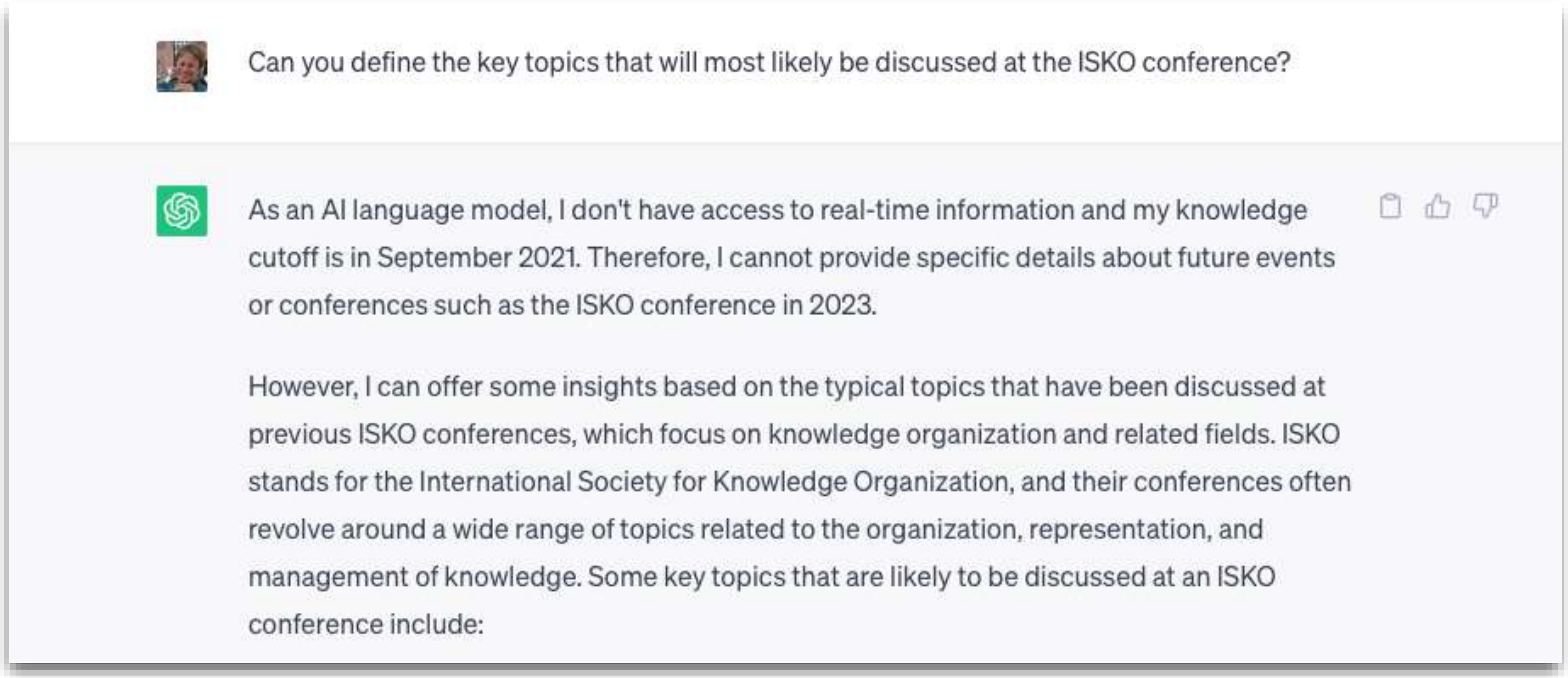
Knowledge Engineering in the Language Model Era

Prof. Paul Groth | @pgroth | pgroth.com |
indelab.org

Thanks:
Bradley Allen, Corey Harper, Elena Simperl, Ilaria Tiddi, Frank van Harmelen

July 24, 2023
7th BIENNIAL ISKO UK CONFERENCE

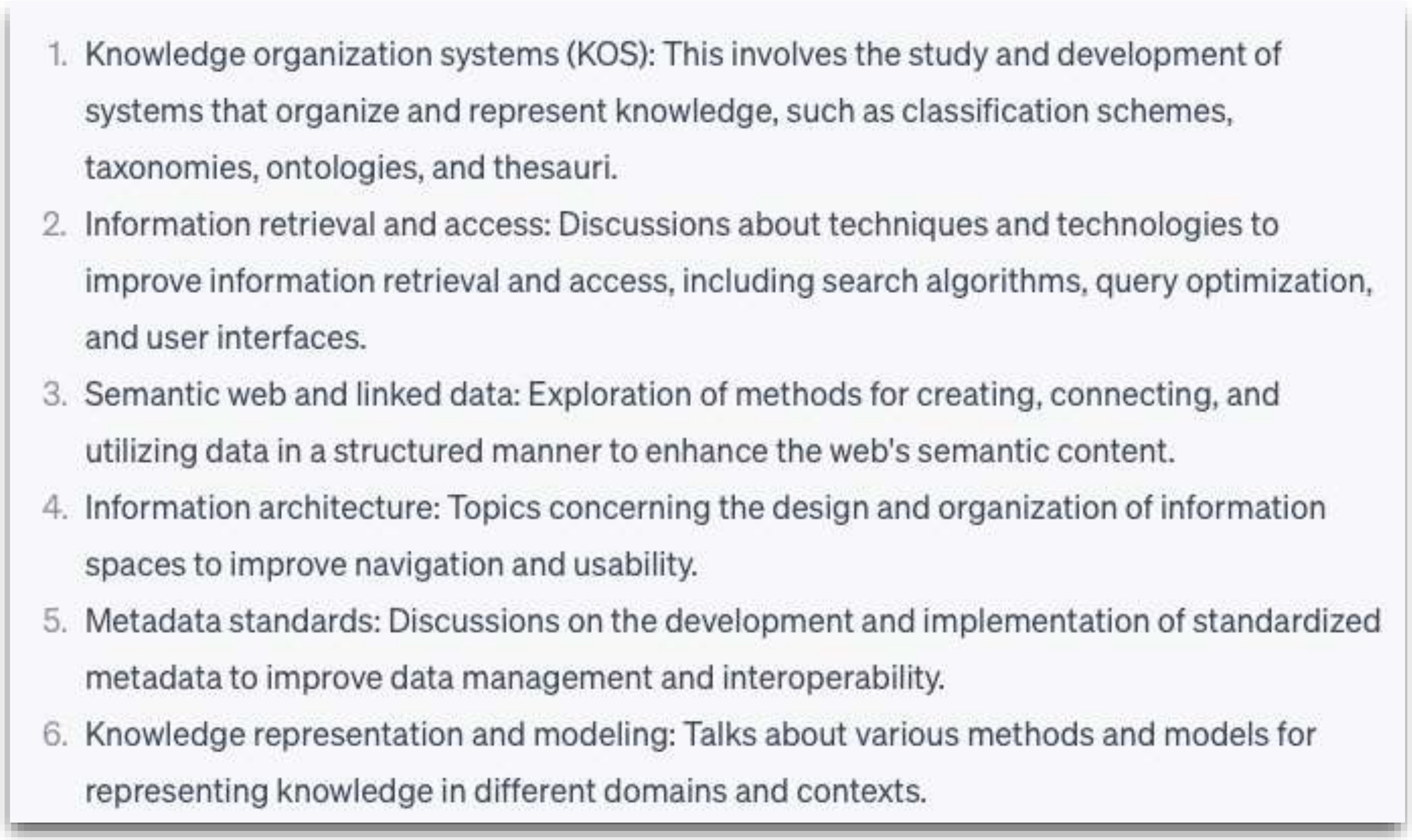
INDE lab
INtelligent Data Engineering
UNIVERSITY OF AMSTERDAM



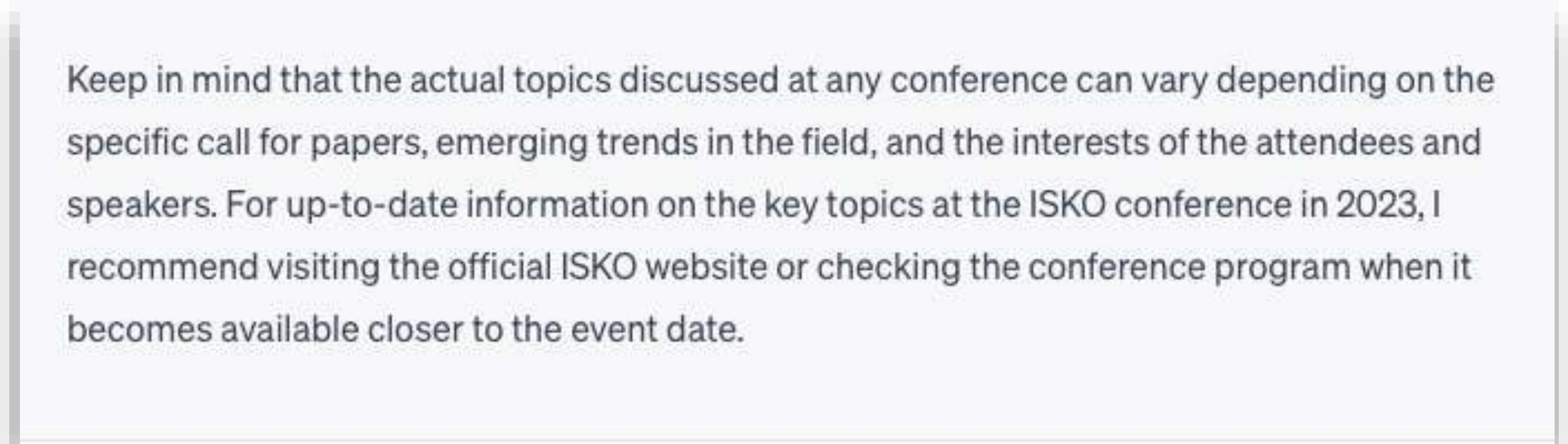
Can you define the key topics that will most likely be discussed at the ISKO conference?

As an AI language model, I don't have access to real-time information and my knowledge cutoff is in September 2021. Therefore, I cannot provide specific details about future events or conferences such as the ISKO conference in 2023.

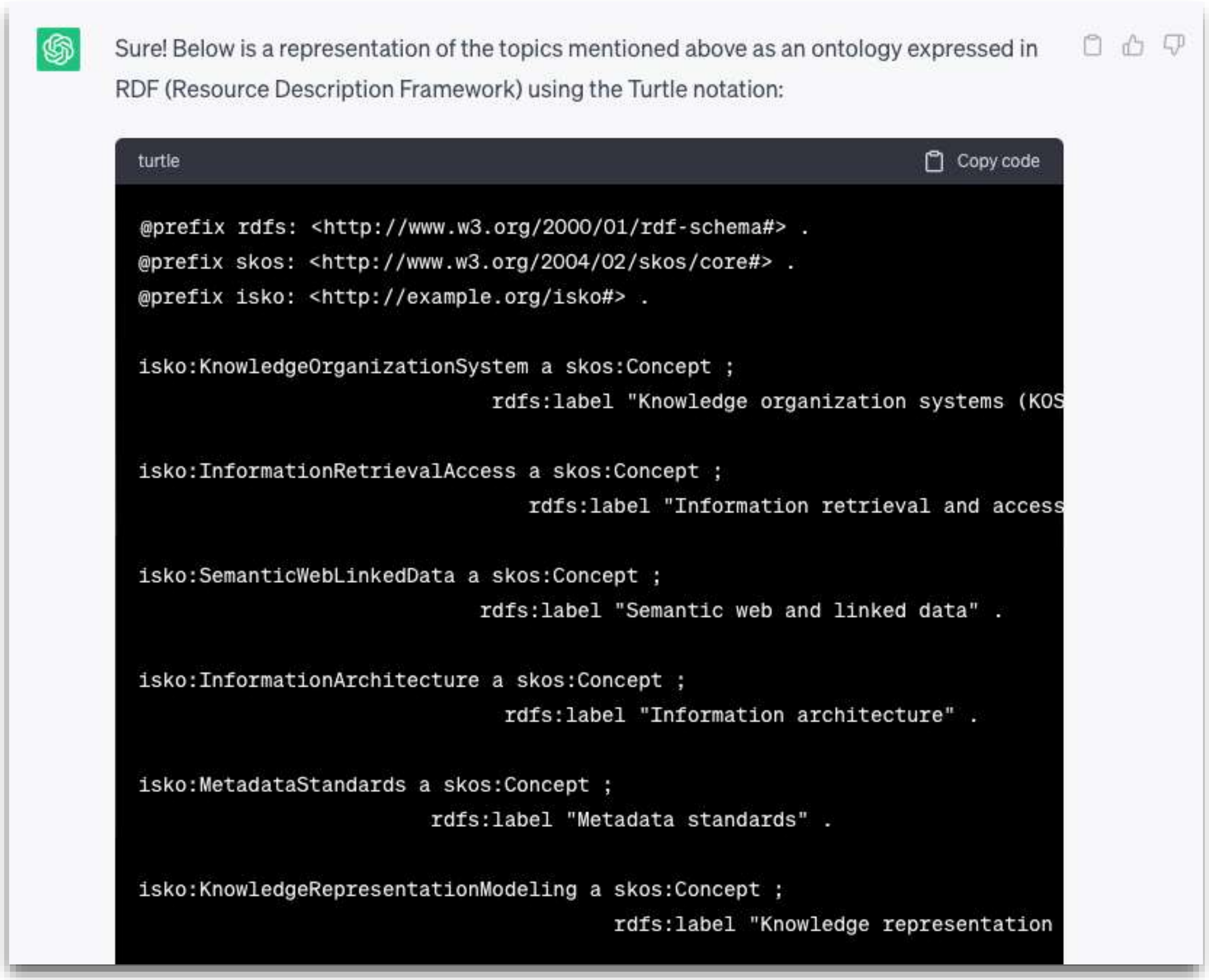
However, I can offer some insights based on the typical topics that have been discussed at previous ISKO conferences, which focus on knowledge organization and related fields. ISKO stands for the International Society for Knowledge Organization, and their conferences often revolve around a wide range of topics related to the organization, representation, and management of knowledge. Some key topics that are likely to be discussed at an ISKO conference include:



1. Knowledge organization systems (KOS): This involves the study and development of systems that organize and represent knowledge, such as classification schemes, taxonomies, ontologies, and thesauri.
2. Information retrieval and access: Discussions about techniques and technologies to improve information retrieval and access, including search algorithms, query optimization, and user interfaces.
3. Semantic web and linked data: Exploration of methods for creating, connecting, and utilizing data in a structured manner to enhance the web's semantic content.
4. Information architecture: Topics concerning the design and organization of information spaces to improve navigation and usability.
5. Metadata standards: Discussions on the development and implementation of standardized metadata to improve data management and interoperability.
6. Knowledge representation and modeling: Talks about various methods and models for representing knowledge in different domains and contexts.

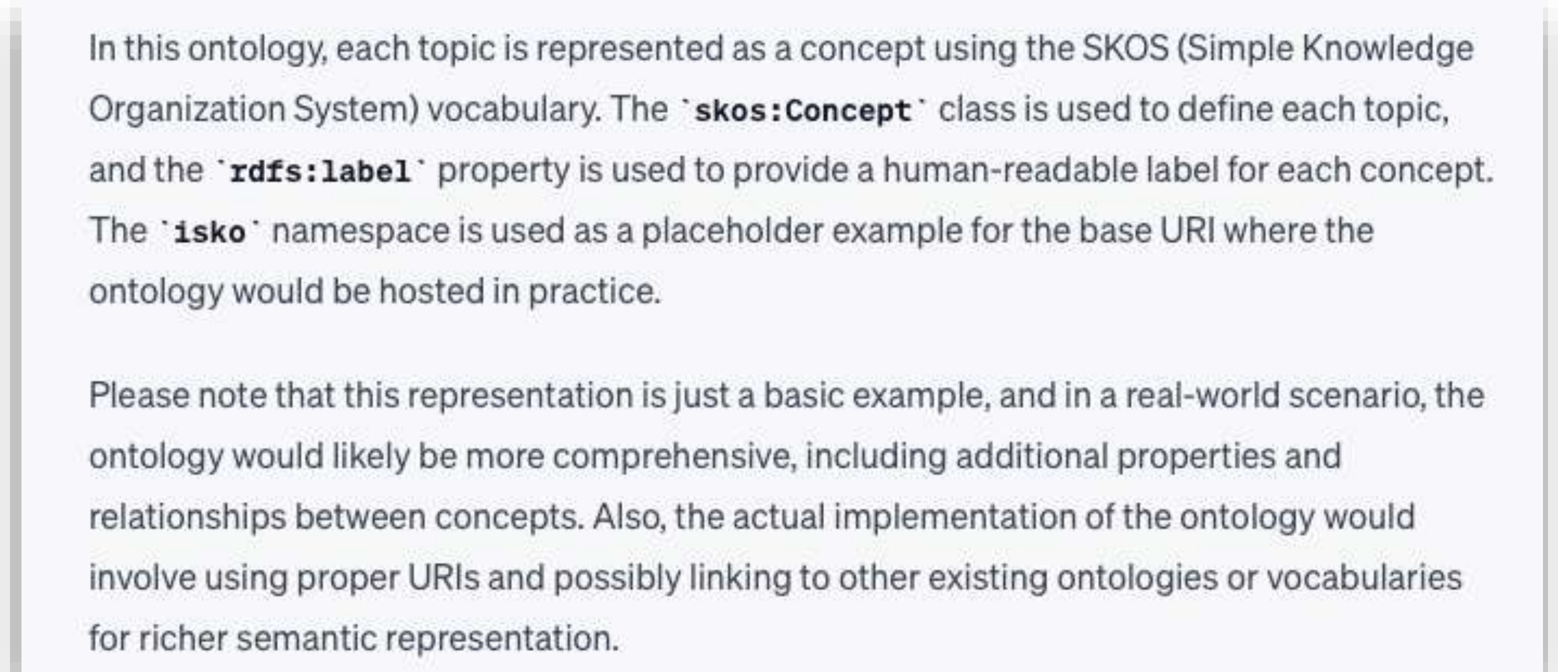


Keep in mind that the actual topics discussed at any conference can vary depending on the specific call for papers, emerging trends in the field, and the interests of the attendees and speakers. For up-to-date information on the key topics at the ISKO conference in 2023, I recommend visiting the official ISKO website or checking the conference program when it becomes available closer to the event date.



