Kogito

ergo sum

A Commonsense **K**n**o**wled**g**e
Inference **To**olkit

Mete Ismayilzada, Antoine Bosselut

Commonsense knowledge and inference

Commonsense knowledge
= implicit, commonly known, sensible world-knowledge

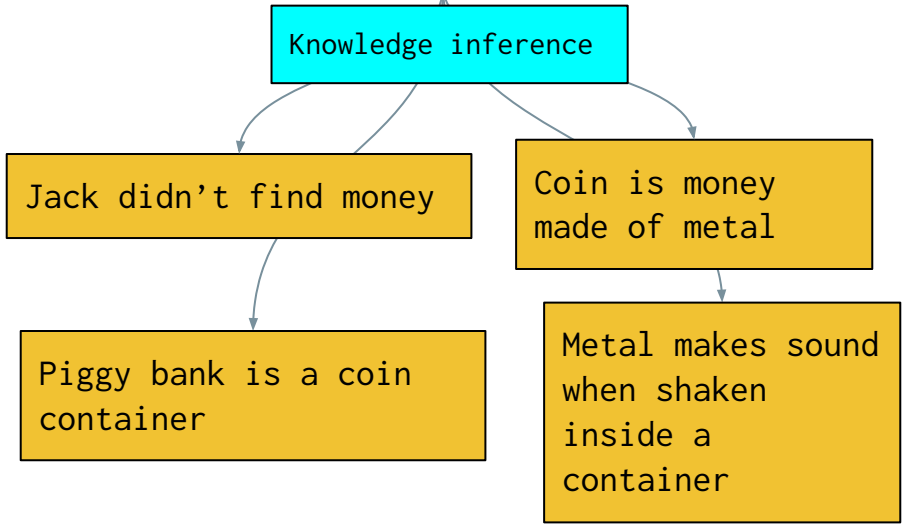
≠ Expert knowledge
"The partial derivative of the magnetic field with respect to time is the curl of the electric field"

≠ Encyclopedic knowledge
"Ljubljana is the capital of Slovenia"

≠ Common knowledge
"The common U.S. coins are pennies, nickels, dimes, and quarters"

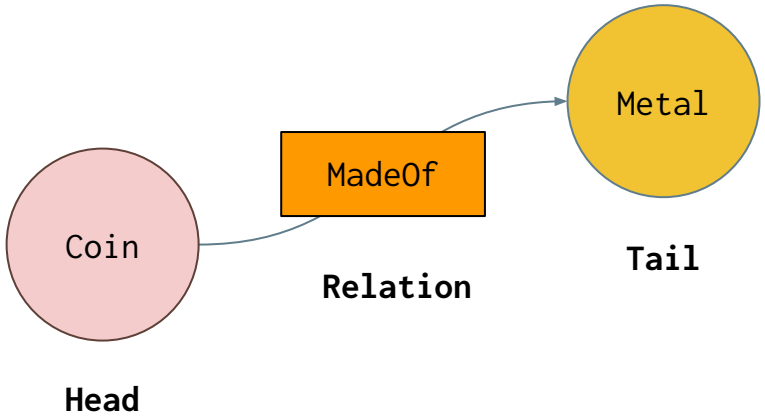
Jack needed some money, so he went and shook his piggy bank. He was disappointed when it made no sound.

Minsky (2000)



