Guest Lecture
Language Models for Ontology Alignment

Knowledge and Data

Yuan He

About Me

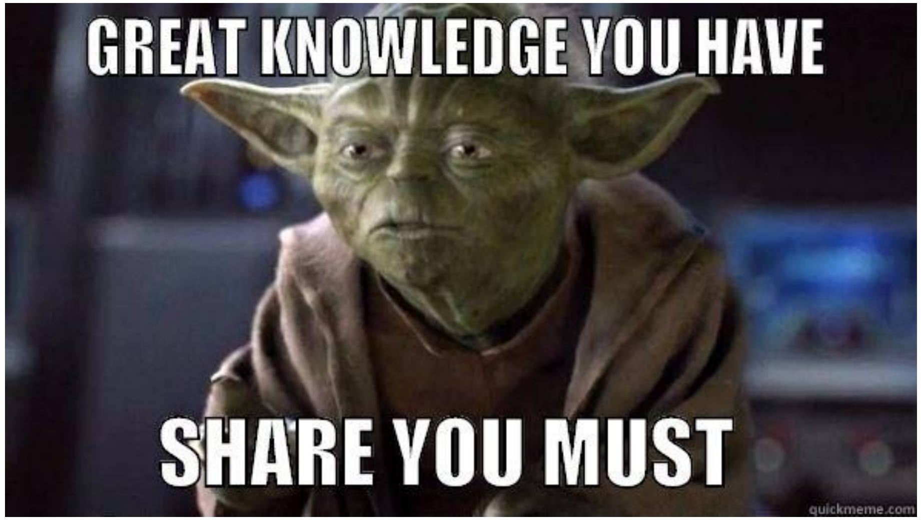
- **Applied Scientist at Amazon Rufus** 🐕
  - *LLMs for E-commerce*
- **Visiting Researcher at CAMEL-AI** 🐪
  - *Autonomous AI Agents*
- **Oxford PhD & Postdoc** 🦬
  - *LLMs for Knowledge Engineering*
- **Fun fact:** I seem to only work with animals

Outline

1. Ontology Alignment: Motivation & Challenges
2. Primer on Language Models
3. Methods for Ontology Alignment
4. Towards Agentic Workflow
5. Q&A / Discussion

Ontology Alignment

- We build knowledge — we must **reuse** and **share** it.
- **Ontology alignment** = linking entities across different ontologies so information can flow between them



Ontology Alignment

- More formally, ontology alignment means identifying **mappings between entities in different ontologies** to specify their relationships