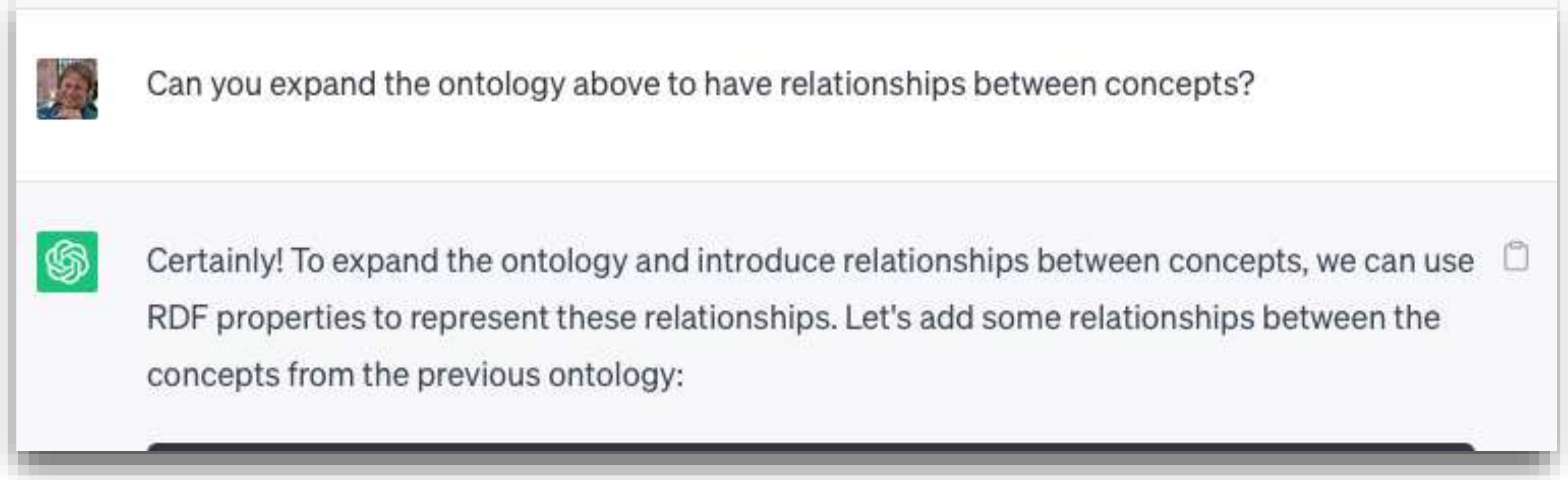
Sure! Below is a representation of the topics mentioned above as an ontology expressed in RDF (Resource Description Framework) using the Turtle notation:

```turtle
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix isko: <http://example.org/isko#> .

isko:KnowledgeOrganizationSystem a skos:Concept ;
                             rdfs:label "Knowledge organization systems (KOS

isko:InformationRetrievalAccess a skos:Concept ;
                                rdfs:label "Information retrieval and access

isko:SemanticWebLinkedData a skos:Concept ;
                            rdfs:label "Semantic web and linked data" .

isko:InformationArchitecture a skos:Concept ;
                              rdfs:label "Information architecture" .

isko:MetadataStandards a skos:Concept ;
                        rdfs:label "Metadata standards" .

isko:KnowledgeRepresentationModeling a skos:Concept ;
                                       rdfs:label "Knowledge representation
```



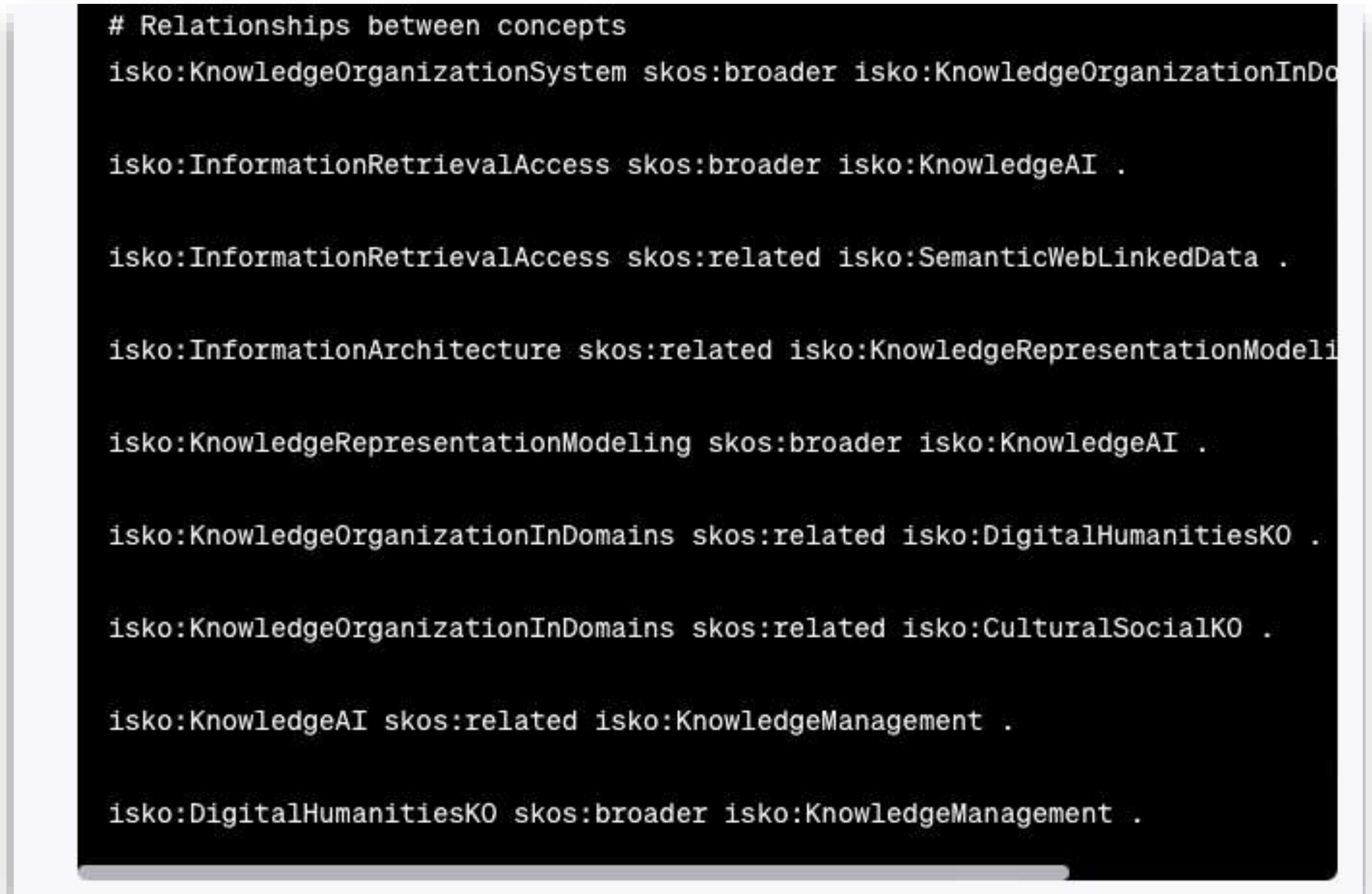
In this ontology, each topic is represented as a concept using the SKOS (Simple Knowledge Organization System) vocabulary. The `skos:Concept` class is used to define each topic, and the `rdfs:label` property is used to provide a human-readable label for each concept. The `isko` namespace is used as a placeholder example for the base URI where the ontology would be hosted in practice.

Please note that this representation is just a basic example, and in a real-world scenario, the ontology would likely be more comprehensive, including additional properties and relationships between concepts. Also, the actual implementation of the ontology would involve using proper URIs and possibly linking to other existing ontologies or vocabularies for richer semantic representation.

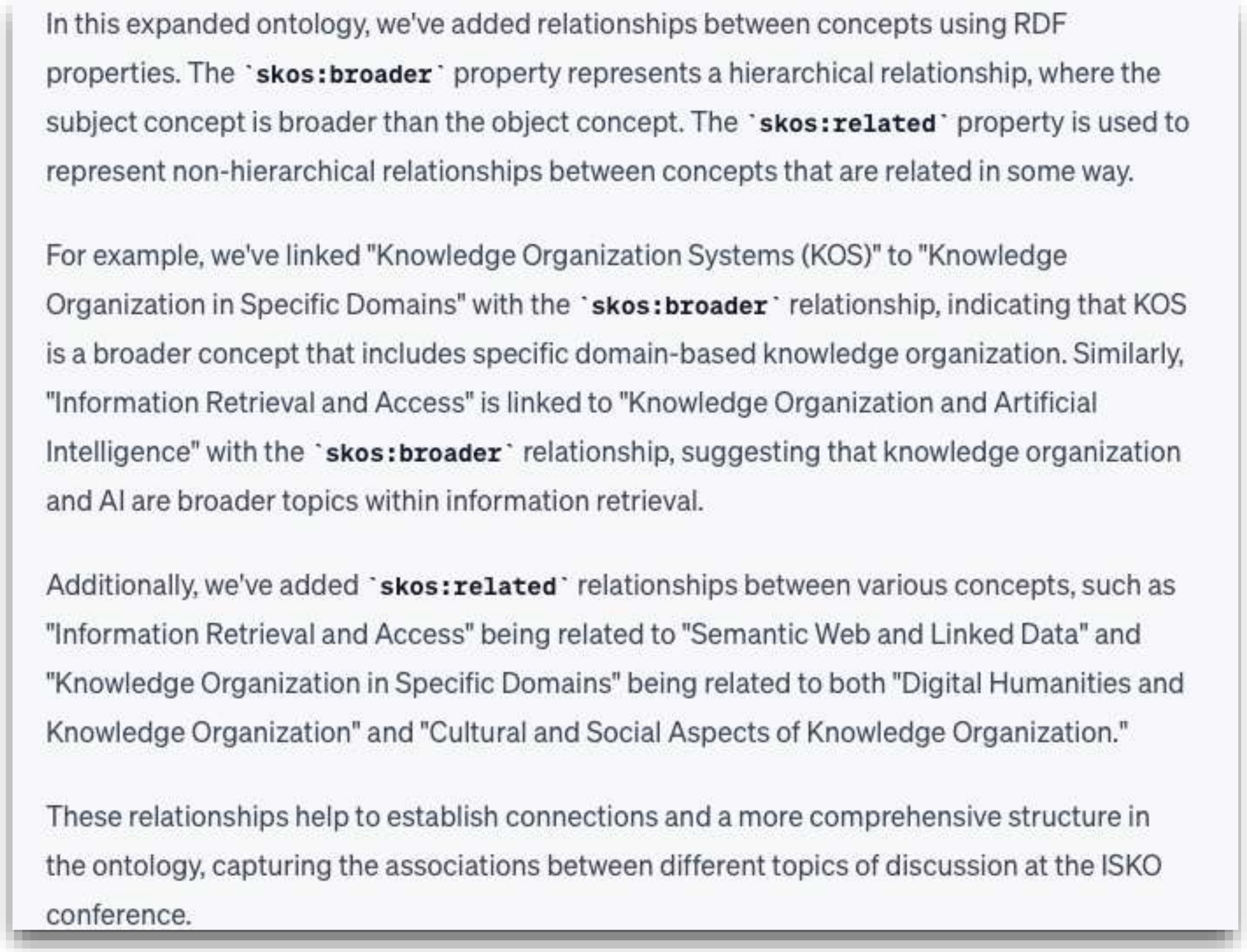


Can you expand the ontology above to have relationships between concepts?

Certainly! To expand the ontology and introduce relationships between concepts, we can use RDF properties to represent these relationships. Let's add some relationships between the concepts from the previous ontology:



```
# Relationships between concepts
isko:KnowledgeOrganizationSystem skos:broader isko:KnowledgeOrganizationInDo

isko:InformationRetrievalAccess skos:broader isko:KnowledgeAI .

isko:InformationRetrievalAccess skos:related isko:SemanticWebLinkedData .

isko:InformationArchitecture skos:related isko:KnowledgeRepresentationModeli

isko:KnowledgeRepresentationModeling skos:broader isko:KnowledgeAI .

isko:KnowledgeOrganizationInDomains skos:related isko:DigitalHumanitiesKO .

isko:KnowledgeOrganizationInDomains skos:related isko:CulturalSocialKO .

isko:KnowledgeAI skos:related isko:KnowledgeManagement .

isko:DigitalHumanitiesKO skos:broader isko:KnowledgeManagement .
```

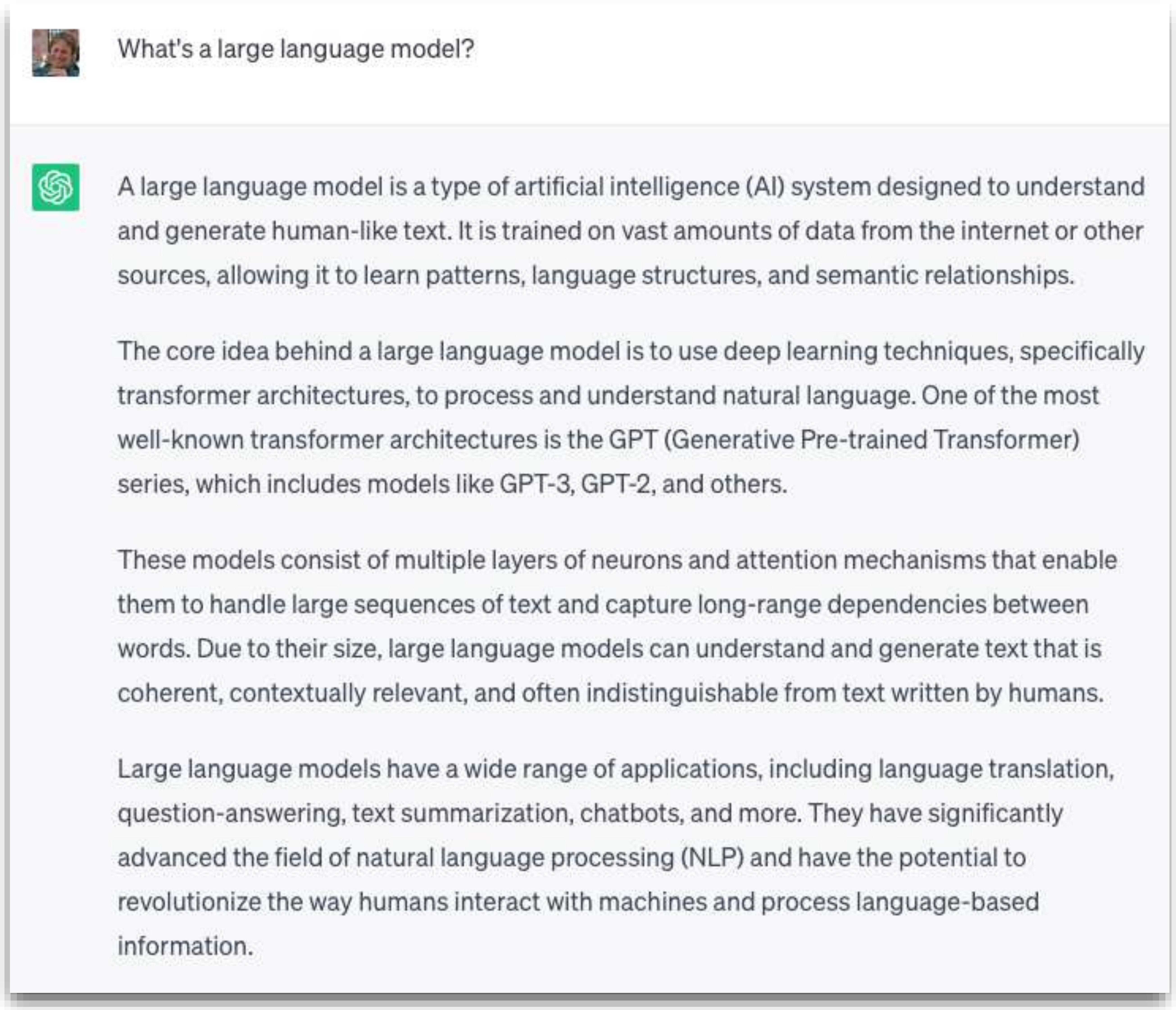


In this expanded ontology, we've added relationships between concepts using RDF properties. The `skos:broader` property represents a hierarchical relationship, where the subject concept is broader than the object concept. The `skos:related` property is used to represent non-hierarchical relationships between concepts that are related in some way.

For example, we've linked "Knowledge Organization Systems (KOS)" to "Knowledge Organization in Specific Domains" with the `skos:broader` relationship, indicating that KOS is a broader concept that includes specific domain-based knowledge organization. Similarly, "Information Retrieval and Access" is linked to "Knowledge Organization and Artificial Intelligence" with the `skos:broader` relationship, suggesting that knowledge organization and AI are broader topics within information retrieval.

Additionally, we've added `skos:related` relationships between various concepts, such as "Information Retrieval and Access" being related to "Semantic Web and Linked Data" and "Knowledge Organization in Specific Domains" being related to both "Digital Humanities and Knowledge Organization" and "Cultural and Social Aspects of Knowledge Organization."

These relationships help to establish connections and a more comprehensive structure in the ontology, capturing the associations between different topics of discussion at the ISKO conference.



What's a large language model?

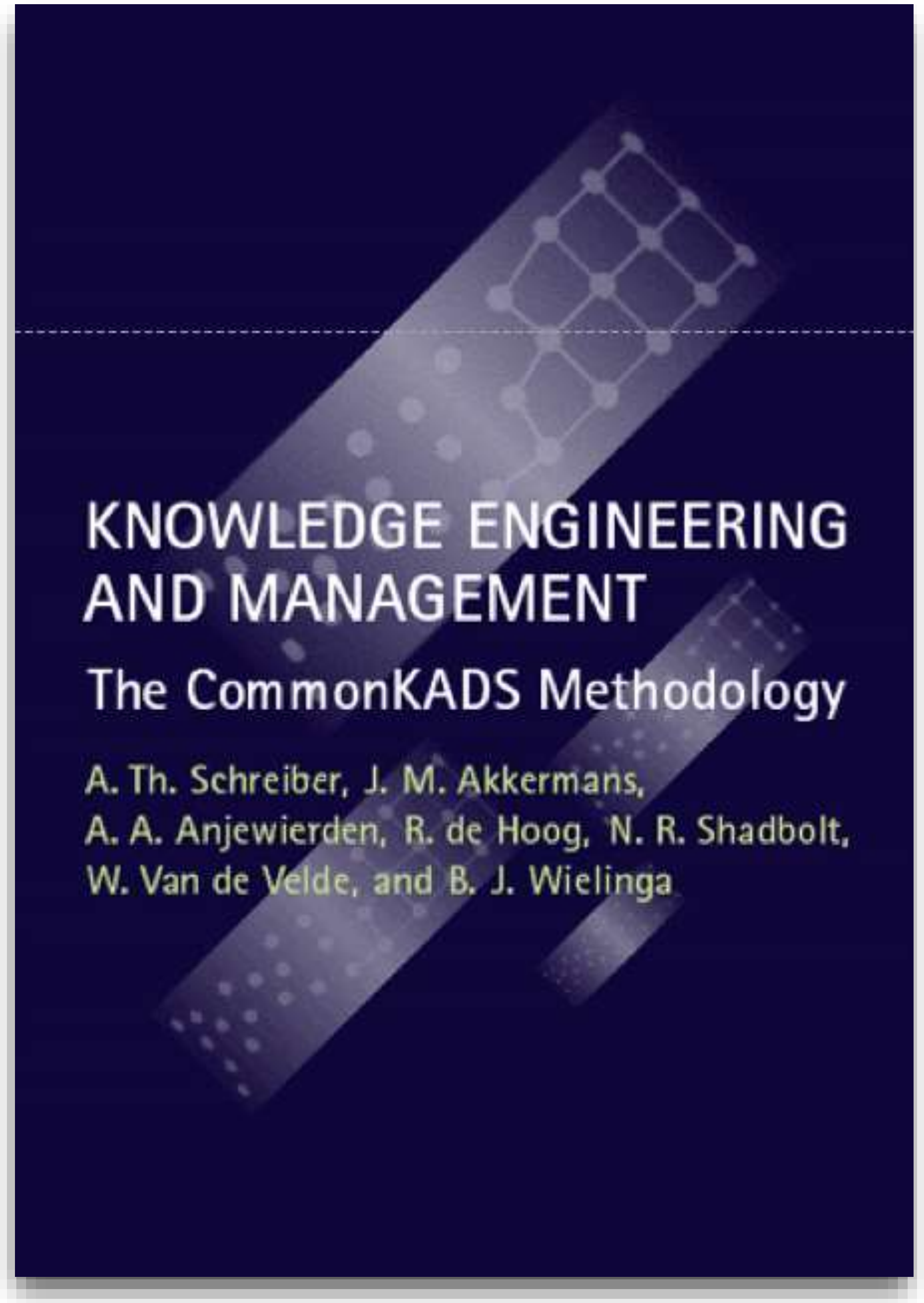
A large language model is a type of artificial intelligence (AI) system designed to understand and generate human-like text. It is trained on vast amounts of data from the internet or other sources, allowing it to learn patterns, language structures, and semantic relationships.

The core idea behind a large language model is to use deep learning techniques, specifically transformer architectures, to process and understand natural language. One of the most well-known transformer architectures is the GPT (Generative Pre-trained Transformer) series, which includes models like GPT-3, GPT-2, and others.