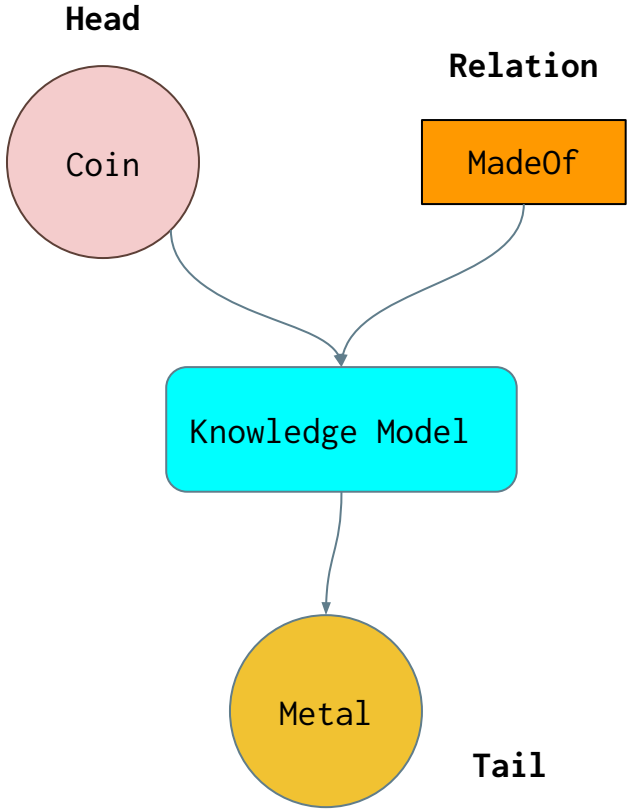
Knowledge representation

In NLP, (commonsense) knowledge is typically represented as a triplet **(head, relation, tail)** where **head** and **tail** refer to concepts (a.k.a nodes) and **relation** to a commonsense link (a.k.a edge) between them in a knowledge graph.



Knowledge model

Knowledge models are large-scale **language models** trained on knowledge graph tuples (triplets of ***head***, ***relation***, ***tail*** entity) and learn to express **knowledge** encoded in the parameters of the model when provided with a *head* entity and *relation* (e.g. COMET)



Motivation

Consequently, the success of language models as knowledge bases has inspired the field to deploy them in various downstream tasks:

- Generating figurative language
- Producing sarcastic responses
- Designing plots for stories and text based games
- Developing persona-grounded dialogue agents

However, each work re-implements the same pipeline for performing knowledge inference from text in different ways often resulting in duplicated efforts.

To this end, **kogito** offers an all-in-one **intuitive**, **modular** and **extensible** interface to facilitate access to knowledge models.

