kogito: A Commonsense Knowledge Inference Toolkit

Mete Ismayilzada, Antoine Bosselut

What is kogito?

kogito is an open-source, **modular** and **extensible** python toolkit to generate **commonsense knowledge** inferences from text.

```python
from kogito.models.bart.comet import COMETBART
from kogito.inference import CommonsenseInference

# Load pre-trained knowledge model from HuggingFace
model = COMETBART.from_pretrained()

# Initialize inference module
csi = CommonsenseInference()

# Run inference
text = "Student gets a library card"
context = "library"
kgraph = csi.infer(text, model, context=context)

# Save output knowledge graph to JSON file
kgraph.to_jsonl("kgraph.jsonl")
```

What is commonsense knowledge?

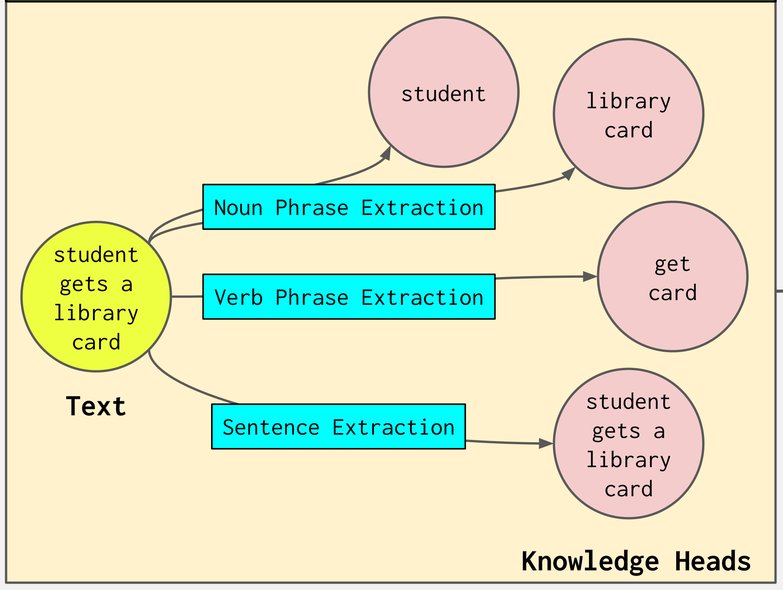
Commonsense knowledge consists of **implicit**, but **commonly known** facts about the everyday world such as “Lemons are sour”.

Where would I use it?

If you would like to integrate rich **commonsense knowledge** about the world **relevant** to your use-case, then kogito offers **all-in-one** solution.

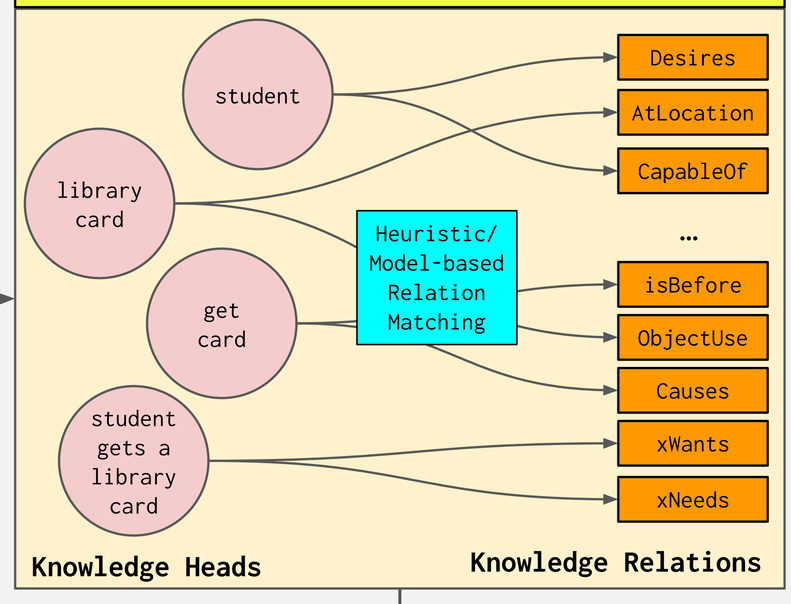
Head Extraction

First, potential relevant concepts called **knowledge heads** are extracted from the **text** using dependency parses produced from spaCy. Users can also define and plug in **arbitrary** head extraction methods.