These models consist of multiple layers of neurons and attention mechanisms that enable them to handle large sequences of text and capture long-range dependencies between words. Due to their size, large language models can understand and generate text that is coherent, contextually relevant, and often indistinguishable from text written by humans.

Large language models have a wide range of applications, including language translation, question-answering, text summarization, chatbots, and more. They have significantly advanced the field of natural language processing (NLP) and have the potential to revolutionize the way humans interact with machines and process language-based information.

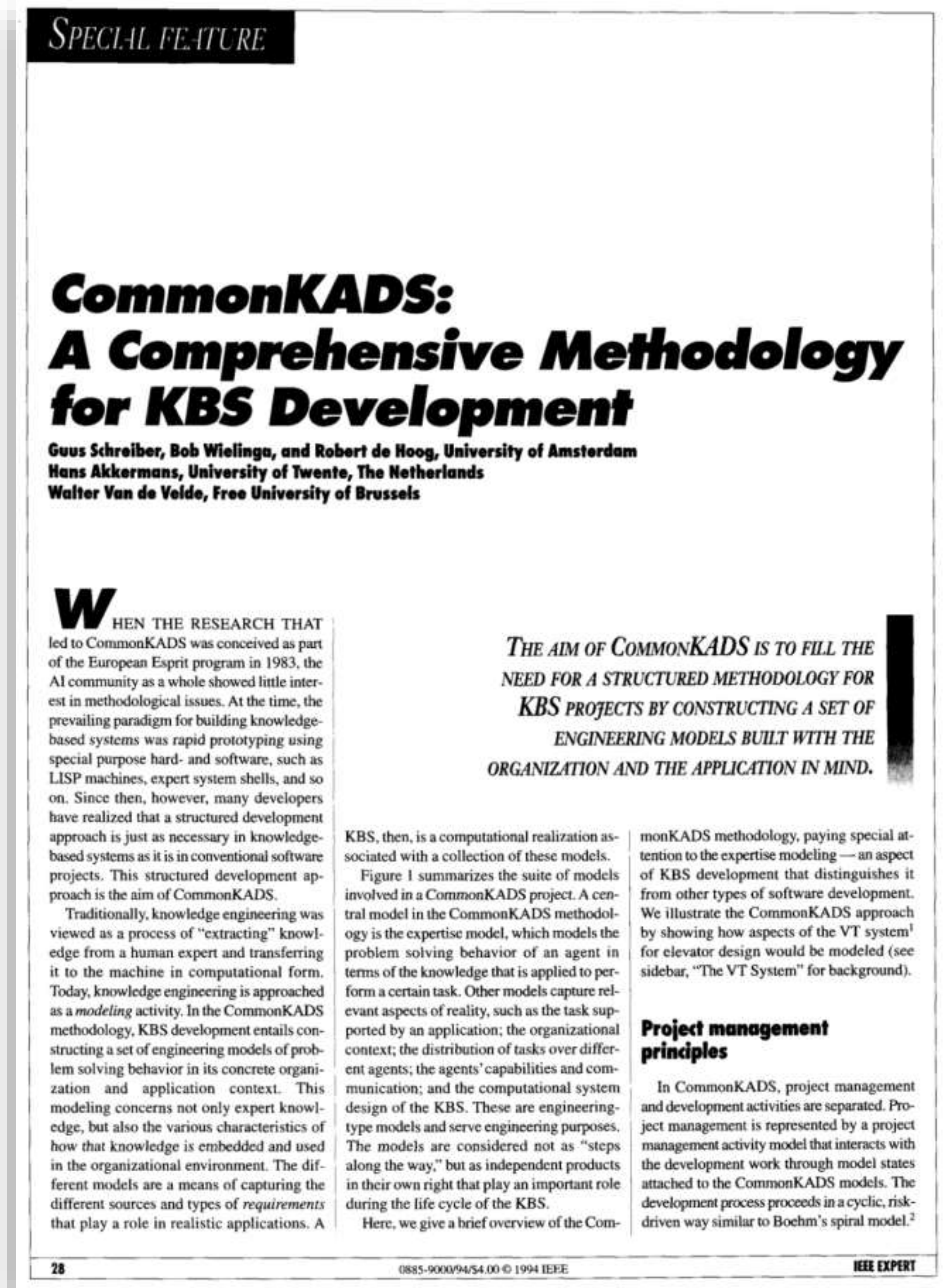
Knowledge Engineering Revisited



KNOWLEDGE ENGINEERING AND MANAGEMENT

The CommonKADS Methodology

A. Th. Schreiber, J. M. Akkermans,
A. A. Anjewierden, R. de Hoog, N. R. Shadbolt,
W. Van de Velde, and B. J. Wielinga



SPECIAL FEATURE

CommonKADS: A Comprehensive Methodology for KBS Development

Guus Schreiber, Bob Wielinga, and Robert de Hoog, University of Amsterdam
Hans Akkermans, University of Twente, The Netherlands
Walter Van de Velde, Free University of Brussels

THE AIM OF COMMONKADS IS TO FILL THE NEED FOR A STRUCTURED METHODOLOGY FOR KBS PROJECTS BY CONSTRUCTING A SET OF ENGINEERING MODELS BUILT WITH THE ORGANIZATION AND THE APPLICATION IN MIND.

WHEN THE RESEARCH THAT led to CommonKADS was conceived as part of the European Esprit program in 1983, the AI community as a whole showed little interest in methodological issues. At the time, the prevailing paradigm for building knowledge-based systems was rapid prototyping using special purpose hard- and software, such as LISP machines, expert system shells, and so on. Since then, however, many developers have realized that a structured development approach is just as necessary in knowledge-based systems as it is in conventional software projects. This structured development approach is the aim of CommonKADS.

Traditionally, knowledge engineering was viewed as a process of "extracting" knowledge from a human expert and transferring it to the machine in computational form. Today, knowledge engineering is approached as a *modeling* activity. In the CommonKADS methodology, KBS development entails constructing a set of engineering models of problem solving behavior in its concrete organization and application context. This modeling concerns not only expert knowledge, but also the various characteristics of how that knowledge is embedded and used in the organizational environment. The different models are a means of capturing the different sources and types of *requirements* that play a role in realistic applications. A KBS, then, is a computational realization associated with a collection of these models.

Figure 1 summarizes the suite of models involved in a CommonKADS project. A central model in the CommonKADS methodology is the expertise model, which models the problem solving behavior of an agent in terms of the knowledge that is applied to perform a certain task. Other models capture relevant aspects of reality, such as the task supported by an application; the organizational context; the distribution of tasks over different agents; the agents' capabilities and communication; and the computational system design of the KBS. These are engineering-type models and serve engineering purposes. The models are considered not as "steps along the way," but as independent products in their own right that play an important role during the life cycle of the KBS.

Here, we give a brief overview of the CommonKADS methodology, paying special attention to the expertise modeling — an aspect of KBS development that distinguishes it from other types of software development. We illustrate the CommonKADS approach by showing how aspects of the VT system[1] for elevator design would be modeled (see sidebar, "The VT System" for background).

Project management principles

In CommonKADS, project management and development activities are separated. Project management is represented by a project management activity model that interacts with the development work through model states attached to the CommonKADS models. The development process proceeds in a cyclic, risk-driven way similar to Boehm's spiral model.[2]

28 0885-9000/94/$4.00 © 1994 IEEE IEEE EXPERT

My first motivation: building knowledge graphs

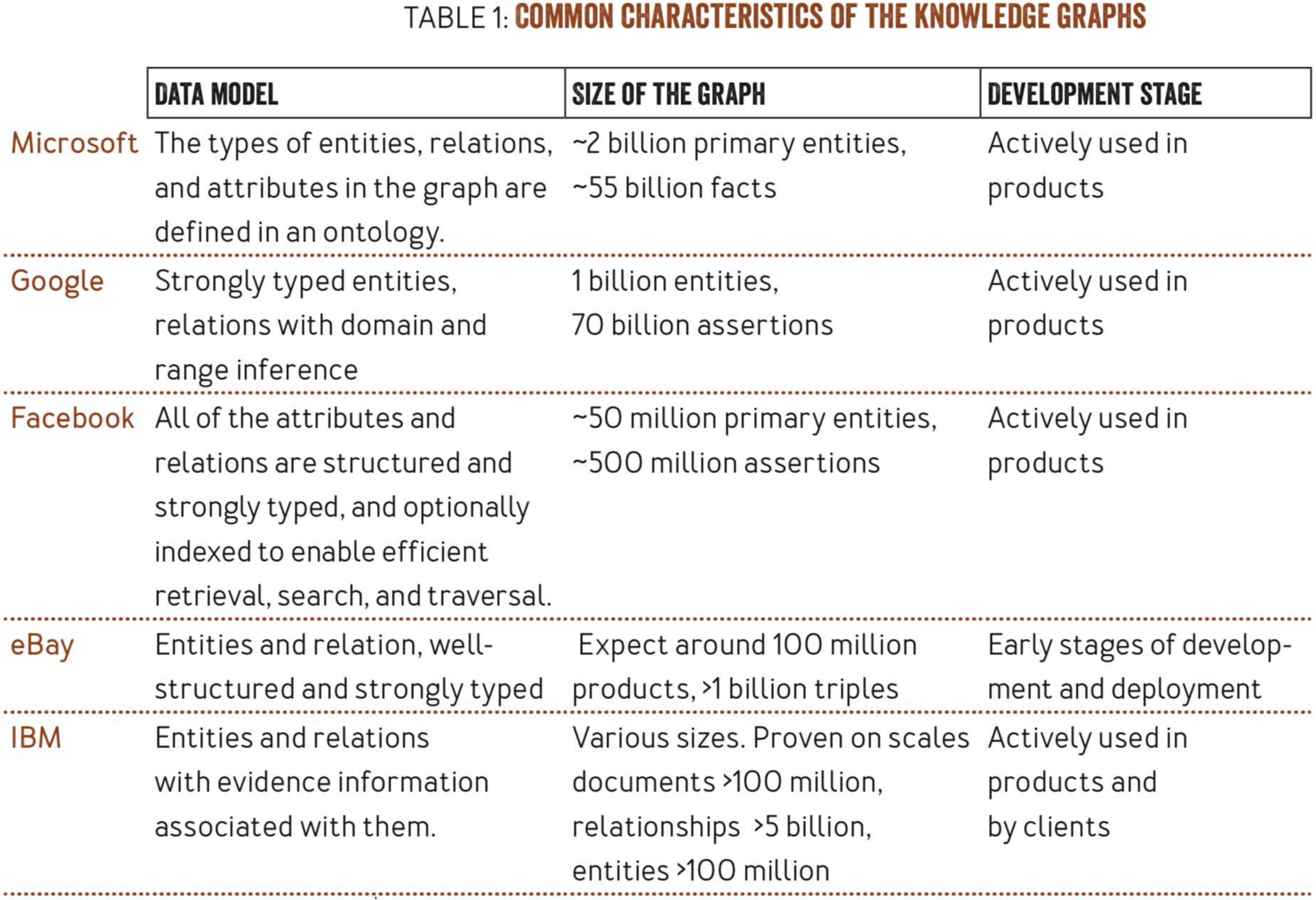


TABLE 1: **COMMON CHARACTERISTICS OF THE KNOWLEDGE GRAPHS**

	DATA MODEL	SIZE OF THE GRAPH	DEVELOPMENT STAGE
Microsoft	The types of entities, relations, and attributes in the graph are defined in an ontology.	~2 billion primary entities, ~55 billion facts	Actively used in products
Google	Strongly typed entities, relations with domain and range inference	1 billion entities, 70 billion assertions	Actively used in products
Facebook	All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal.	~50 million primary entities, ~500 million assertions	Actively used in products
eBay	Entities and relation, well-structured and strongly typed	Expect around 100 million products, >1 billion triples	Early stages of development and deployment
IBM	Entities and relations with evidence information associated with them.	Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million	Actively used in products and by clients

Source:
Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale Knowledge Graphs: Lessons and Challenges. Queue 17, 2, pages 20 (April 2019), 28 pages. DOI: https://doi.org/10.1145/3329781.3332266

What does the knowledge graph development cycle look like?

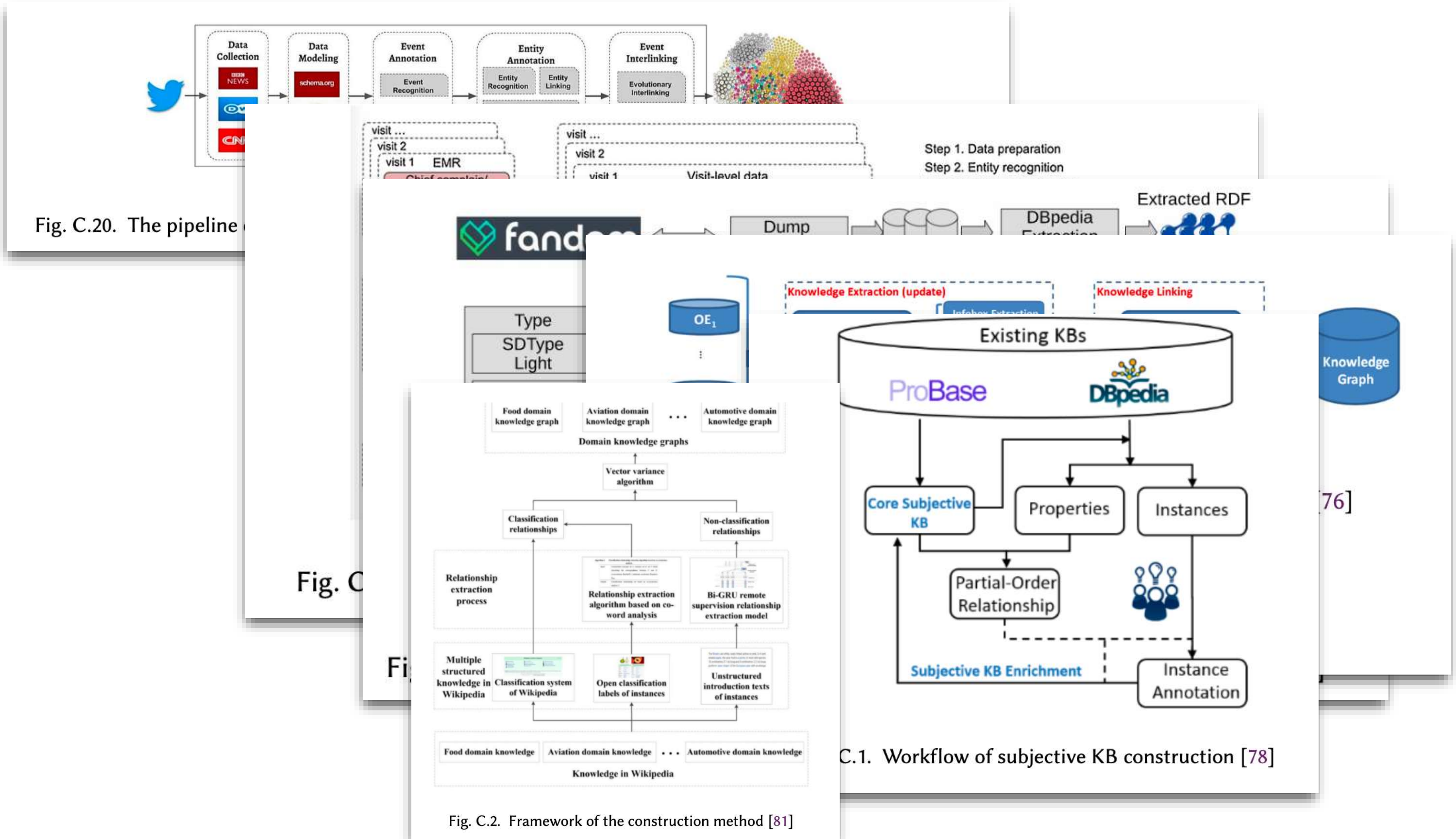

Data Collection
NEWS
Data Modeling
schema.org
Event Annotation
Event Recognition
Entity Annotation
Entity Recognition
Entity Linking
Event Interlinking
Evolutionary Interlinking
visit ...
visit 2
visit 1
EMR
visit ...
visit 2
visit 1
Visit-level data
Step 1. Data preparation
Step 2. Entity recognition
fandom
Dump
DBpedia
Extracted RDF
Type
SDType
Light
OE1
Knowledge Extraction (update)
Knowledge Linking
Knowledge Graph
Existing KBs
ProBase
DBpedia
Core Subjective KB
Properties
Instances
Partial-Order Relationship
Subjective KB Enrichment
Instance Annotation
Food domain knowledge graph
Aviation domain knowledge graph
Automotive domain knowledge graph
Domain knowledge graphs
Vector variance algorithm
Classification relationships
Non-classification relationships
Relationship extraction process
Relationship extraction algorithm based on co-word analysis
Bi-GRU remote supervision relationship extraction model
Multiple structured knowledge in Wikipedia
Classification system of Wikipedia
Open classification labels of instances
Unstructured introduction texts of instances
Food domain knowledge
Aviation domain knowledge
Automotive domain knowledge
Knowledge in Wikipedia

Fig. C.20. The pipeline

Fig. C

Fi

[76]

C.1. Workflow of subjective KB construction [78]

Fig. C.2. Framework of the construction method [81]

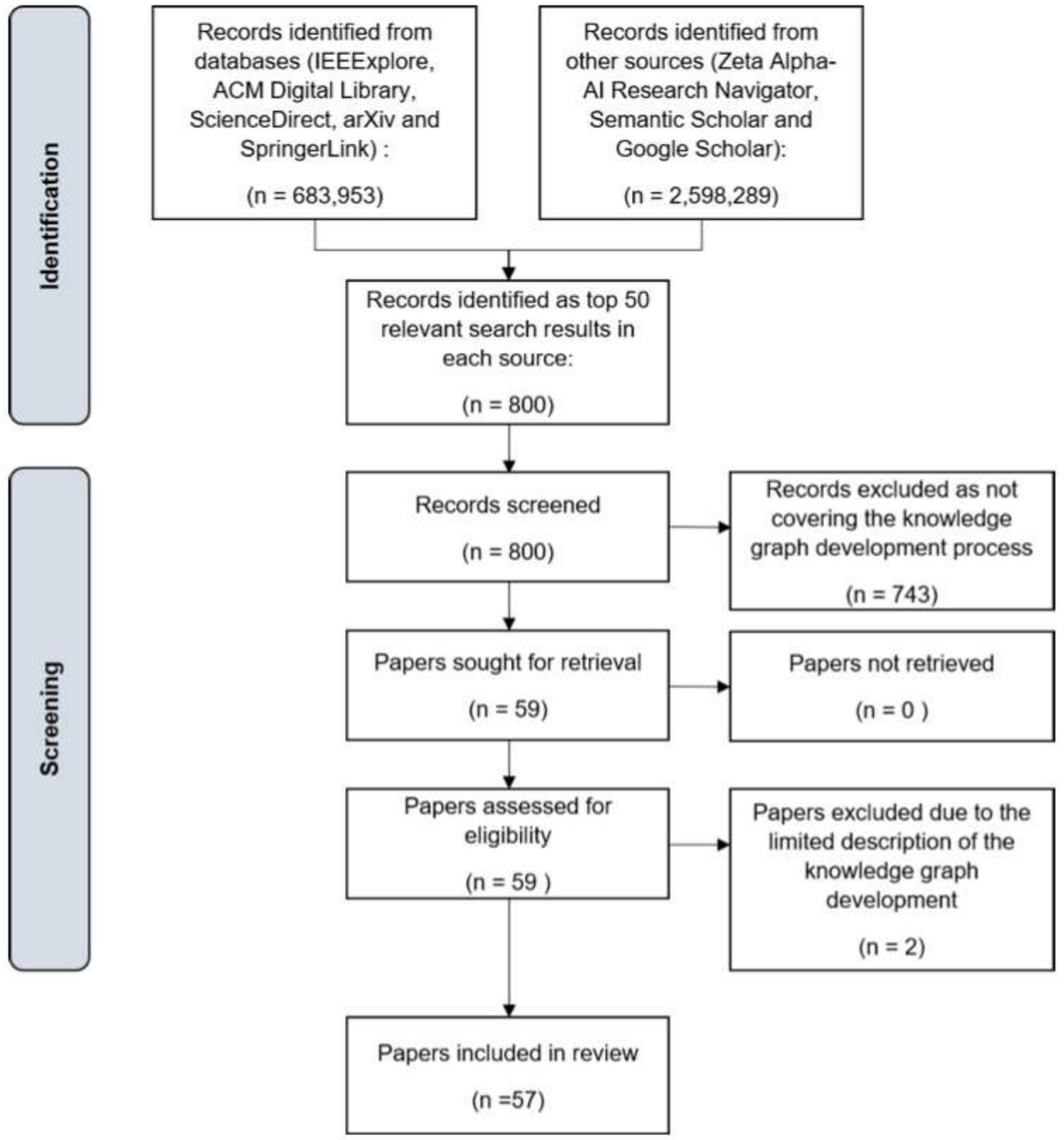

Identification
Screening
Records identified from databases (IEEExplore, ACM Digital Library, ScienceDirect, arXiv and SpringerLink) :
(n = 683,953)
Records identified from other sources (Zeta Alpha- AI Research Navigator, Semantic Scholar and Google Scholar):
(n = 2,598,289)
Records identified as top 50 relevant search results in each source:
(n = 800)
Records screened
(n = 800)
Records excluded as not covering the knowledge graph development process
(n = 743)
Papers sought for retrieval
(n = 59)
Papers not retrieved
(n = 0)
Papers assessed for eligibility
(n = 59)
Papers excluded due to the limited description of the knowledge graph development
(n = 2)
Papers included in review
(n =57)