Quick look

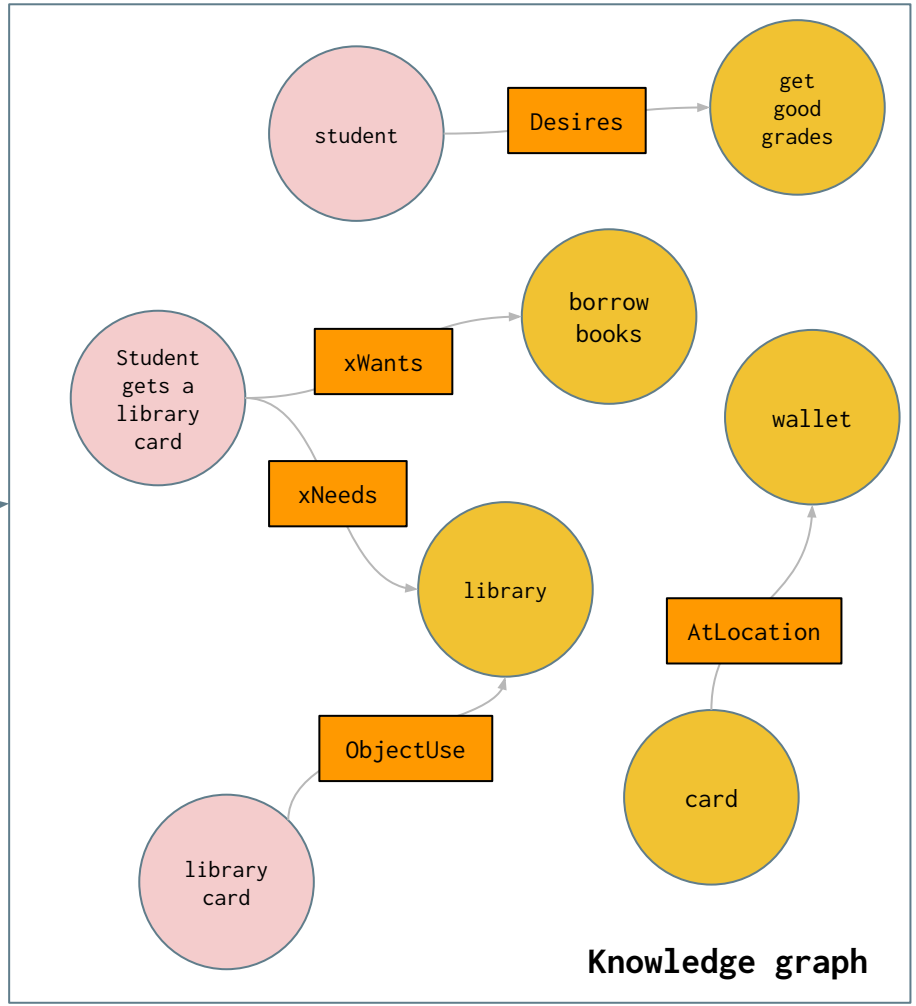
Text

Student gets a library card

Context

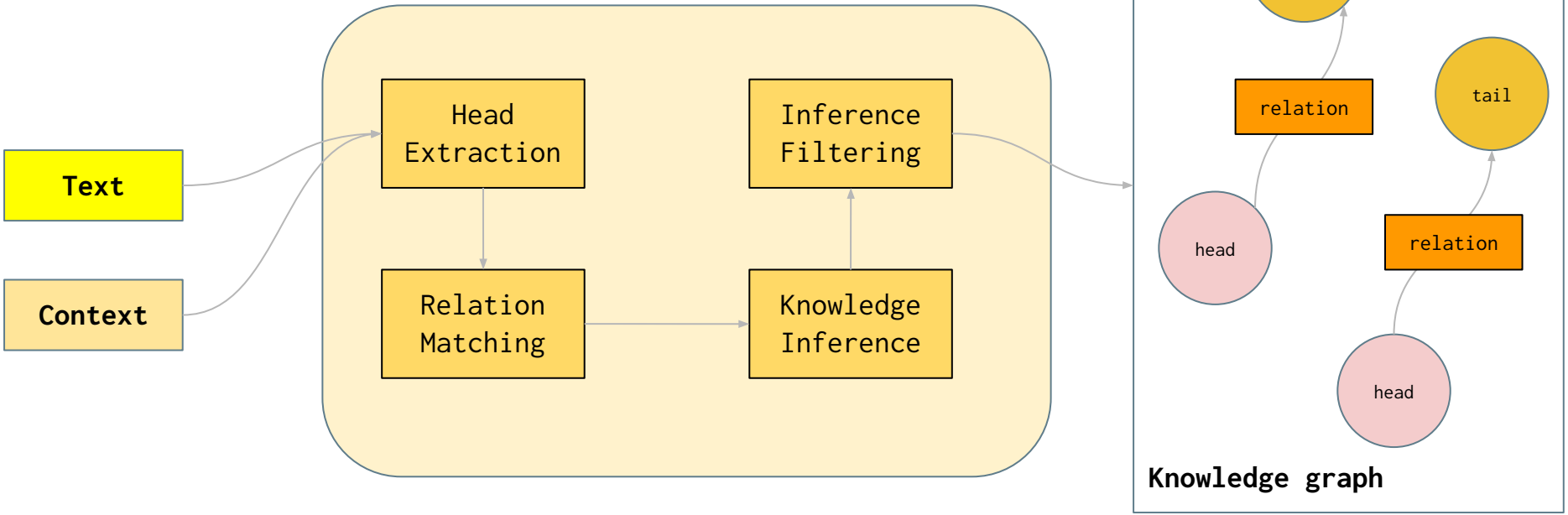
library

kogito



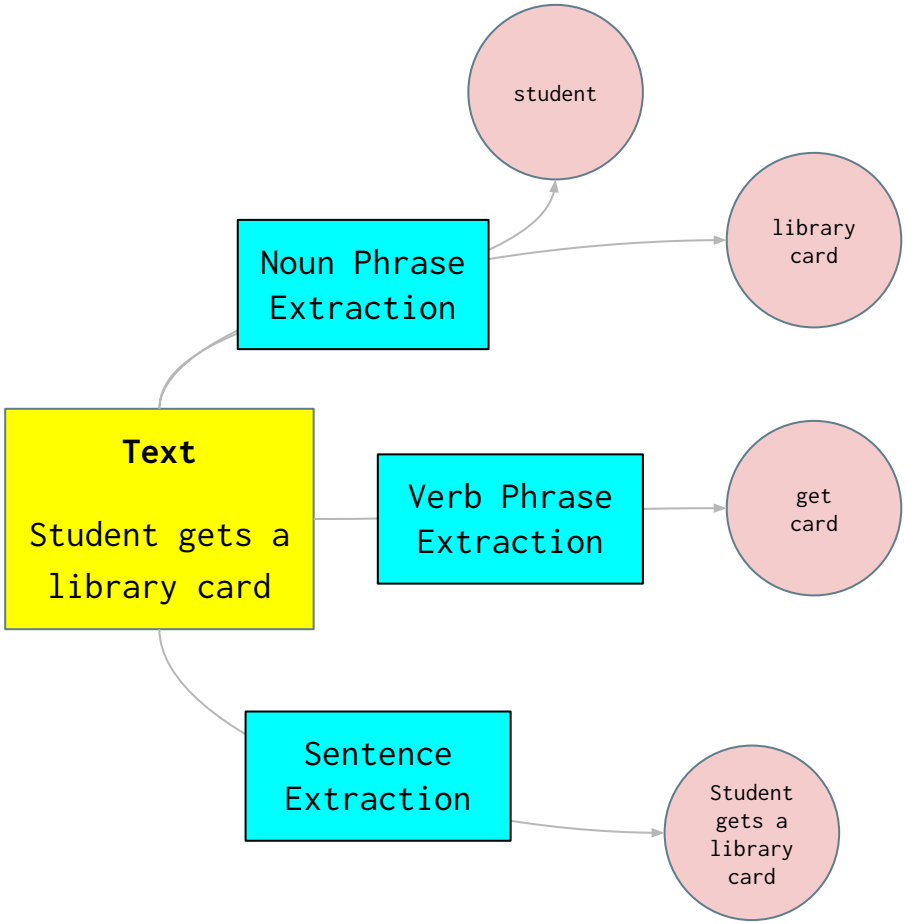
At a high level, kogito accepts a **text** and optionally a **context** and outputs a graph of commonsense inferences about this text **relevant** to the context.

Pipeline


Text
Context