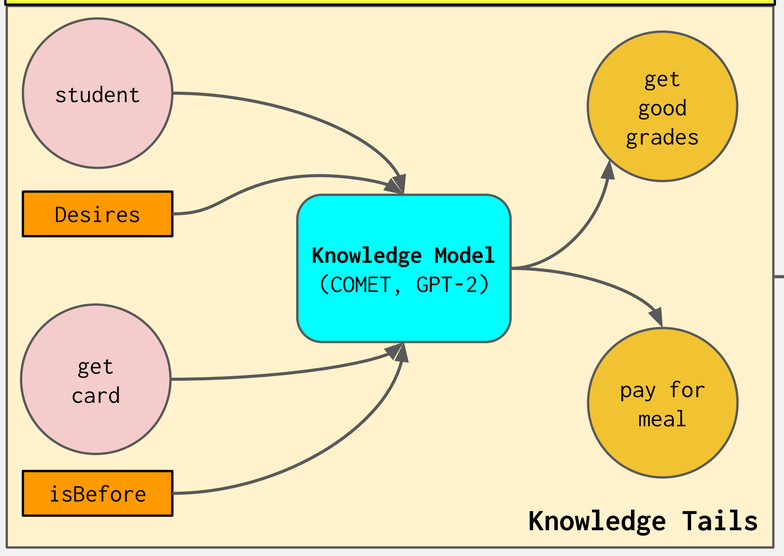
Relation Matching

Then, extracted knowledge heads are matched with relevant **knowledge relations** from **ATOMIC** and **CONCEPTNET** using **heuristic** or **model-based** matching algorithms. Users can also define **arbitrary** relation matching methods.



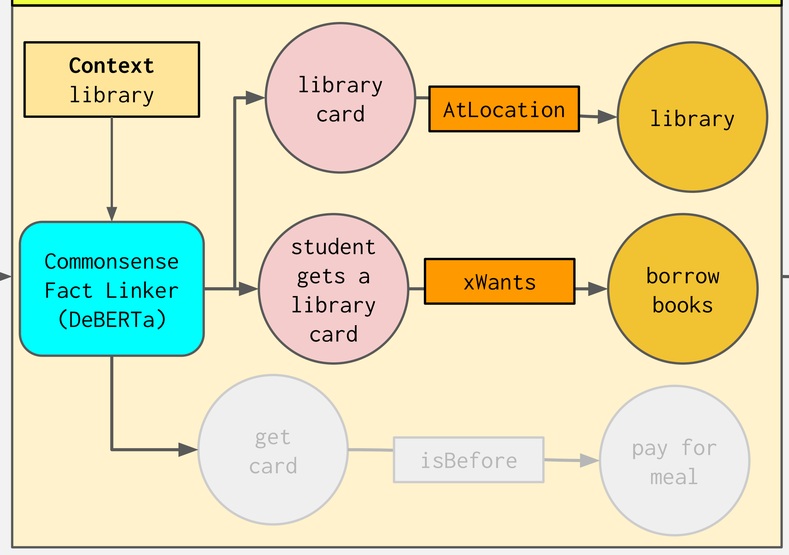
Knowledge Inference

Once we have **knowledge head** and **relation pairs**, we run them through **knowledge models** such as COMET to produce commonsense knowledge inferences also known as **knowledge tails**. Users can also define and plug in **custom** knowledge models.



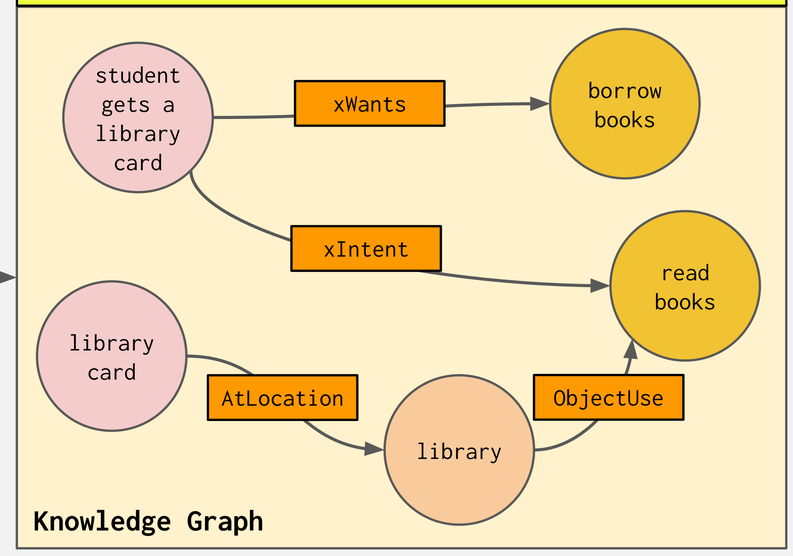
Inference Filtering

Optionally, in order to make the generated commonsense inferences (facts) more **relevant** to a provided **context**, we employ a commonsense fact linker model to **filter out irrelevant** facts. Users can define **custom** linker models as well.



