikipedia.org/wiki/Relational_model
- https://en.wikipedia.org/wiki/Binary_relation
- http://www.slideshare.net/jamietaylor/freebase-schema
- http://cgi.csc.liv.ac.uk/~frank/teaching/comp08/lecture12.pdf
- https://piggydb.net/
- https://github.com/zadam/trilium/wiki/Promoted-attributes