Head Extraction
Relation Matching
Knowledge Inference
Inference Filtering
tail
relation
head
tail
relation
head
Knowledge graph

Head extraction

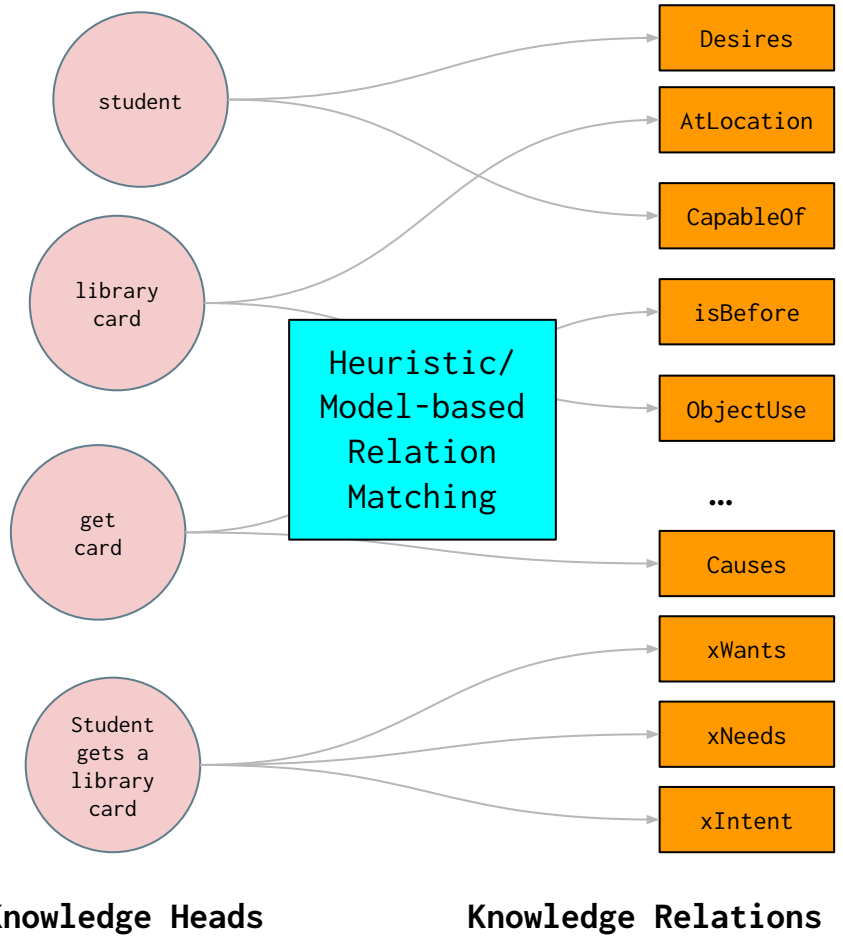
First, potential relevant concepts called **knowledge heads** are extracted from the text using dependency parses produced from spaCy. Users can also define and plug in **arbitrary** head extraction methods.