Fig. 2. The PRISMA workflow of selecting papers for the systematic review

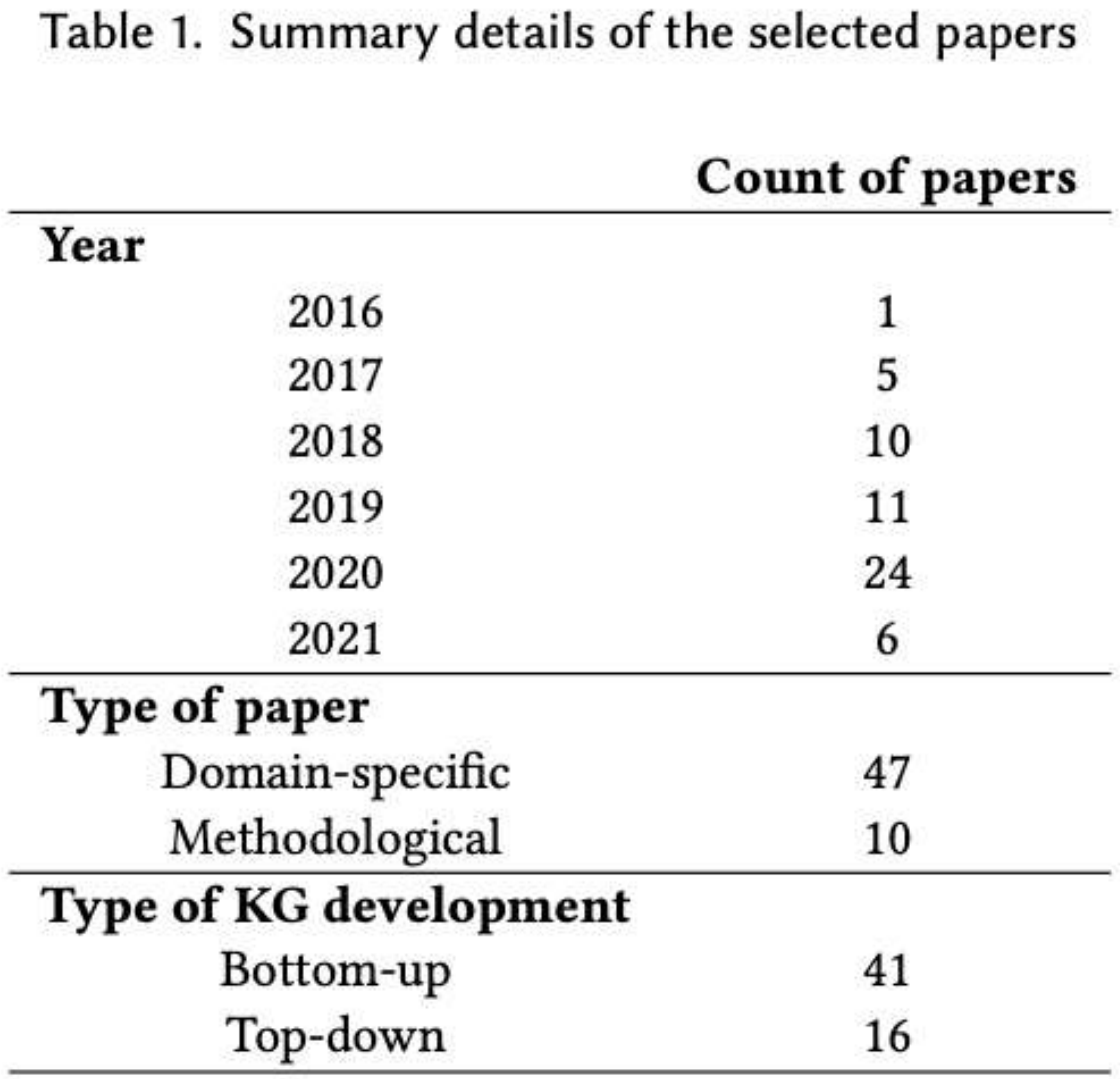
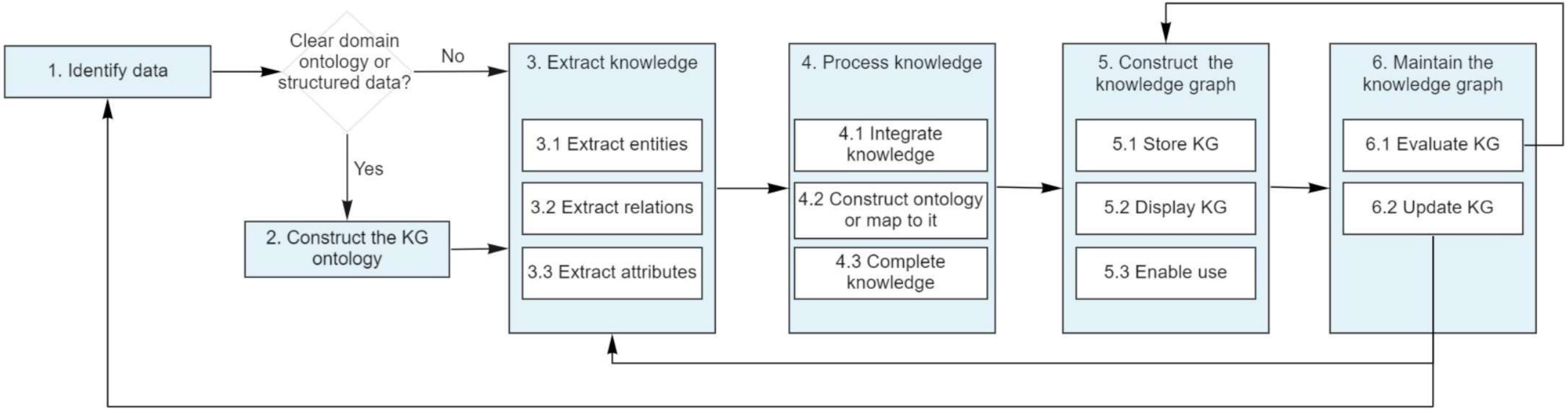


Table 1. Summary details of the selected papers

		Count of papers
Year		
	2016	1
	2017	5
	2018	10
	2019	11
	2020	24
	2021	6
Type of paper		
	Domain-specific	47
	Methodological	10
Type of KG development		
	Bottom-up	41
	Top-down	16

Gytė Tamašauskaitė and Paul Groth. 2022. Defining a Knowledge Graph Development Process Through a Systematic Review. ACM Trans. Softw. Eng. Methodol. Just Accepted (February 2022). https://doi.org/10.1145/3522586



1. Identify data

Clear domain ontology or structured data?

No

Yes

2. Construct the KG ontology

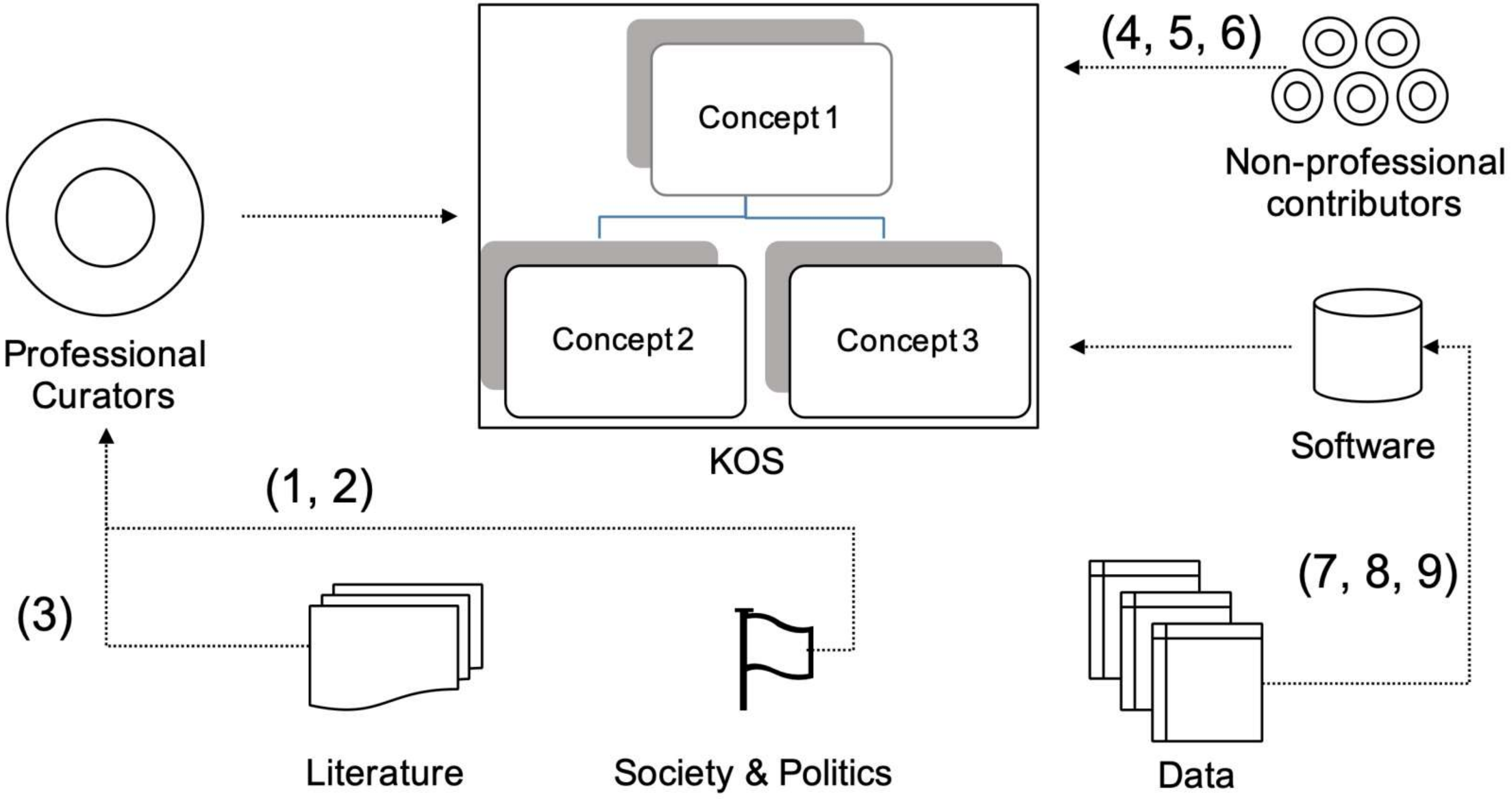
3. Extract knowledge
- 3.1 Extract entities
- 3.2 Extract relations
- 3.3 Extract attributes

4. Process knowledge
- 4.1 Integrate knowledge
- 4.2 Construct ontology or map to it
- 4.3 Complete knowledge

5. Construct the knowledge graph
- 5.1 Store KG
- 5.2 Display KG
- 5.3 Enable use

6. Maintain the knowledge graph
- 6.1 Evaluate KG
- 6.2 Update KG

Gytė Tamašauskaitė and Paul Groth. 2023. Defining a Knowledge Graph Development Process Through a Systematic Review. ACM Trans. Softw. Eng. Methodol. 32, 1, Article 27 (January 2023), 40 pages.
https://doi.org/10.1145/3522586

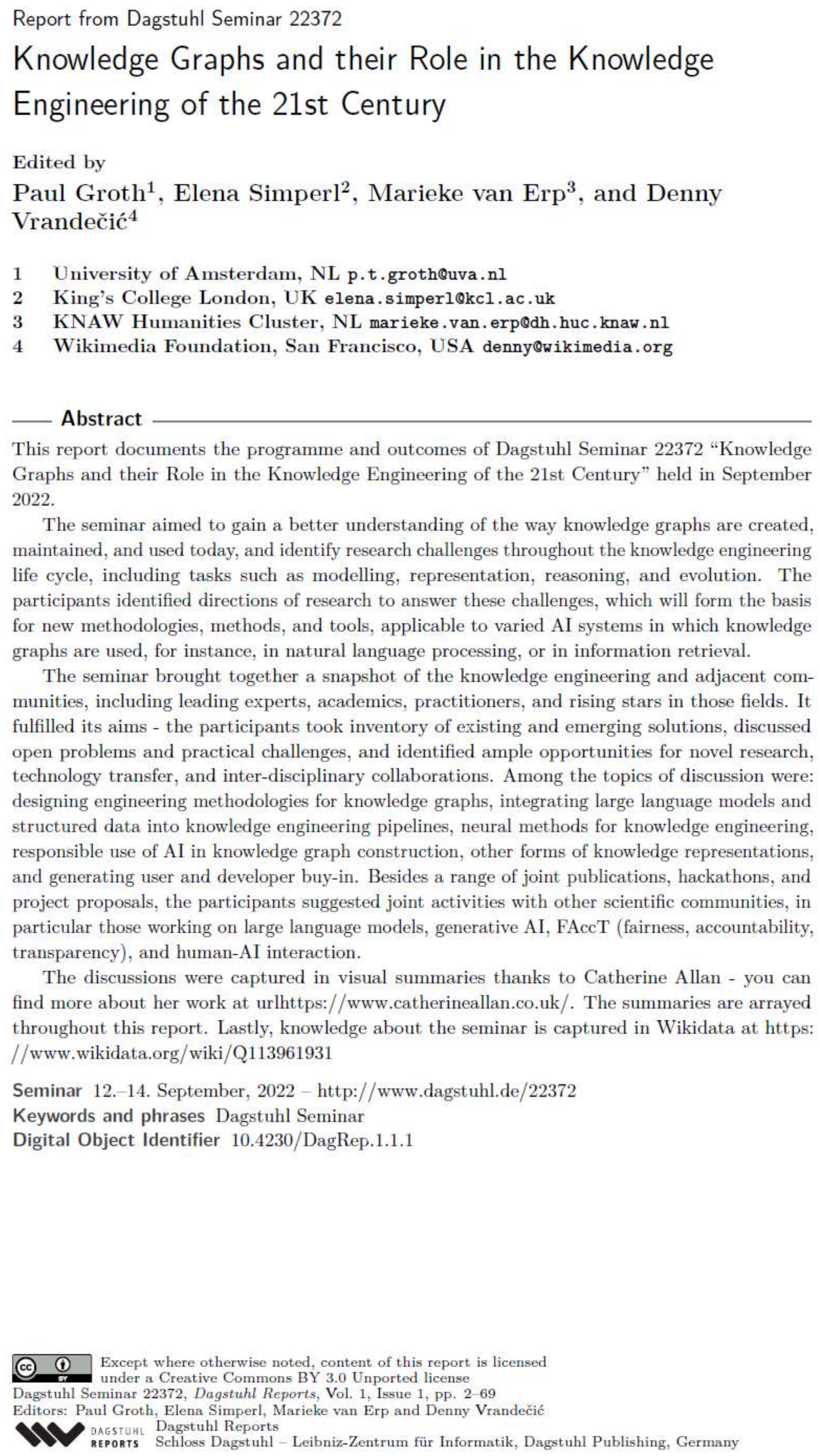


1. dealing with changing cultural and societal norms, specifically to address or correct bias;
2. political influence
3. new concepts and terminology arising from discoveries or change in perspective within a technical/scientific community
4. gardening
5. incremental contributorship
6. progressive formalization
7. software and automation
8. integration of large numbers of data sources
9. variance in algorithm training data

Source:
Michael Lauruhn and Paul Groth.
“Sources of Change for Modern Knowledge Organization Systems.” Knowledge Organization 43, no. 8 (2016).

Timely question

- Dagstuhl seminar 22372, 11-14.09.2022
- Organised by
  - Paul Groth (University of Amsterdam, NL)
  - Elena Simperl (King's College London, UK)
  - Marieke van Erp (KNAW Humanities Cluster - Amsterdam, NL)
  - Denny Vrandecic (Wikimedia - San Francisco, US)
- More information at: https://www.dagstuhl.de/seminars/seminar-calendar/seminar-details/22372
- Other places too: aaai-make.info



Report from Dagstuhl Seminar 22372

Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century

Edited by
Paul Groth[1], Elena Simperl[2], Marieke van Erp[3], and Denny Vrandečić[4]

1 University of Amsterdam, NL `p.t.groth@uva.nl`
2 King's College London, UK `elena.simperl@kcl.ac.uk`
3 KNAW Humanities Cluster, NL `marieke.van.erp@dh.huc.knaw.nl`
4 Wikimedia Foundation, San Francisco, USA `denny@wikimedia.org`

Abstract

This report documents the programme and outcomes of Dagstuhl Seminar 22372 "Knowledge Graphs and their Role in the Knowledge Engineering of the 21st Century" held in September 2022.

The seminar aimed to gain a better understanding of the way knowledge graphs are created, maintained, and used today, and identify research challenges throughout the knowledge engineering life cycle, including tasks such as modelling, representation, reasoning, and evolution. The participants identified directions of research to answer these challenges, which will form the basis for new methodologies, methods, and tools, applicable to varied AI systems in which knowledge graphs are used, for instance, in natural language processing, or in information retrieval.

The seminar brought together a snapshot of the knowledge engineering and adjacent communities, including leading experts, academics, practitioners, and rising stars in those fields. It fulfilled its aims - the participants took inventory of existing and emerging solutions, discussed open problems and practical challenges, and identified ample opportunities for novel research, technology transfer, and inter-disciplinary collaborations. Among the topics of discussion were: designing engineering methodologies for knowledge graphs, integrating large language models and structured data into knowledge engineering pipelines, neural methods for knowledge engineering, responsible use of AI in knowledge graph construction, other forms of knowledge representations, and generating user and developer buy-in. Besides a range of joint publications, hackathons, and project proposals, the participants suggested joint activities with other scientific communities, in particular those working on large language models, generative AI, FAccT (fairness, accountability, transparency), and human-AI interaction.