Knowledge Graph

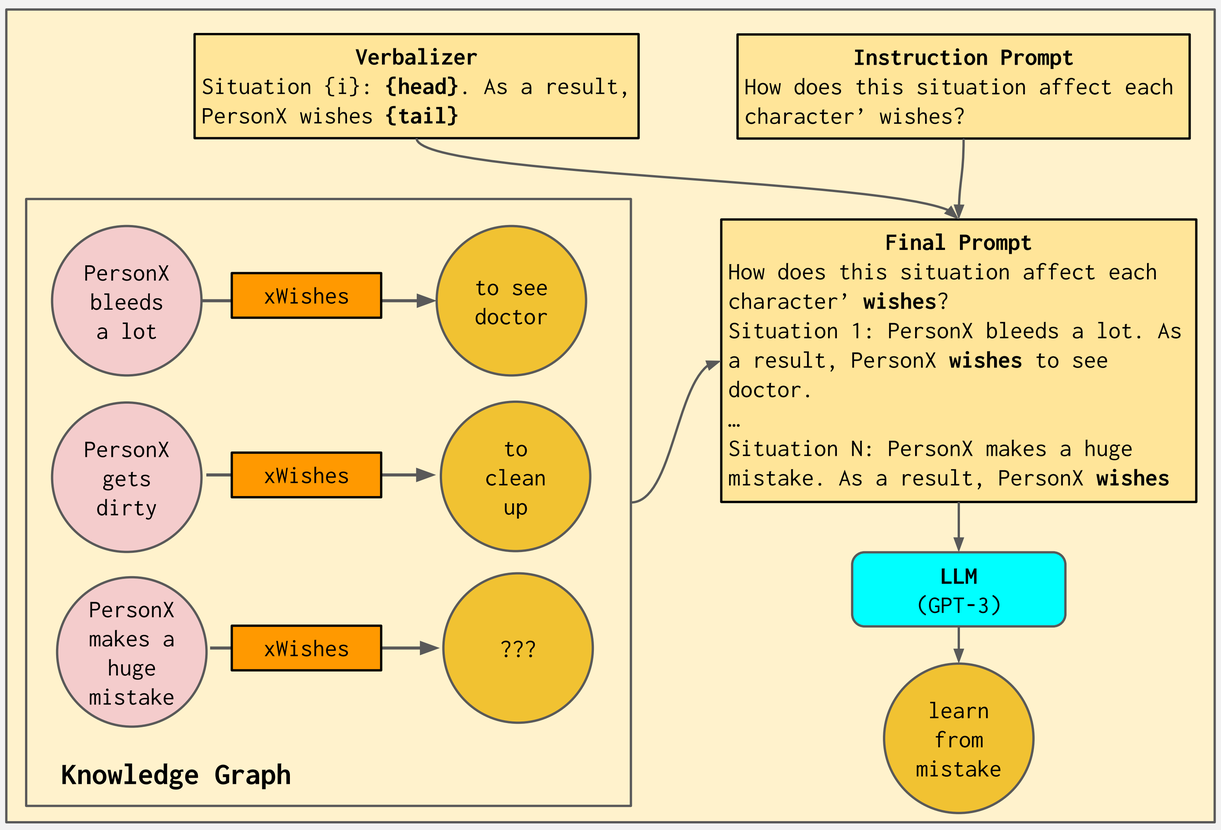
Finally, the resulting collection of knowledge **(head, relation, tail)** triplets also known as a **knowledge graph** is returned. This graph can be **saved** and later **loaded** for further processing. These abstractions allow for standardized I/O.



Custom Relations

In addition to the pre-defined relations, kogito also offers a way to define and use **custom new relations** via a technique called **symbolic knowledge distillation** (West et al.) from large language models such as GPT-3.

Figure on the right shows the technique conceptually for a new relation called **xWishes** (what does the PersonX wish?)



Paper

Code

Docs

Demo