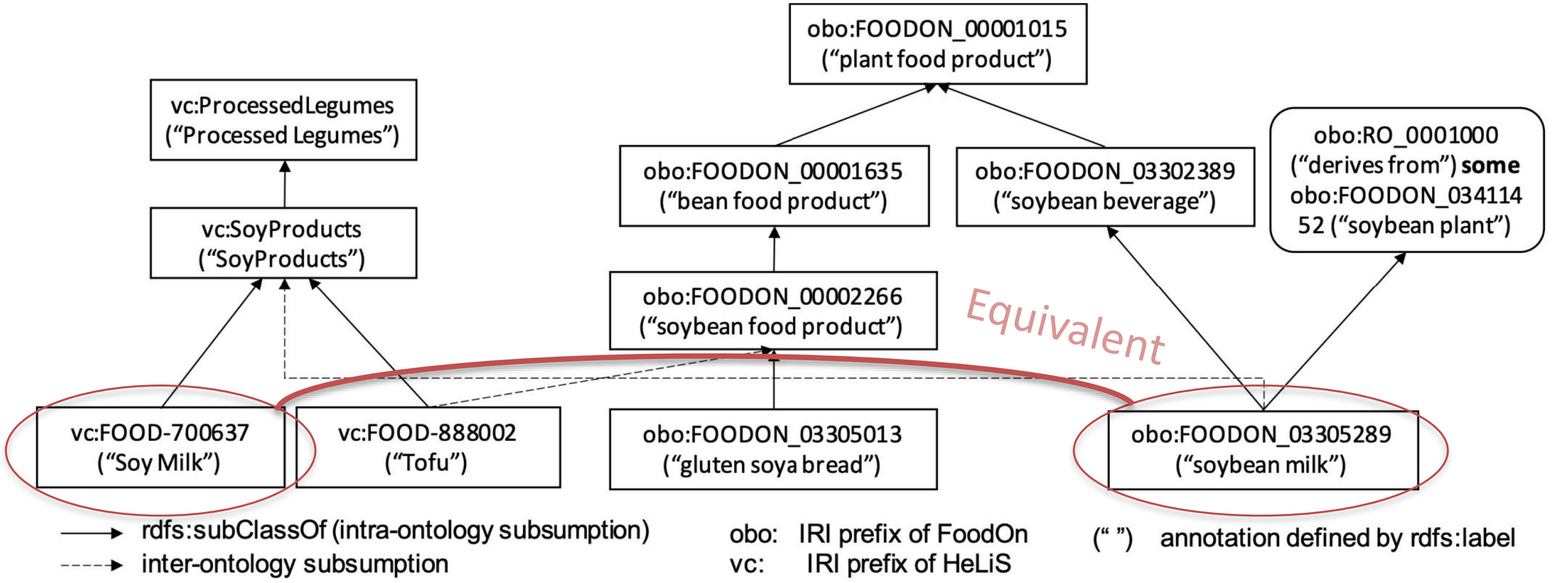


Fig. Example subsumption alignment between HeLis (Left) and FoodOn (Right)
[Chen et al., WWW 2023]

Ontology Alignment

- **Q**: How does alignment help knowledge sharing?
  - Example: $SoyMilk_{HeLis} \equiv SoybeanMilk_{FoodOn}$
  - In HeLis, there is no class named $SoybeanBeverage$
  - Through alignment, beverage information can be transferred from FoodOn to HeLis

- **Challenge**: Ontologies are often designed for different domain-specific purposes. As a result, taxonomy structures and entity naming can differ significantly.

Ontology Alignment

- Two prevalent relationships for alignment:
  - **Equivalence (≡)**: two entities are the same
  - **Subsumption (⊑)**: one entity is a subclass of another

- **Mapping**:
  - Triple form: $(entity_1, entity_2, relation)$
  - Quadruple form: $(entity_1, entity_2, relation, score)$ (probabilistic)
  - Example: $(SoyMilk, SoybeanMilk, \equiv, 0.82)$

- Beyond ≡ and ⊑: $partOf$, $derivedFrom$ …

Ontology Alignment

- **Q**: What if alignment leads to **inconsistency**❓
    - Axioms in the merged ontology **logically contradict** each other

- E.g., We have an equivalence mapping $SoyMilk \equiv SoyBeanMilk$ but …
    - One ontology: $SoyMilk \sqsubseteq DairyProduct$
    - Another ontology: $SoyBeanMilk \sqsubseteq \neg DairyProduct$
    - → **Contradiction** arises.

Ontology Alignment

- Two possibilities:
  1. Alignment is **wrong**
  2. Alignment **reveals hidden errors** in individual ontologies

- Solutions:
  - *Human resolution*: **Experts** consolidate conflicts
  - *Automated resolution*: **Repair algorithms** remove a **minimal** set of mappings to restore consistency

Ontology Alignment

- Ontology alignment is **labor-intensive** — yet reusing knowledge is essential
- We need **less manual effort** 🛠️ and **more autonomy** 🤖
- **Language models** offer a promising path towards automation

Outline

1. Ontology Alignment: Motivation & Challenges
2. Primer on Language Models
3. Methods for Ontology Alignment
4. Towards Agentic Workflow
5. Q&A / Discussion