The discussions were captured in visual summaries thanks to Catherine Allan - you can find more about her work at urlhttps://www.catherineallan.co.uk/. The summaries are arrayed throughout this report. Lastly, knowledge about the seminar is captured in Wikidata at https://www.wikidata.org/wiki/Q113961931

Seminar 12.–14. September, 2022 – http://www.dagstuhl.de/22372
Keywords and phrases Dagstuhl Seminar
Digital Object Identifier 10.4230/DagRep.1.1.1

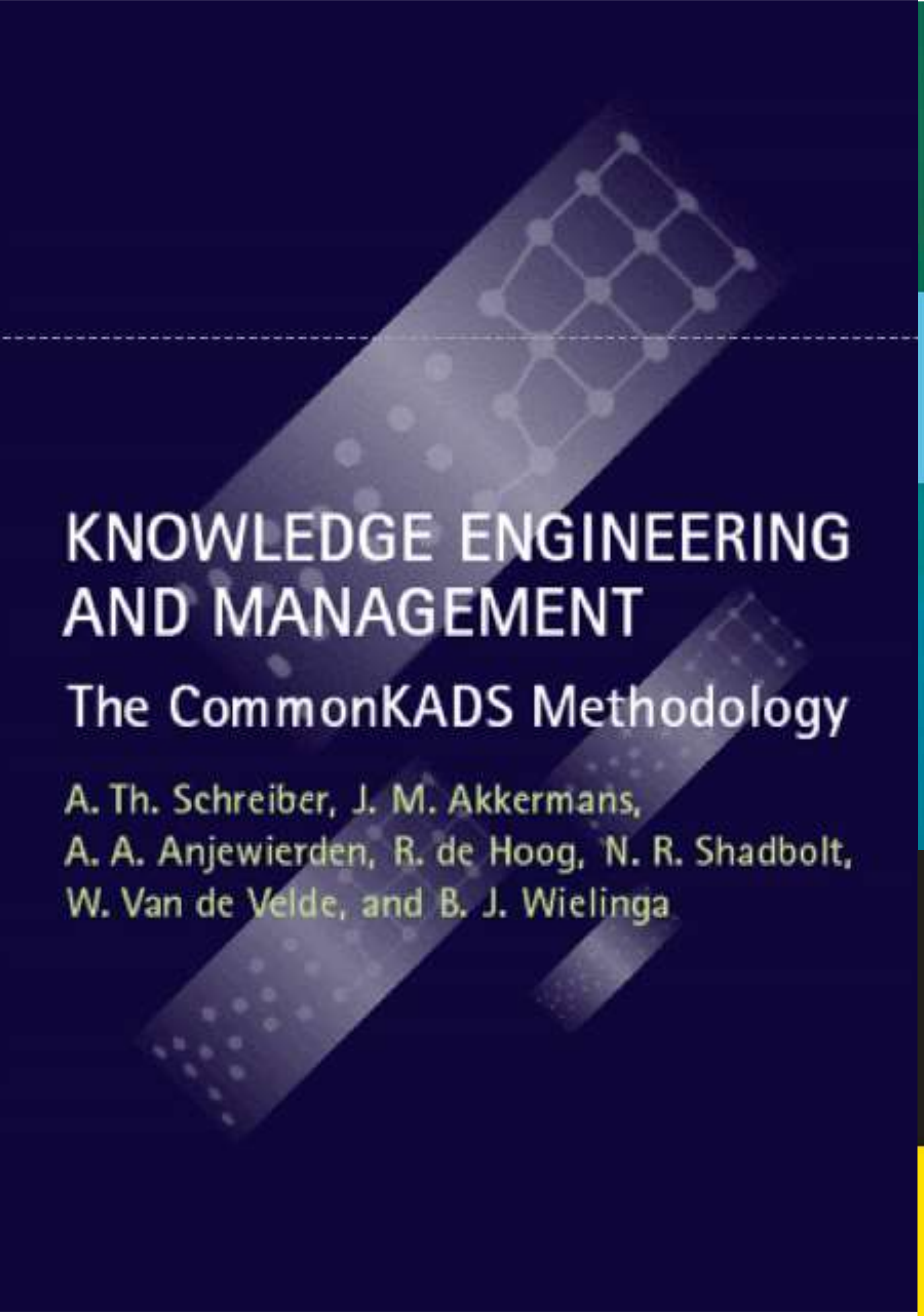
Except where otherwise noted, content of this report is licensed
under a Creative Commons BY 3.0 Unported license
Dagstuhl Seminar 22372, *Dagstuhl Reports*, Vol. 1, Issue 1, pp. 2–69
Editors: Paul Groth, Elena Simperl, Marieke van Erp and Denny Vrandečić
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Marcel R. Ackermann (Schloss Dagstuhl - Trier, DE), Mehwish Alam (FIZ Karlsruhe, DE), Bradley Allen (Merit - Millbrae, US), Sören Auer (TIB - Hannover, DE), Eva Blomqvist (Linköping University, SE), George Fletcher (TU Eindhoven, NL), Paul Groth (University of Amsterdam, NL), Aidan Hogan (University of Chile - Santiago de Chile, CL), Filip Ilievski (USC - Marina del Rey, US), Antoine Isaac (Europeana Foundation - Den Haag, NL), Diana Maynard (University of Sheffield, UK), Deborah L. McGuinness (Rensselaer Polytechnic Institute - Troy, US), Axel-Cyrille Ngonga Ngomo (Universität Paderborn, DE), Heiko Paulheim (Universität Mannheim, DE), Lydia Pintscher (Wikimedia - Germany, DE), Valentina Presutti (University of Bologna, IT), Florian Reitz (Schloss Dagstuhl - Trier, DE), Marta Sabou (Wirtschaftsuniversität Wien, AT), Harald Sack (FIZ Karlsruhe, DE), Stefan Schlobach (VU University Amsterdam, NL), Juan F. Sequeda (data.world - Austin, US), Elena Simperl (King's College London, UK), Steffen Staab (Universität Stuttgart, DE), Lise Stork (VU University Amsterdam, NL), Hideaki Takeda (National Institute of Informatics - Tokyo, JP), Katherine Thornton (Yale University Library - New Haven, US), Marieke van Erp (KNAW Humanities Cluster - Amsterdam, NL), Denny Vrandecic (Wikimedia - San Francisco, US)

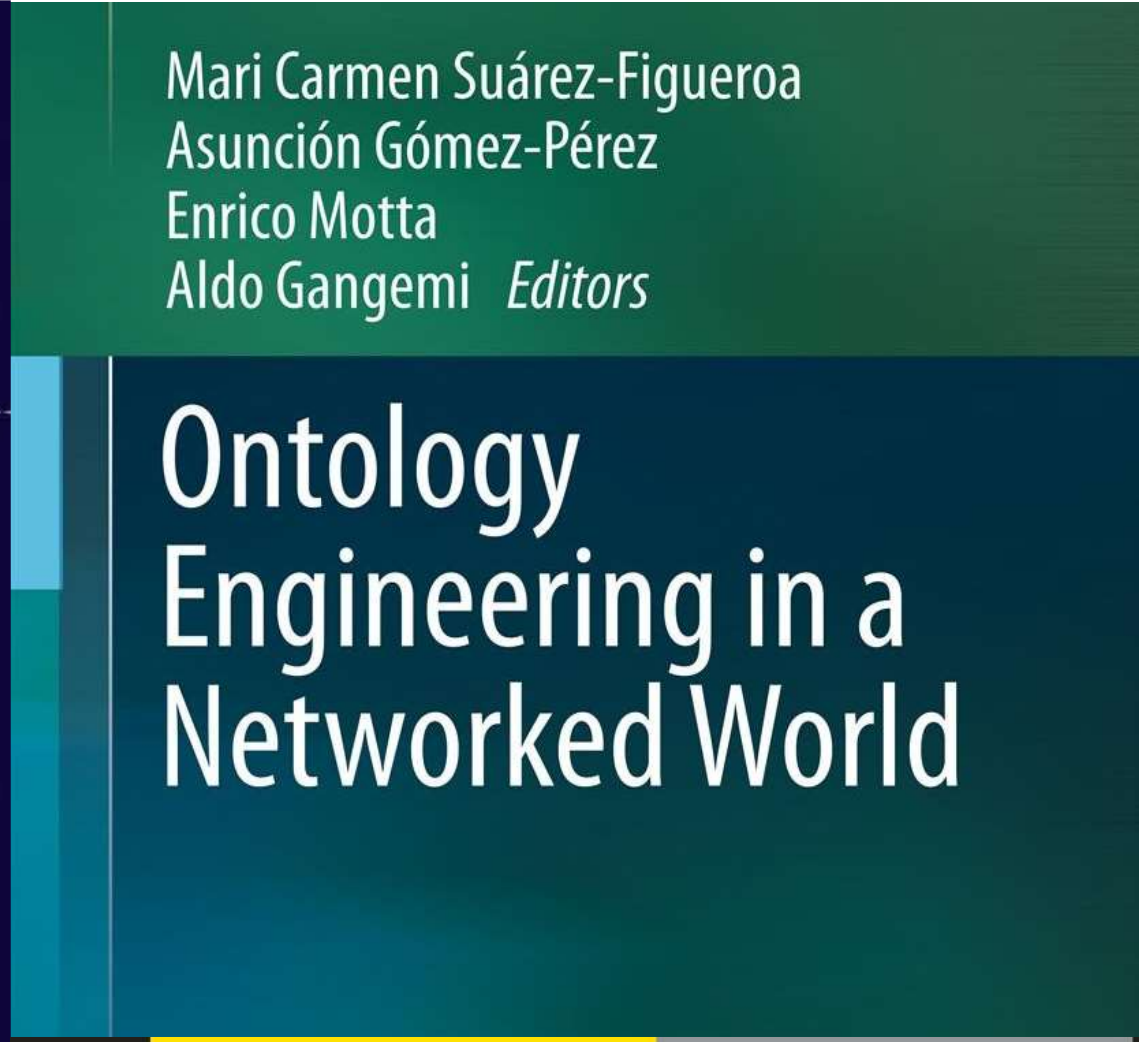
CHLOSS DAGSTUHL – LZI GMBH

sed under Creative Commons License CC BY-NC-ND

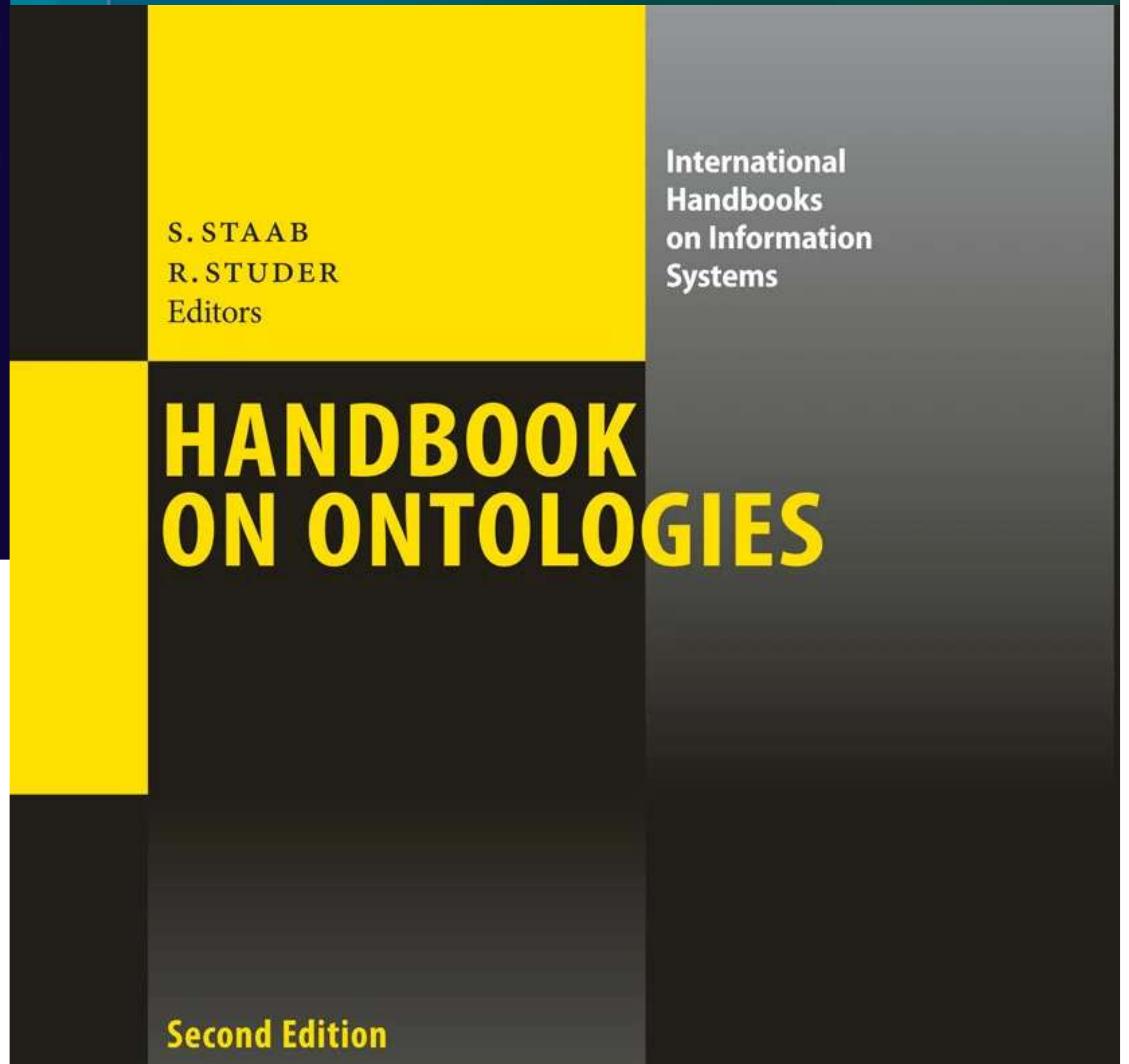
Knowledge engineering: before



KNOWLEDGE ENGINEERING
AND MANAGEMENT
The CommonKADS Methodology
A. Th. Schreiber, J. M. Akkermans,
A. A. Anjewierden, R. de Hoog, N. R. Shadbolt,
W. Van de Velde, and B. J. Wielinga



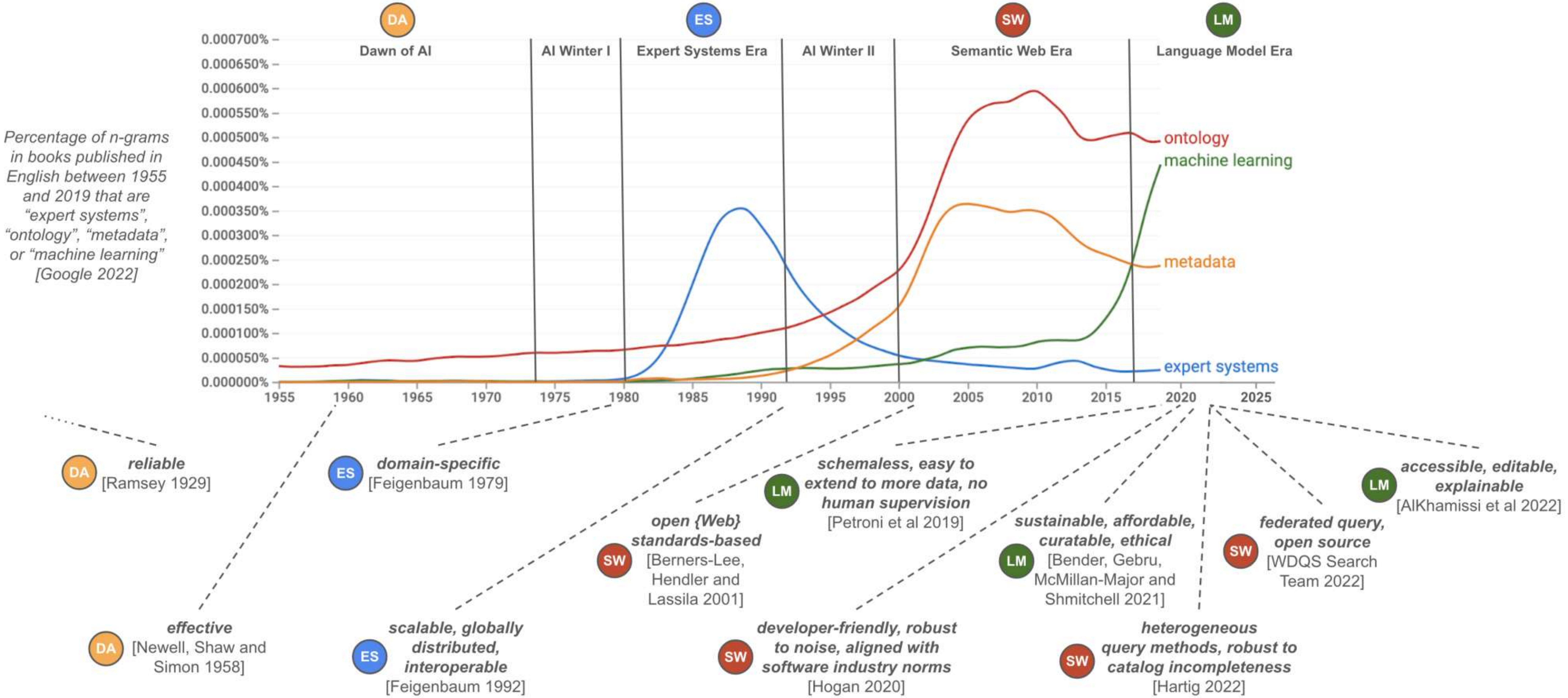
Mari Carmen Suárez-Figueroa
Asunción Gómez-Pérez
Enrico Motta
Aldo Gangemi *Editors*
Ontology
Engineering in a
Networked World



S. STAAB
R. STUDER
Editors
International
Handbooks
on Information
Systems
HANDBOOK
ON ONTOLOGIES
Second Edition

- Gathering highly curated knowledge from experts and encoding it into computational representations in knowledge bases.
- Mostly manual process, focusing on how knowledge was structured and organised rather than the domain data.
- Results used in expert systems, requiring considerable up-front investment.

KE Requirements over time



Percentage of n-grams in books published in English between 1955 and 2019 that are "expert systems", "ontology", "metadata", or "machine learning" [Google 2022]

DA ***reliable*** [Ramsey 1929]

ES ***domain-specific*** [Feigenbaum 1979]

LM ***schemaless, easy to extend to more data, no human supervision*** [Petroni et al 2019]

LM ***accessible, editable, explainable*** [AlKhamissi et al 2022]

SW ***open {Web} standards-based*** [Berners-Lee, Hendler and Lassila 2001]

LM ***sustainable, affordable, curatable, ethical*** [Bender, Gebru, McMillan-Major and Shmitchell 2021]

SW ***federated query, open source*** [WDQS Search Team 2022]

DA ***effective*** [Newell, Shaw and Simon 1958]

ES ***scalable, globally distributed, interoperable*** [Feigenbaum 1992]

SW ***developer-friendly, robust to noise, aligned with software industry norms*** [Hogan 2020]

SW ***heterogeneous query methods, robust to catalog incompleteness*** [Hartig 2022]

See Allen et. al 2023
https://arxiv.org/abs/2306.15124

Knowledge engineering: today

Automatic process with human-in-the-loop

Large knowledge bases, drawn from heterogeneous data, using a mix of data management, machine learning, knowledge representation, crowdsourcing

Provided access to data and (off-the-shelf) AI capabilities, costs are a fraction from what they were decades ago.

This has led to mainstream adoption in search, intelligent assistants, digital twins, supply chain management, legal compliance etc.