Knowledge Heads

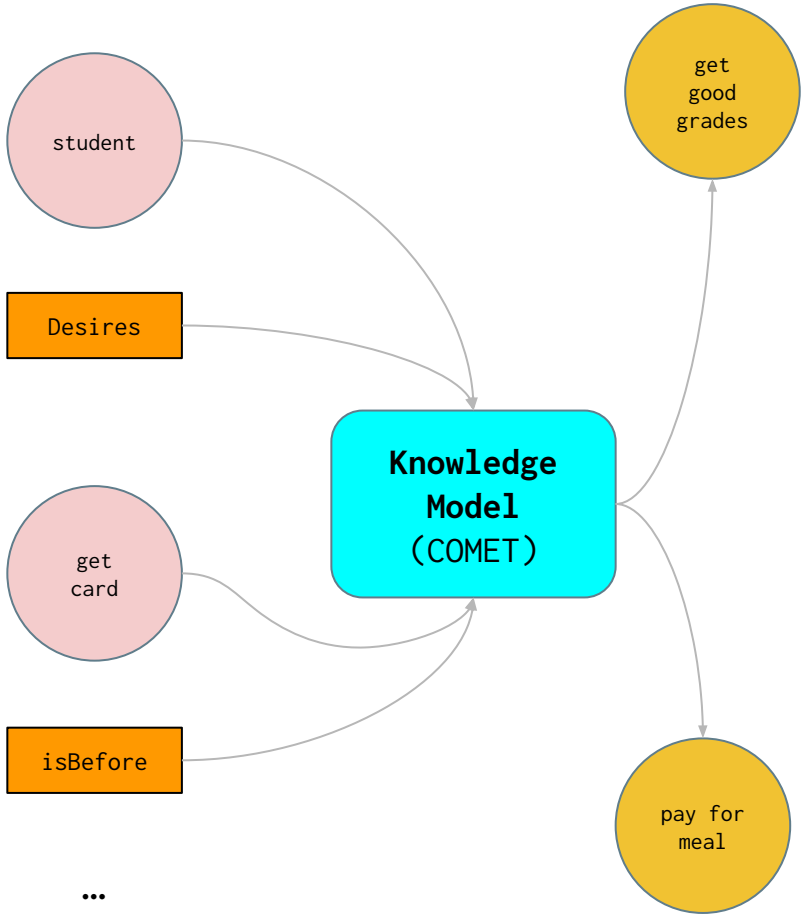
Relation matching

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



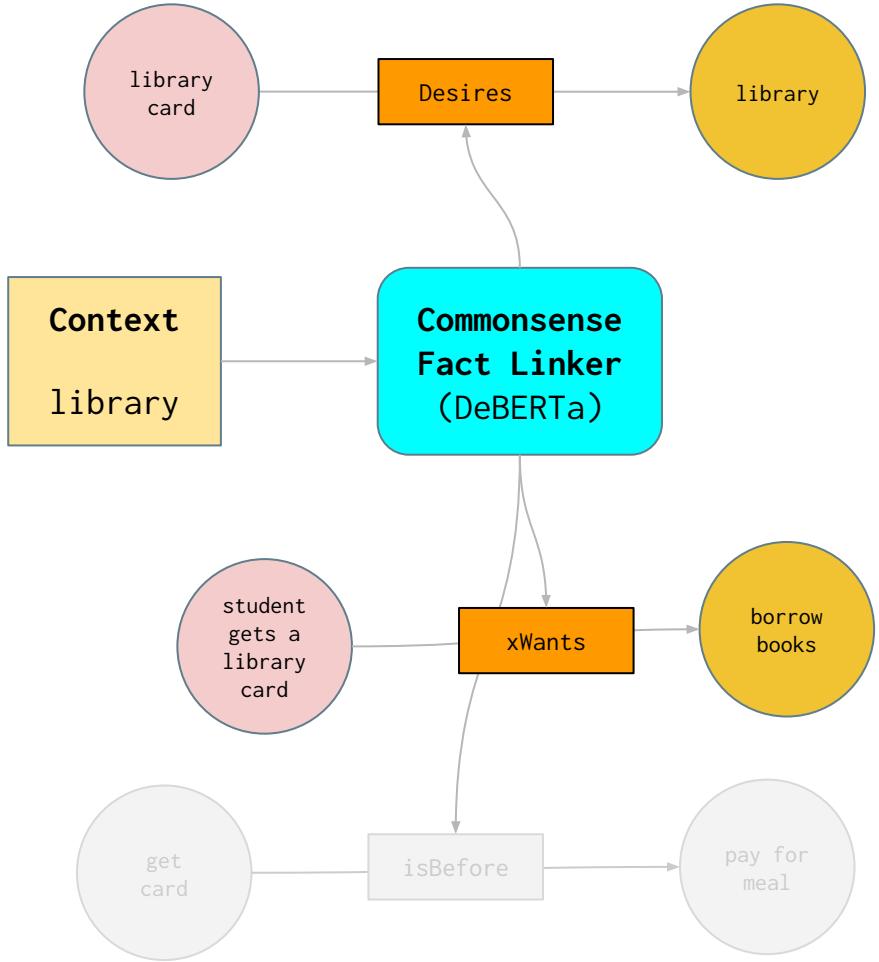
Knowledge inference

Once we have **knowledge heads** 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 and plug in **custom** linker models as well.

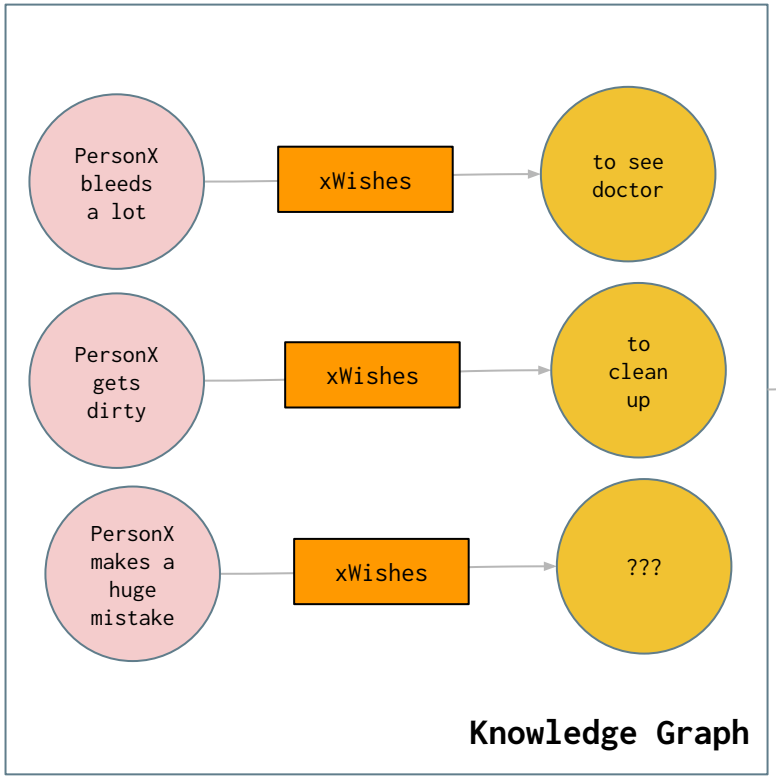


Custom relations

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.

In order to do this, user needs to provide a **sample knowledge graph** showing examples for the new relation which will be used for **few-shot prompting** the large language model. In the next slide, we show conceptually how this technique works for a new relation called **xWishes** (what does PersonX wish?)

Custom relations



Verbalizer

Situation {i}: {head}. As a result, PersonX wishes {tail}

Instruction Prompt

How does this situation affect each character' wishes?

Final Prompt

How does this situation affect each character' **wishes**?

Situation 1: PersonX bleeds a lot. As a result, PersonX **wishes** to see doctor.

…

Situation N: PersonX makes a huge mistake. As a result, PersonX **wishes**

LLM (GPT-3)

learn from mistake

Links

thank you

[paper]

https://arxiv.org/abs/2211.08451

[code]

https://github.com/epfl-nlp/kogito

[docs]

https://kogito.readthedocs.io

[demo]

https://kogito.live

[pypi]

https://pypi.org/project/kogito