Language Model

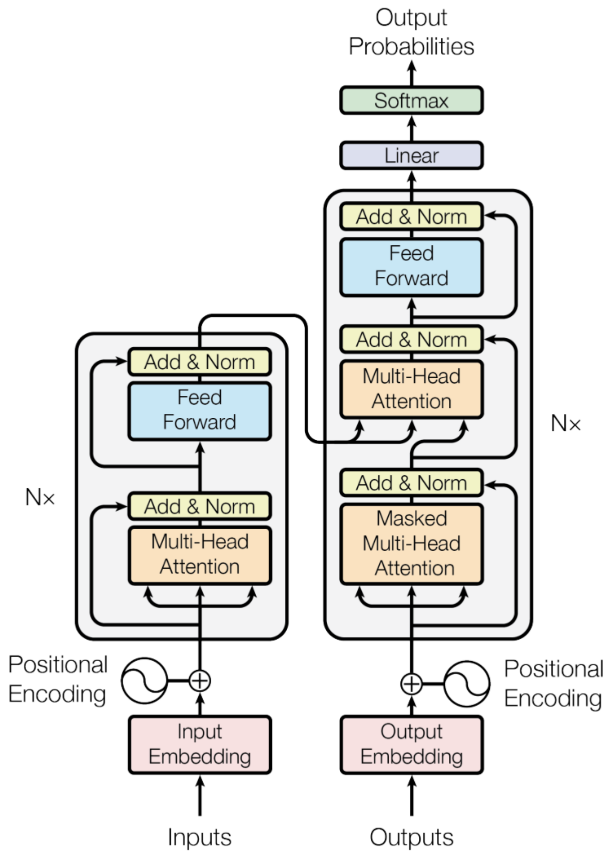
- Sequential Language Modeling: $P(w_i|w_{<i})$
  - *N-gram*: $P(w_i|w_{i-N<t<i})$
  - *RNN*: $P(w_i|w_{i-1}, h_{i-1})$
  - *GPT*: $P(w_i|\ \mathrm{Attention}(w_{<i}))$ ⟶ …

- Masked Language Modeling: $P(w_i|w_{<i}, w_{>i})$
  - *BERT*: $P(w_i|\ \mathrm{Attention}(w_{\backslash i}))$

Transformer

BERT

Encoder

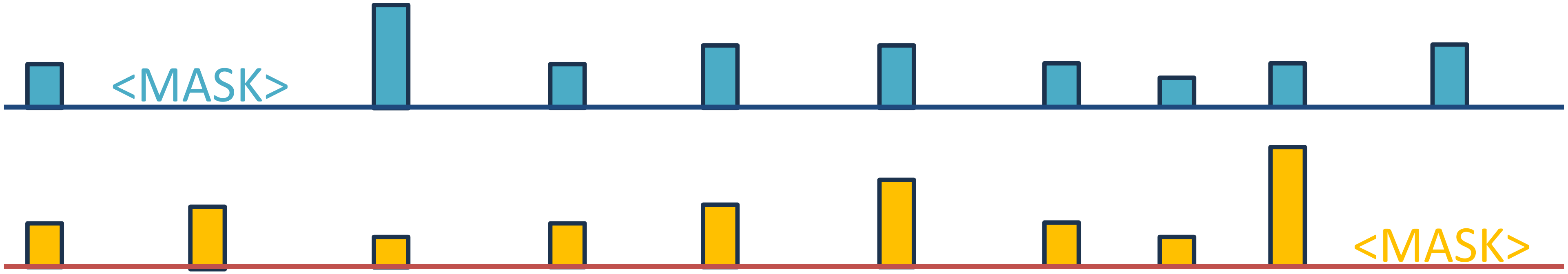


GPT

Decoder

Attention Mechanism

The bank robber was seen fishing on the river bank.



→ Attention enables LMs to capture **word meaning** from **context**

Attention Mechanism

- **Q**: How does this help ontology alignment ❓
  - LMs bring strong **language understanding**
  - They can align **linguistic variations** (e.g., $SoyMilk$ vs. $SoybeanMilk$)
  - **No need to consult dictionaries** (as in the traditional rule-based ontology alignment systems) — context does the work

Pre-training

- Train on **massive unlabeled corpora**
- **Masked** token prediction for **encoder-based** models, e.g., BERT
- **Next** token prediction for **decoder-based** models, e.g., GPT

The LLM pretraining mindset, distilled.