My assumption:
We need a updated theory of knowledge engineering for modern LLM + KG systems

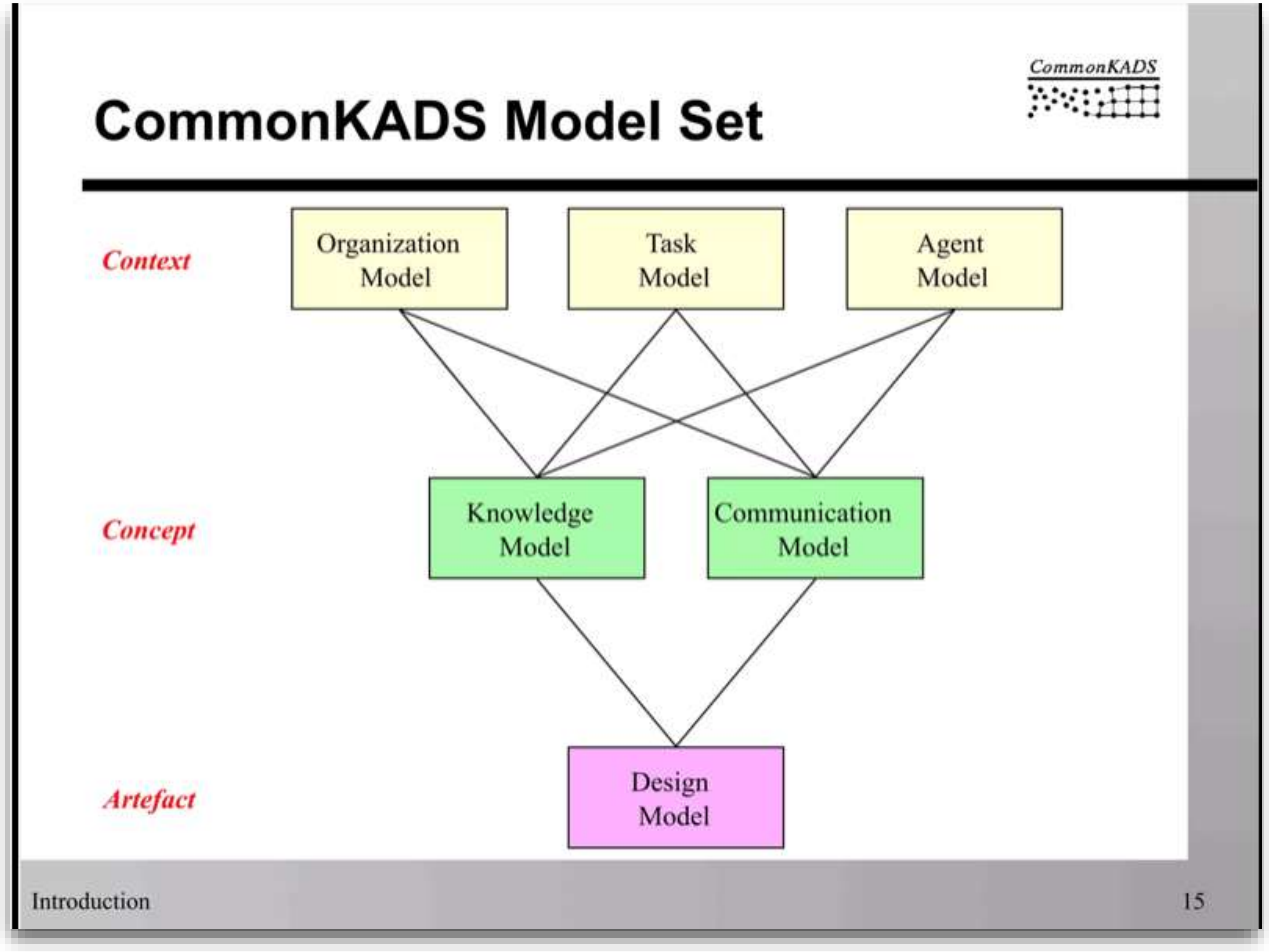
My assumption:
We need a updated theory of knowledge engineering for modern LLM + KG systems

But maybe I’m wrong

CommonKADS in short

Key principles

1. KE constructing different aspect models of human knowledge.
2. Operate at the knowledge level
3. Knowledge has a stable internal structure that is analyzable by distinguishing specific knowledge types and roles.
4. Managed by learning from your experiences in a controlled "spiral" way.



CommonKADS

CommonKADS Model Set

Introduction 15

https://commonkads.org

Prompting as Programming

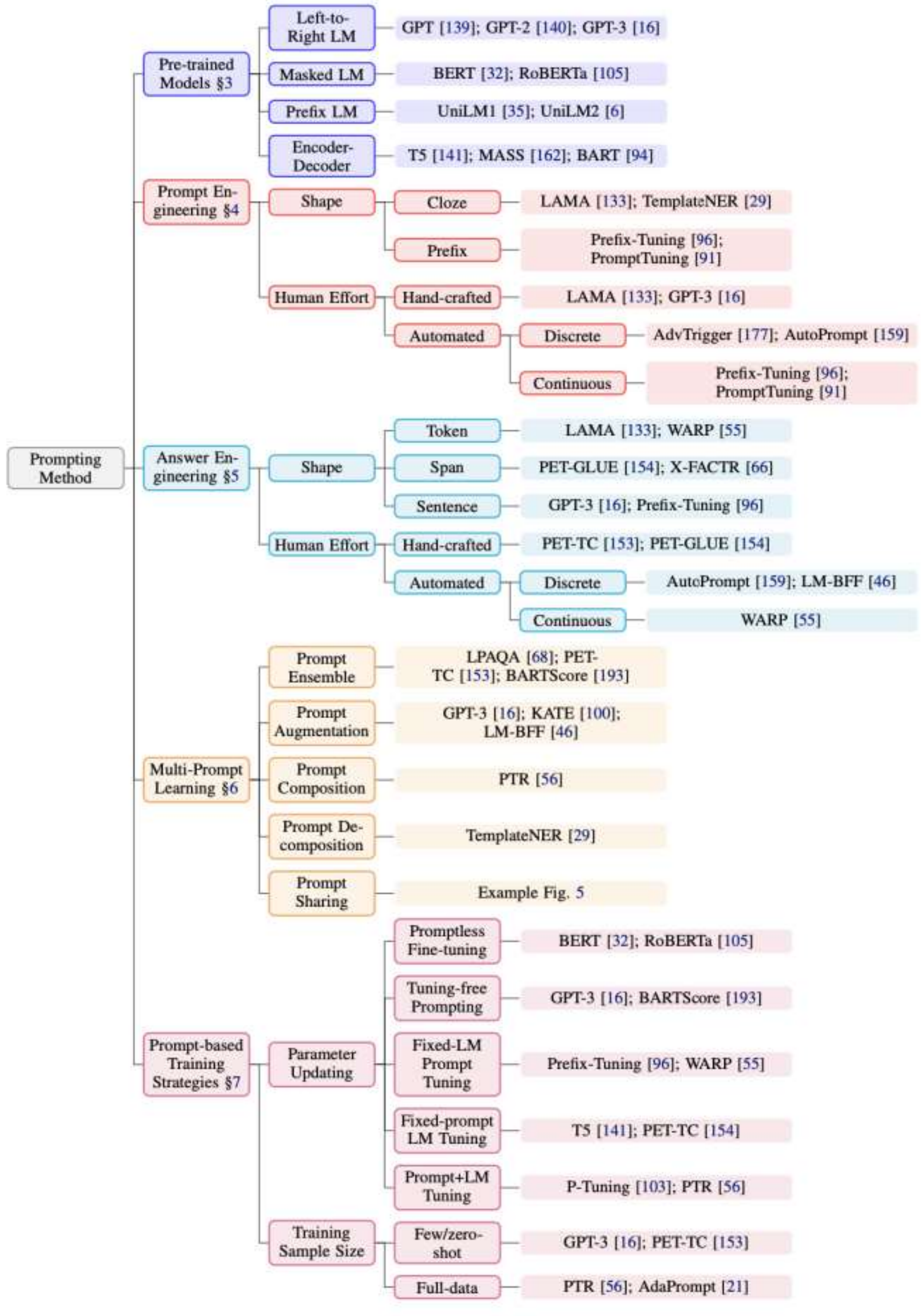
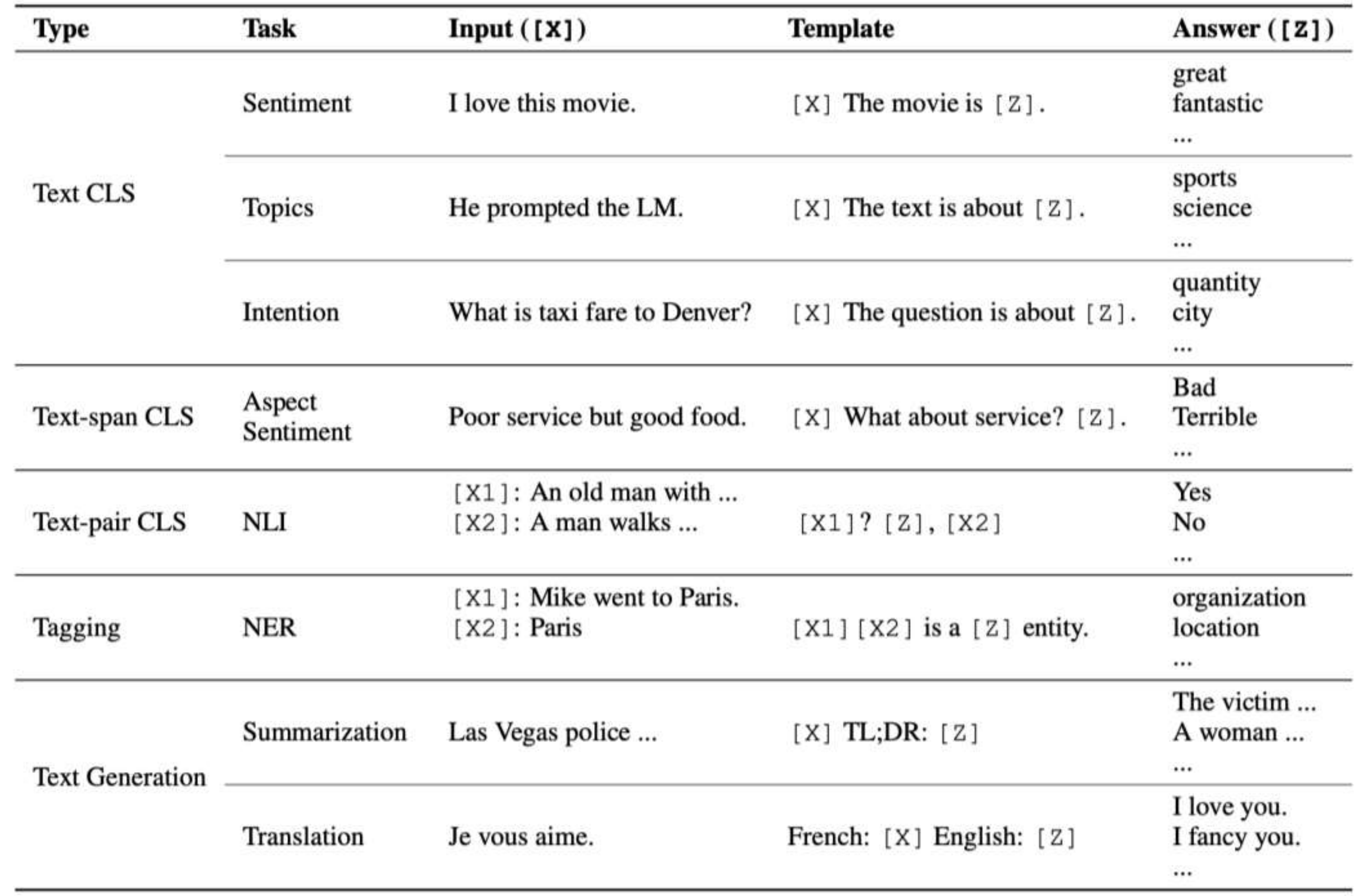


Figure 1: Typology of prompting methods.

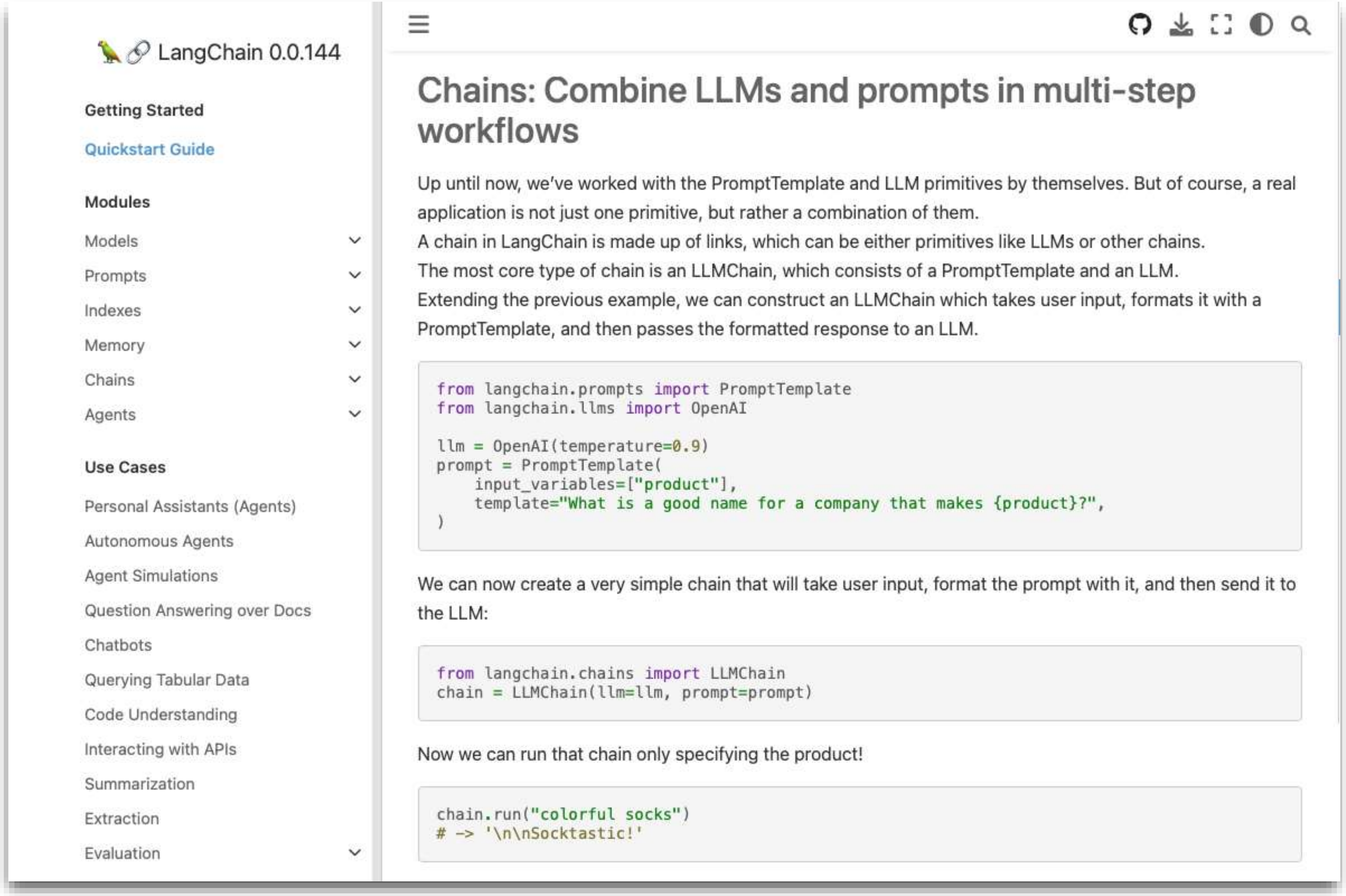
7



Type	**Task**	**Input ([X])**	**Template**	**Answer ([Z])**
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

Table 3: Examples of *input*, *template*, and *answer* for different tasks. In the **Type** column, "CLS" is an abbreviation for "classification". In the **Task** column, "NLI" and "NER" are abbreviations for "natural language inference" (Bowman et al., 2015) and "named entity recognition" (Tjong Kim Sang and De Meulder, 2003) respectively.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. Comput. Surveys 55, 9 (2023), 1–35.



Chains: Combine LLMs and prompts in multi-step workflows

Up until now, we've worked with the PromptTemplate and LLM primitives by themselves. But of course, a real application is not just one primitive, but rather a combination of them.
A chain in LangChain is made up of links, which can be either primitives like LLMs or other chains.
The most core type of chain is an LLMChain, which consists of a PromptTemplate and an LLM.
Extending the previous example, we can construct an LLMChain which takes user input, formats it with a PromptTemplate, and then passes the formatted response to an LLM.

```python
from langchain.prompts import PromptTemplate
from langchain.llms import OpenAI

llm = OpenAI(temperature=0.9)
prompt = PromptTemplate(
    input_variables=["product"],
    template="What is a good name for a company that makes {product}?",
)
```

We can now create a very simple chain that will take user input, format the prompt with it, and then send it to the LLM:

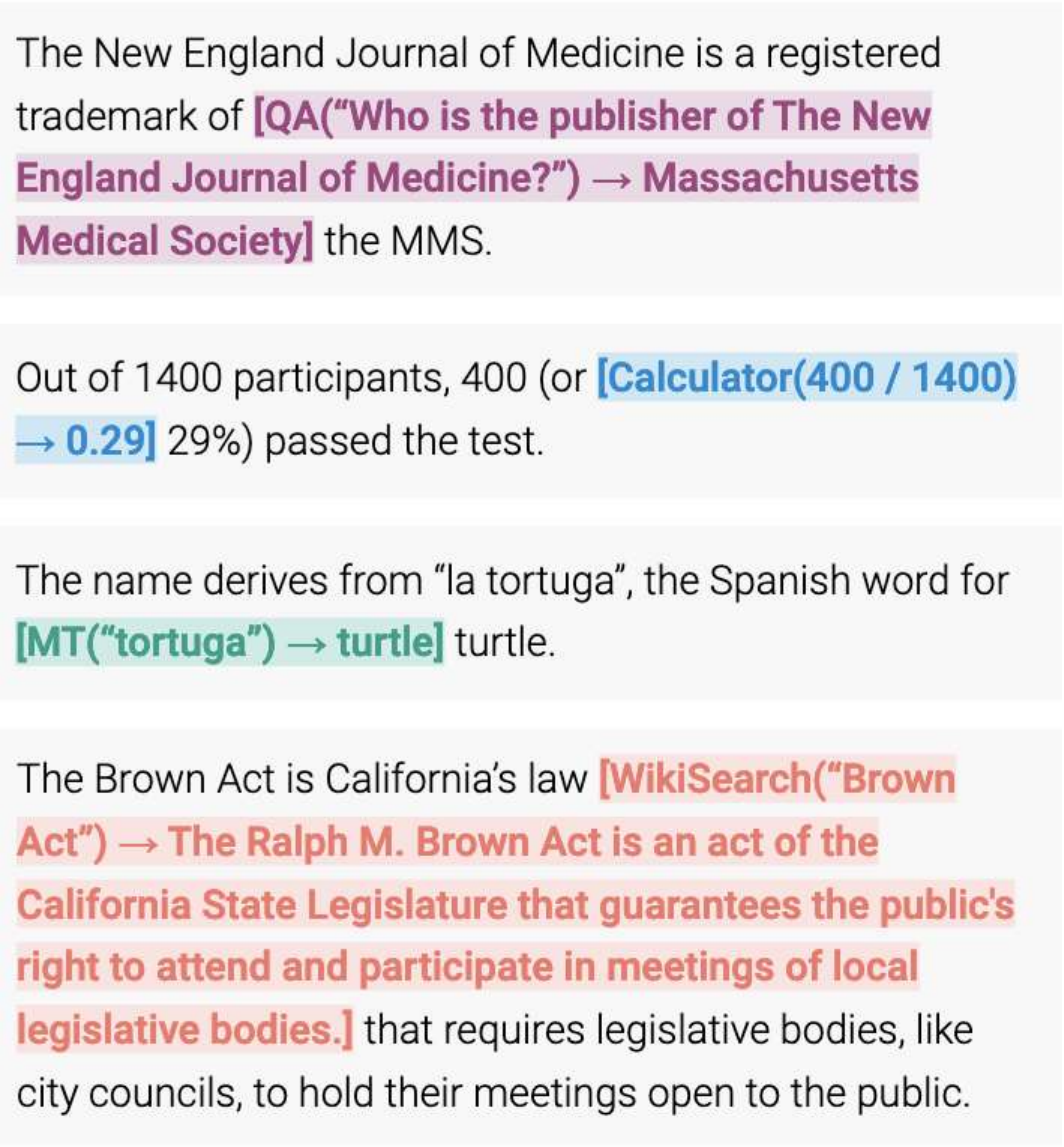
```python
from langchain.chains import LLMChain
chain = LLMChain(llm=llm, prompt=prompt)
```

Now we can run that chain only specifying the product!

```python
chain.run("colorful socks")
# -> '\n\nSocktastic!'
```

https://python.langchain.com/

Connecting LLMs into the wider world



The New England Journal of Medicine is a registered trademark of **[QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society]** the MMS.

Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

The name derives from "la tortuga", the Spanish word for **[MT("tortuga") → turtle]** turtle.

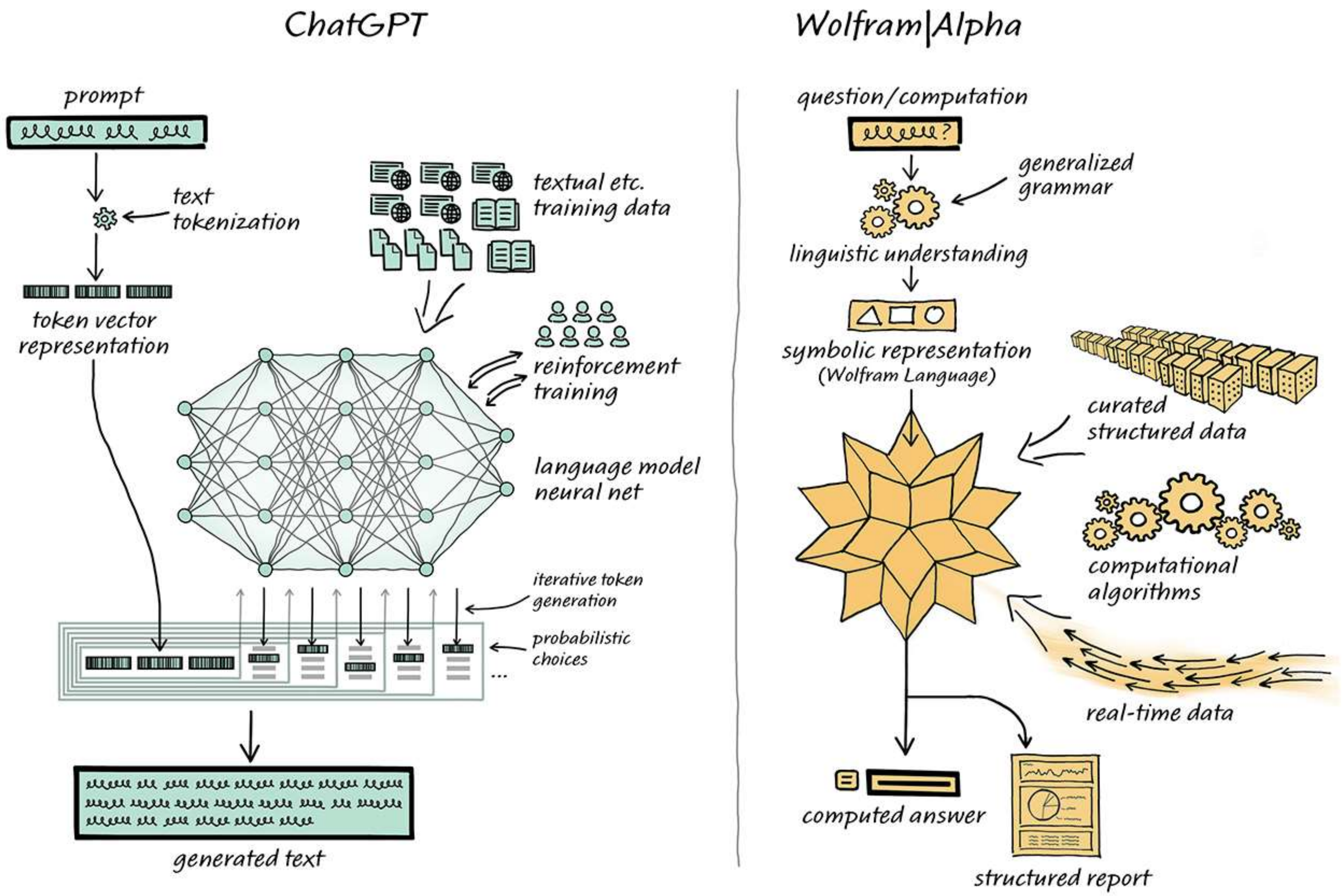
The Brown Act is California's law **[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their meetings open to the public.

- Toolformer: Ability to call out to alternate systems, including information retrieval or other machine learning models.
- Augmented Language Models (ALMs) more generally

- https://arxiv.org/pdf/2302.04761.pdf
- https://arxiv.org/pdf/2302.07842.pdf

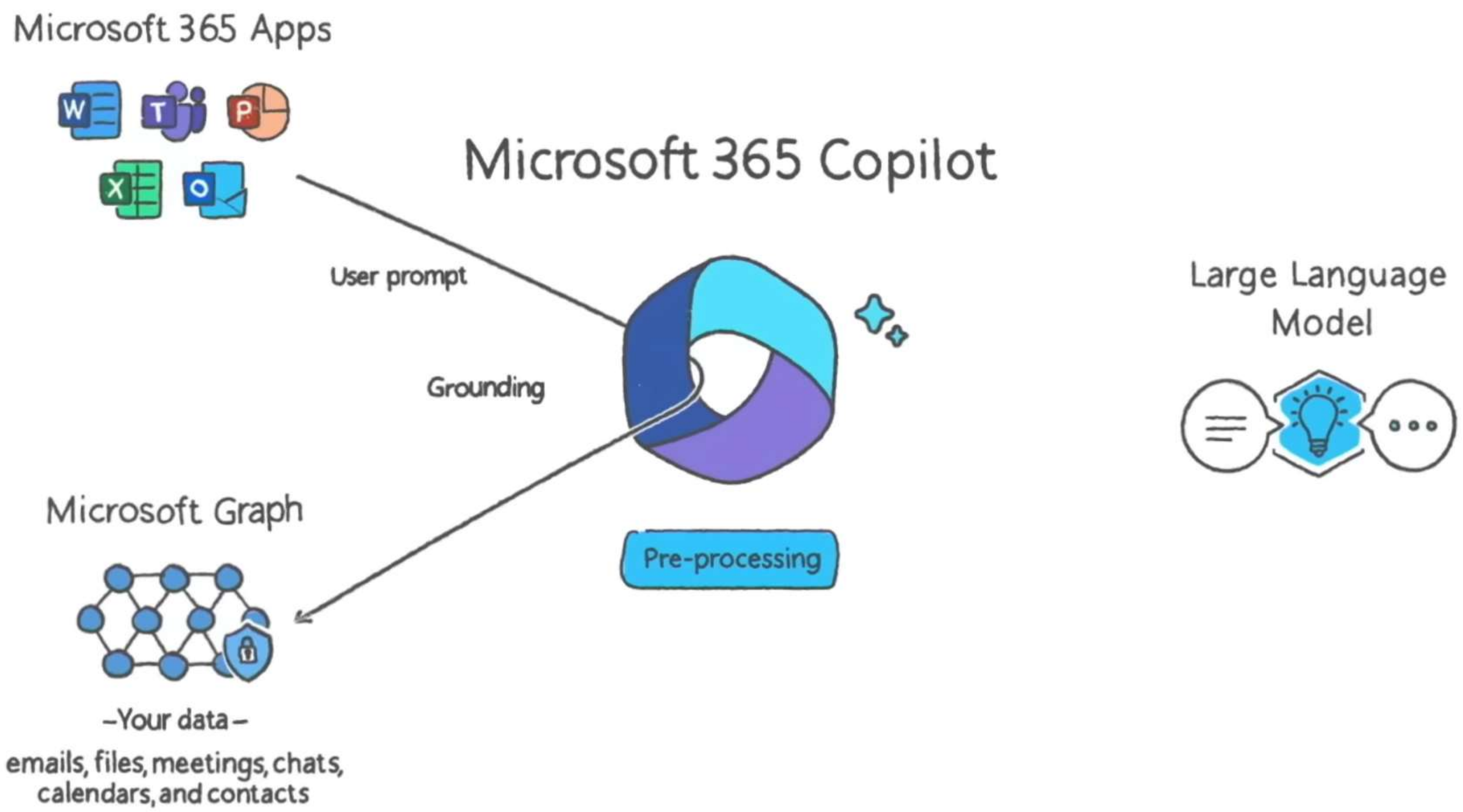
Wolfram|Alpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT

January 9, 2023



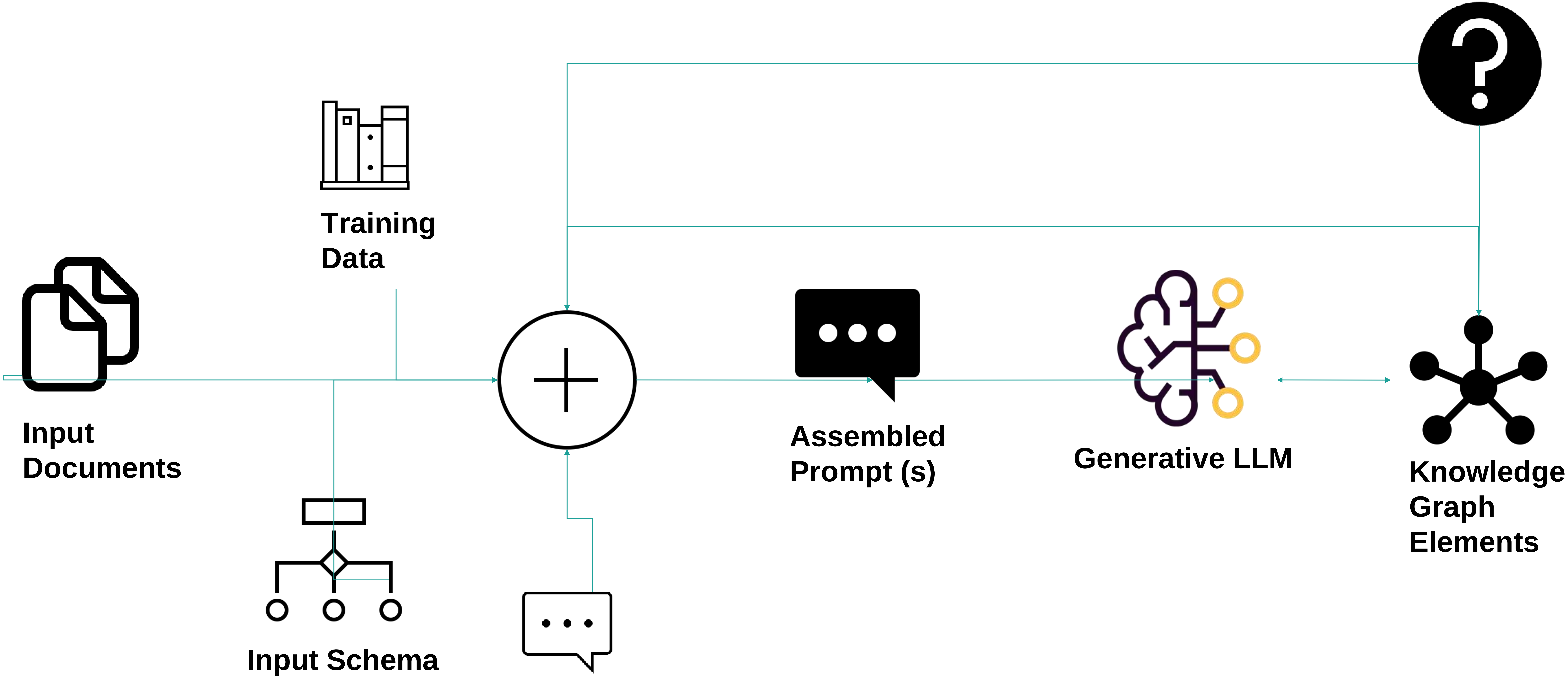
Source: https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/

New architectures



Source: The Future of Work With AI - Microsoft March 2023 Event
https://www.youtube.com/watch?v=Bf-dbS9CcRU&ab_channel=Microsoft

New architectures



Queries

Training Data

Input Documents

Input Schema

Prompt Templates

Assembled Prompt (s)

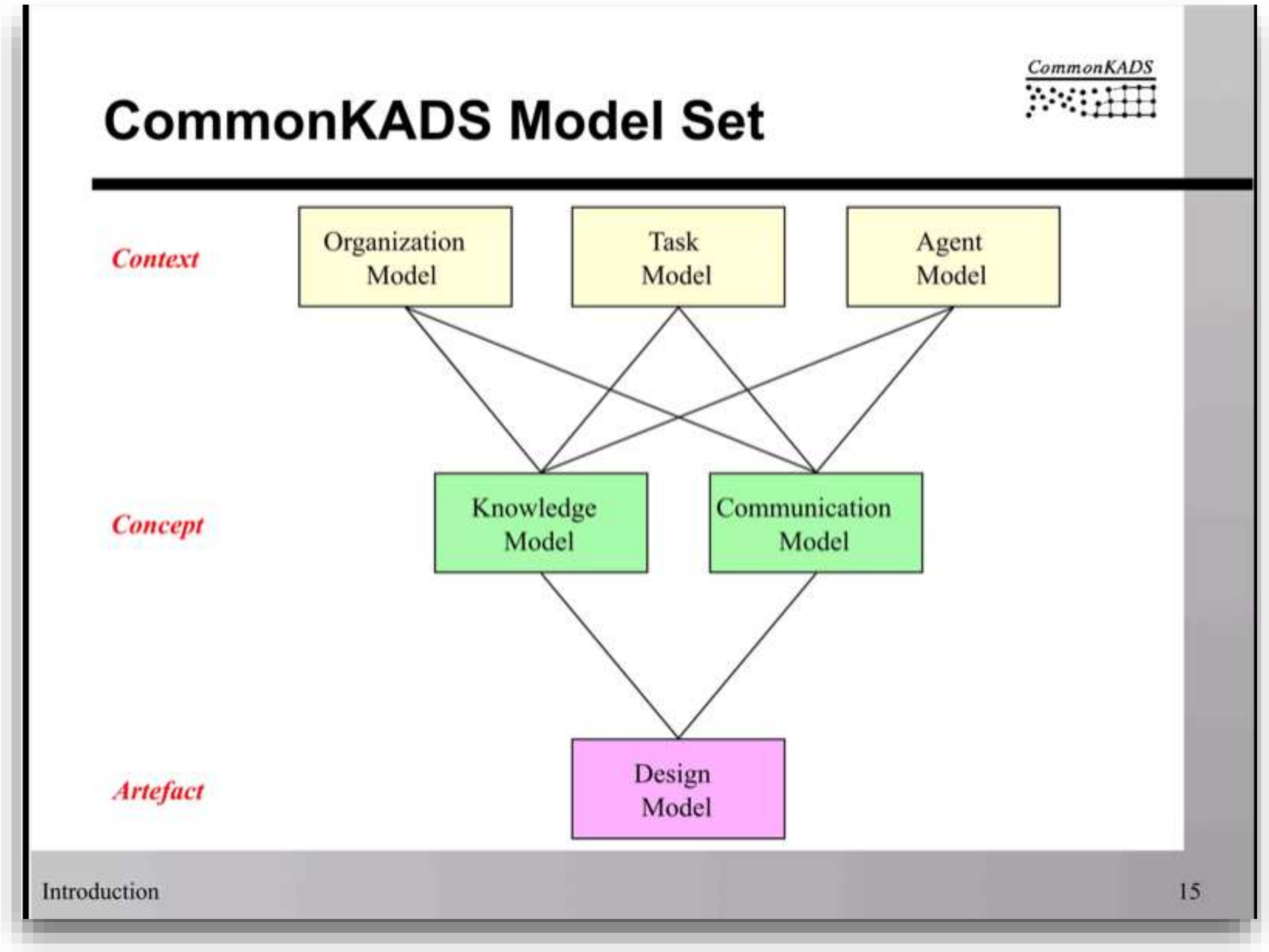
Generative LLM

Knowledge Graph Elements

enexa

The relevance of CommonKADS

- Context
  - Embedding in the organisation
  - What is the task? What is needed to get it done
  - Who does what and who has *authority* to act
- Concept
  - Communication: the wiring together of different prompts
  - Knowledge Model - *this is the hard one* - see next slide
- Artefact
  - You always need documentation (even if it's generated)



CommonKADS Model Set

Introduction 15

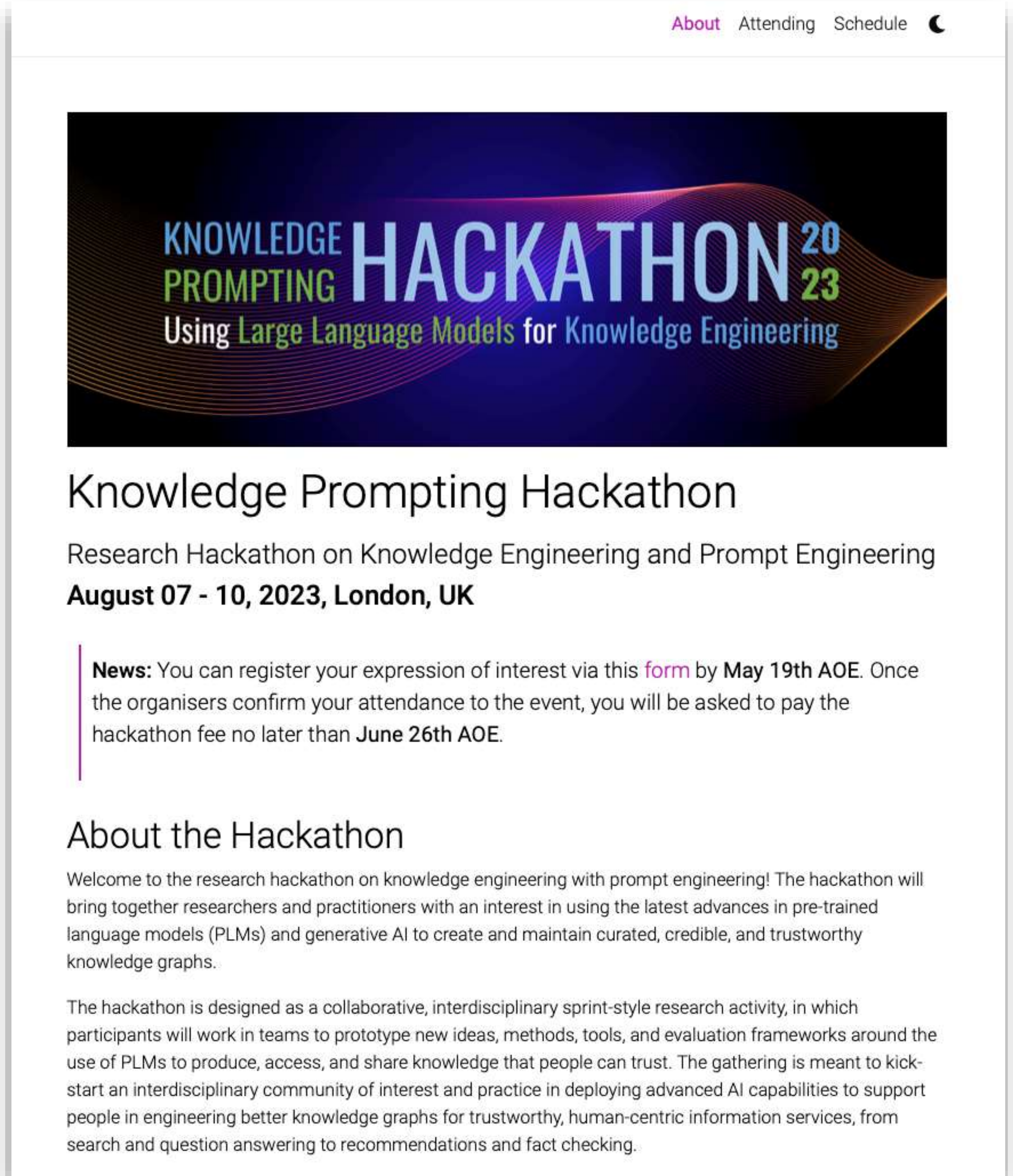
The Knowledge Model

- Classic discussion of knowledge models has been directed at
  - Symbolic representation
  - (formal) languages
  - creation of concept schemes automatically & semi-automatically

- Contention: LLMs actually allow us to work at the knowledge level using natural language
  - Express the concepts and definitions to drive applications we need as natural language
  - Definition of applications becomes one of agreement within an organisational context between people perhaps implicitly

Future

- Use KE methods to design LLM based apps
- A mapping between prompt templates and wiring to KE methodologies
- Identify the wiring diagrams
- Compare best practices in LM application design and KE methodologies
- ….