Fine-tuning

- Pre-trained LMs are **general-purpose**
- Fine-tuning adapts them to **specific tasks & domains**

- Strategies include:
  - **Full fine-tuning** → update all parameters
  - **Parameter-efficient tuning** → adapters, LoRA, etc.
  - **Prompt-based tuning** → reframe tasks into LM-friendly text

Fine-tuning (BERT)

- 💡 **Ex1. Add task-specific layer**
  - Input: $[CLS]\ text1\ [SEP]\ text2\ [SEP]$ ...
  - Use $[CLS]$ embedding for classification
  - Fast training ($\leq\ 5$ epochs)

- 💡 **Ex2. Embedding-based contrastive learning**
  - Tune embeddings directly (no extra layers)
  - Maximize $sim(e, e^{+})$, minimize $sim(e, e^{-})$
  - Example: synonyms closer, antonyms farther

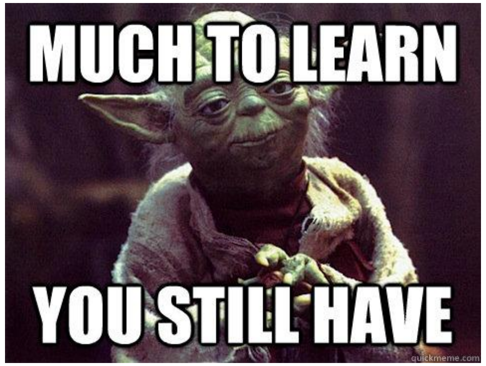
Fine-tuning (GPT)

- 💡 **Ex3. Instruction fine-tuning**
  - System prompts guide models: *“You’re a helpful assistant for some task…”*
  - Model follows the instruction and generates answers accordingly
  - ⚡ Modern LLMs can often handle this in a **zero-shot** setting

- 💡 **Ex4. RLVR**
  - Train LLMs to figure out a reasoning trace that leads to a verifiable reward (e.g., correct number in a math problem)

Many Concepts?

- You may have encountered a flood of terminology — pre-training, mid-training, post-training, instruction fine-tuning, RLHF, RLVR …



- In fact: When you try to figure out how these terms relate to each other, you're already **performing alignment** — connecting concepts across vocabularies.

Back to Alignment!

- **Q**: How does this help ontology alignment ❓
  - LMs gain **vast background knowledge** through pre-training
  - LMs can be **adapted to specific tasks** like ontology alignment through fine-tuning

- **Q**: But ontology alignment isn't really a pure text task, is it ❓
  - Right — which is why we need **tailored fine-tuning objectives** designed specifically for ontology alignment

Outline

1. Ontology Alignment: Motivation & Challenges
2. Primer on Language Models
3. Methods for Ontology Alignment
4. Towards Agentic Workflow
5. Q&A / Discussion

Common Components

- ✅ **Lexical matching** → “The starting point: matching entity names, synonyms, or descriptions.”
- ✅ **Structural matching** → “We can leverage ontology graph structure — parents, children, neighbors.”
- ✅ **Logical repair** → “After mappings, inconsistencies creep in. Repair mechanisms keep the merged ontology consistent.”

LogMap

- A well-known classical ontology aligner [Jiménez-Ruiz et al., 2011]
- Algorithm in a nutshell
  - 1️⃣ **Seed mappings** from *exact lexical matches*
  - 2️⃣ **Expand locally**: check if parents/children of aligned entities are also aligned (*locality principle*)
  - 3️⃣ **Consistency check**: reason over mappings, remove low-scored (through lexical matching) ones if inconsistent
  - 4️⃣ **Iterate** steps 2–3 until no further expansion is possible

LogMap

- ✅ Pros
  - **Fast and scalable**: graph expansion runs in linear time
  - **Consistency-aware**: minimizes logical inconsistency in alignments
- ⛔ Cons
  - Heavy reliance on **lexical heuristics** and external dictionaries
  - Limited to **equivalence matching** (cannot capture subsumption or complex relations)

BERTMap

- The first (arguably) language model-based aligner [He et al., 2022]

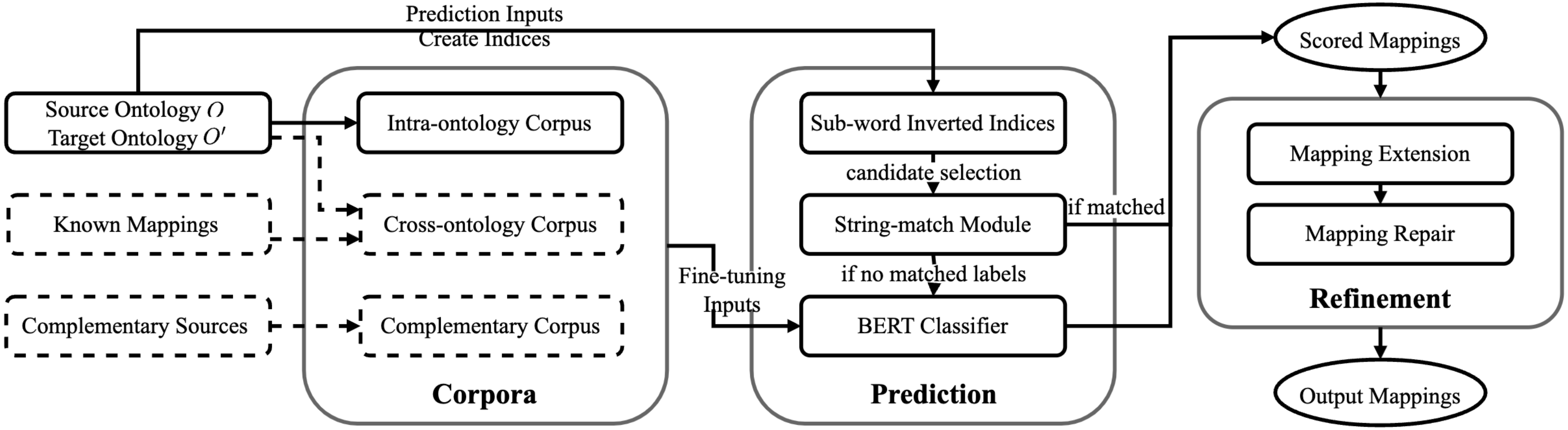


Fig. Illustration of the BERTMap system [He et al., AAAI 2022]

BERTMap

- 📖 **Corpus construction**: Builds rich **intra-ontology** and **cross-ontology corpora** for fine-tuning
- 🧩 **Fine-tuning BERT**: Uses ontology-specific corpora to **adapt BERT** for alignment tasks
- ⚡ **Efficient candidate selection**: Sub-word inverted indices narrow down possible matches
- 🔧 **Refinement module**: Mapping extension + repair to further improve precision and recall of final alignments

Using Text in Ontologies

- 🤔 As mentioned, ontology alignment isn't a pure text task …

- 🧐 But ontologies do contain text …
  - Entity names are usually defined by $rdfs{:}label$
  - Aliases / synonyms from other annotation properties

- 🙂 We can build a **domain-specific text corpus** (like a *thesaurus*) to support alignment

Using Text in Ontologies

- **Assumption 1 (Positive samples):**
  - Labels of the *same entity* are considered synonyms
  - Synonym pairs → **positive training examples**

- **Assumption 2 (Negative samples):**
  - Labels of *different entities* are non-synonymous
  - Non-synonym pairs → **negative training examples**

Using Text in Ontologies

- **Lexical matching as classification**:
  - Fine-tune a language model (e.g., BERT) to decide if two labels are synonyms.
- **Approach in BERTMap:**
  - Use the [CLS] token with a downstream classifier → outputs a binary synonym/non-synonym score.
- **Alternative approach (contrastive):**
  - Train embeddings so that synonym pairs are pulled closer and non-synonyms pushed apart.

Using Text in Ontologies

- **Advantage over rule-based** (e.g., LogMap)
  - Captures nuanced text semantics (beyond string matching)
  - No heuristics or external dictionaries needed

- **Advantage over other ML approaches**
  - **Self-supervised by default** → many ontologies naturally provide synonym and non-synonym pairs
  - No manual labels required; can still be extended with supervision or auxiliary data

More Advanced Methods?

- **Q:** Can we go beyond text classification and leverage more advanced LMs such as ChatGPT or Gemini for ontology alignment ❓