About
Attending
Schedule
KNOWLEDGE PROMPTING HACKATHON 2023
Using Large Language Models for Knowledge Engineering

Knowledge Prompting Hackathon

Research Hackathon on Knowledge Engineering and Prompt Engineering

August 07 - 10, 2023, London, UK

> **News:** You can register your expression of interest via this form by **May 19th AOE**. Once the organisers confirm your attendance to the event, you will be asked to pay the hackathon fee no later than **June 26th AOE**.

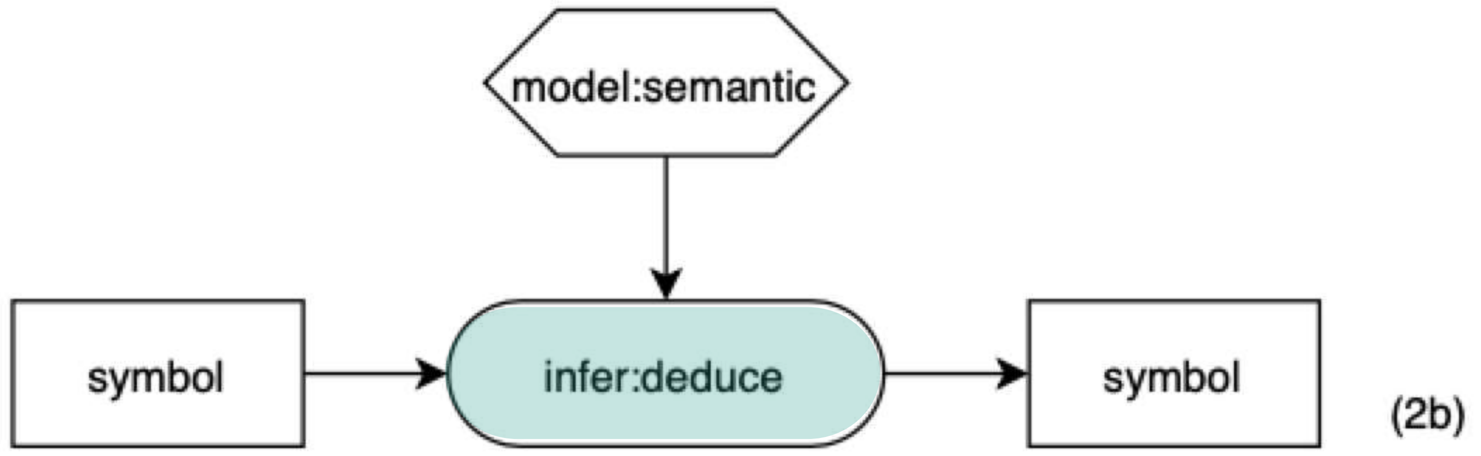
About the Hackathon

Welcome to the research hackathon on knowledge engineering with prompt engineering! The hackathon will bring together researchers and practitioners with an interest in using the latest advances in pre-trained language models (PLMs) and generative AI to create and maintain curated, credible, and trustworthy knowledge graphs.

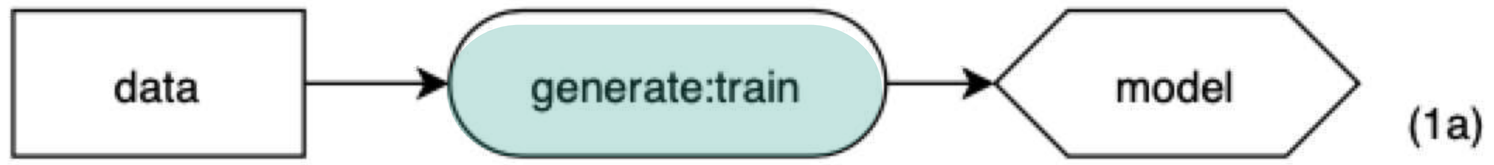
The hackathon is designed as a collaborative, interdisciplinary sprint-style research activity, in which participants will work in teams to prototype new ideas, methods, tools, and evaluation frameworks around the use of PLMs to produce, access, and share knowledge that people can trust. The gathering is meant to kick-start an interdisciplinary community of interest and practice in deploying advanced AI capabilities to support people in engineering better knowledge graphs for trustworthy, human-centric information services, from search and question answering to recommendations and fact checking.

More broadly the (Software) Engineering of AI applications

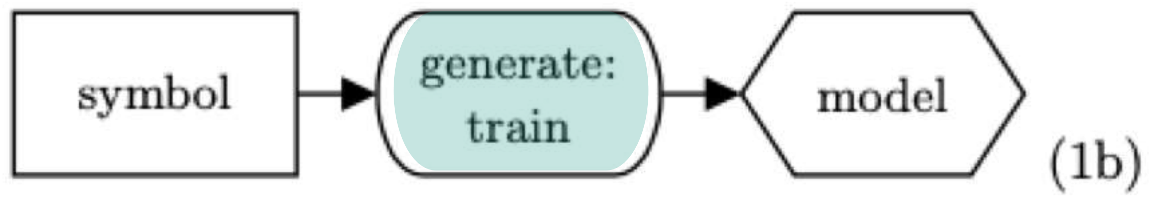
basic patterns



Symbolic Reasoning

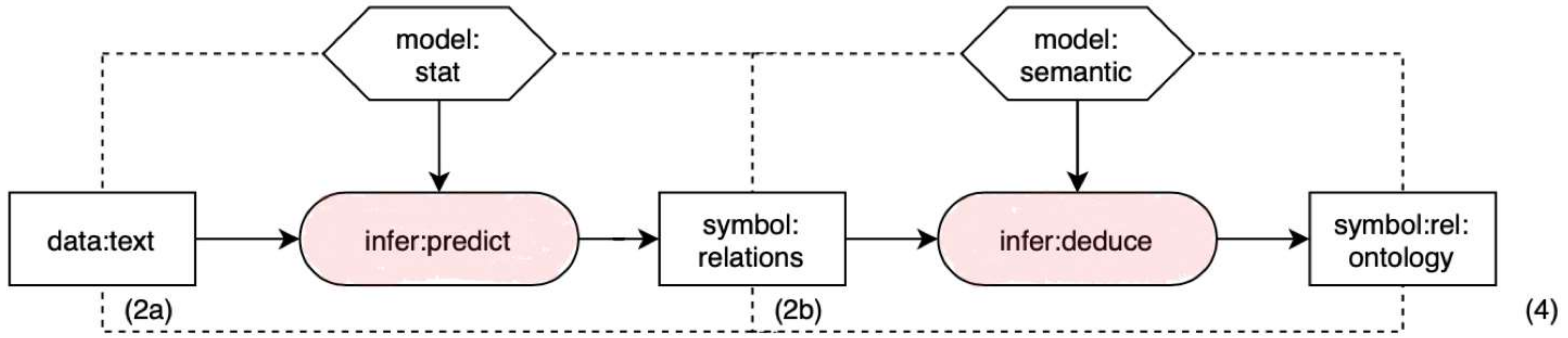


Machine Learning training

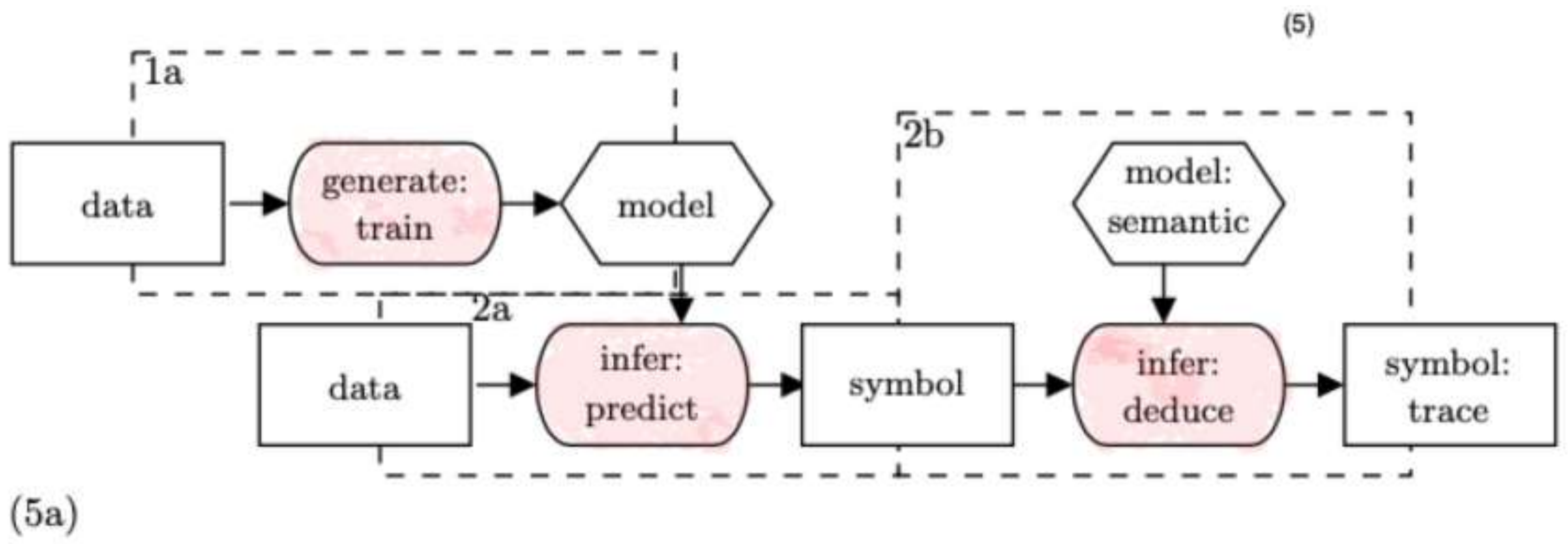


Learning models from symbols

Combination of basic patterns



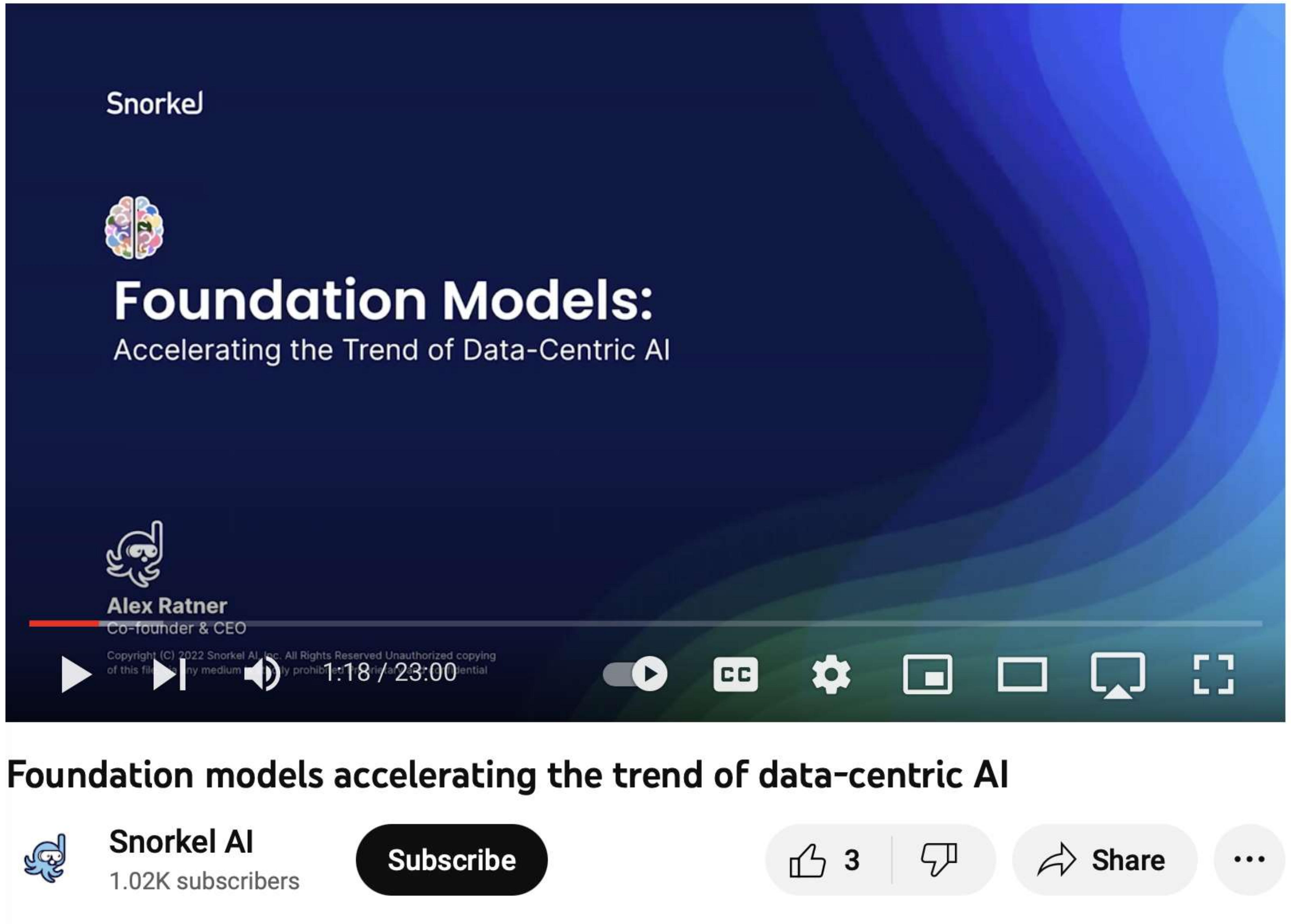
Ontology Learning



Explainable ML systems using inference

van Bekkum et al. (2021). *Modular design patterns for hybrid learning and reasoning systems*. Applied Intelligence, pp.1-19.
van Harmelen et al (2019). *A Boxology of Design Patterns for Hybrid Learning and Reasoning Systems*. Journal of Web Engineering (2019).

More broadly the (Software) Engineering of AI applications



Snorkel

Foundation Models:

Accelerating the Trend of Data-Centric AI

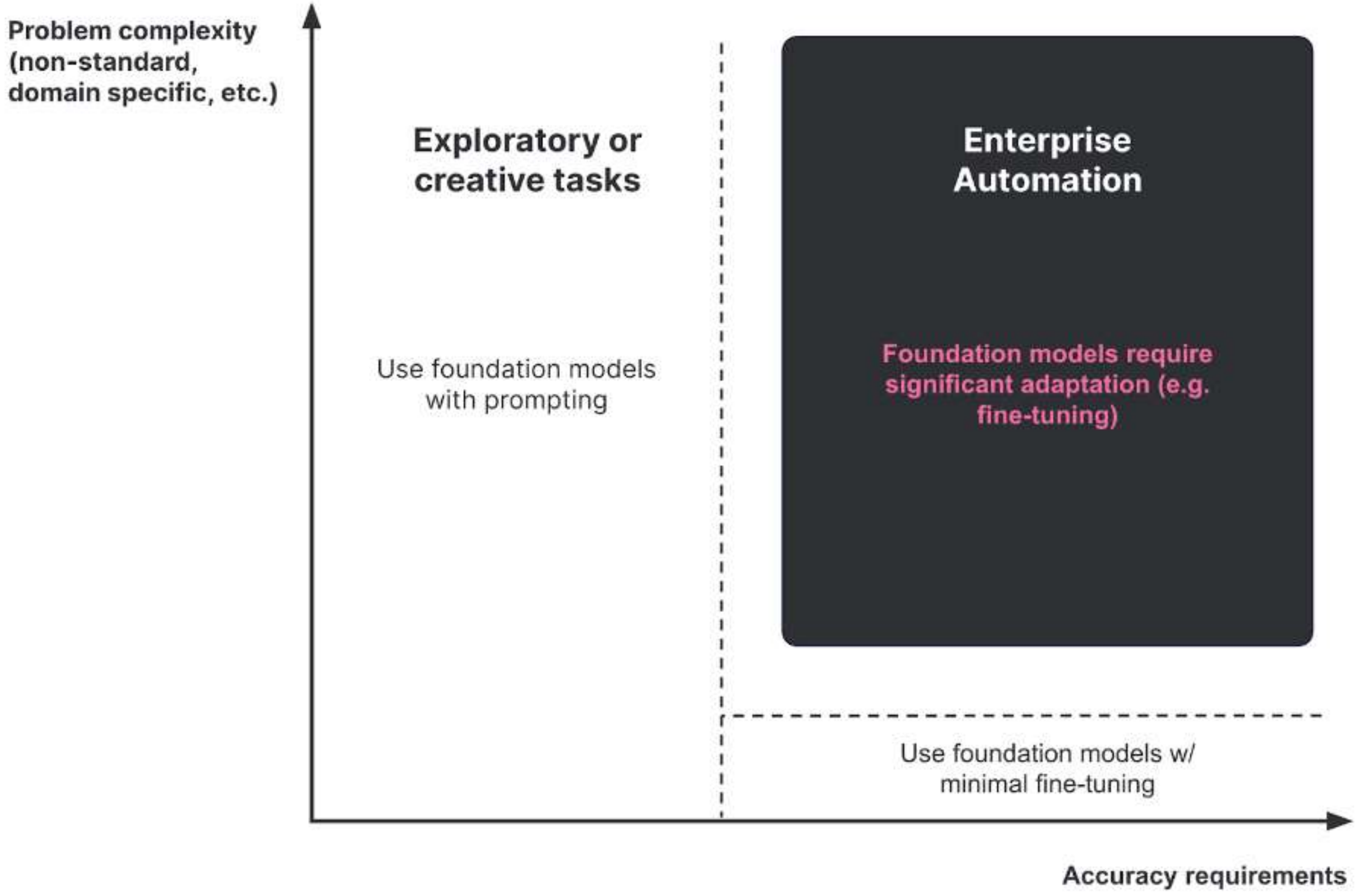
Alex Ratner

Co-founder & CEO

Foundation models accelerating the trend of data-centric AI

Snorkel AI

1.02K subscribers



Problem complexity (non-standard, domain specific, etc.)

Exploratory or creative tasks

Use foundation models with prompting

Enterprise Automation

Foundation models require significant adaptation (e.g. fine-tuning)

Use foundation models w/ minimal fine-tuning

Accuracy requirements

Source: https://snorkel.ai/data-centric-foundation-model-development-bridging-the-gap-between-foundation-models-and-enterprise-ai/

Alex Ratner

https://youtu.be/RFOEhGQhizQ

Conclusion

- With LLMs, application construction is about knowledge engineering
- KE methodologies provide a powerful structured basis for developing knowledge oriented applications
- Opportunity to update KE through the lens of prompt programming
- A brilliant chance to inform this dynamic area based on a well investigated scientific base

Paul Groth | @pgroth | pgroth.com | indelab.org

INDE lab
INtelligent Data Engineering

UNIVERSITY OF AMSTERDAM