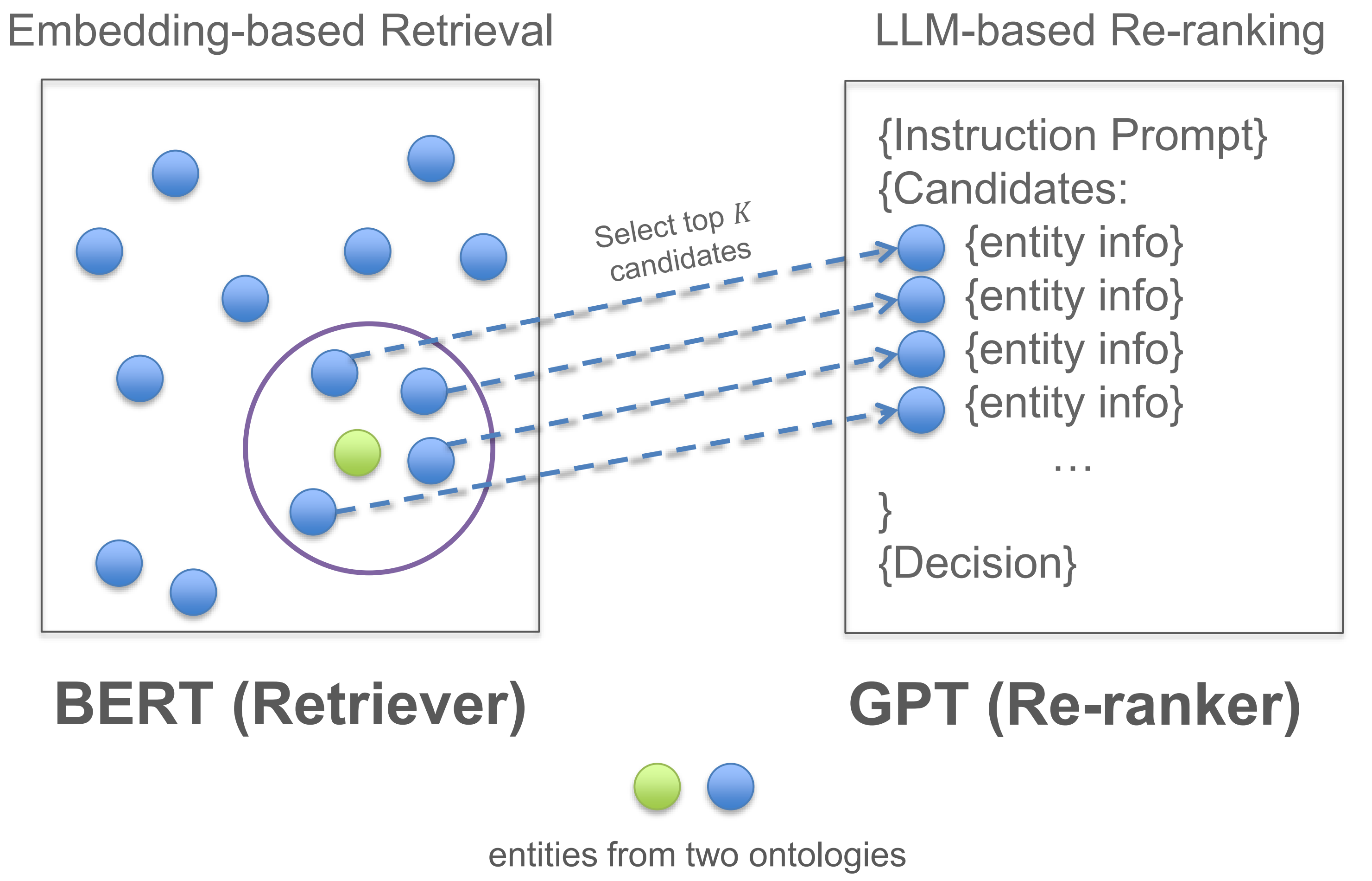
✅ $\boldsymbol{Yes}$, **but** …

- Naively comparing every entity pair is $O(N^2)$ — infeasible for large ontologies
- Thus, we need to **pair LLMs with an efficient retriever**
- Recall:
  - LogMap's local expansion → linear
  - BERTMap's sub-word index–based candidate selection → linear
  - ⚡ Advanced methods must also preserve efficiency

Retrieve-then-Rerank

- The **Retrieve-then-Rerank** paradigm: a hybrid two-stage approach
  - **Retriever** → efficiently narrows down candidate alignments
  - **Re-ranker** → reorders candidates using deeper, fine-grained reasoning
- **Key idea**: Efficiency and recall first, precision second.
- In practice:
  - BERT (or similar encoder models) = Retriever
  - ChatGPT / Gemini (decoder LMs) = Re-ranker

Retrieve-then-Rerank



Even More Autonomous?

- So far, we’ve looked at ontology alignment through structured pipelines …

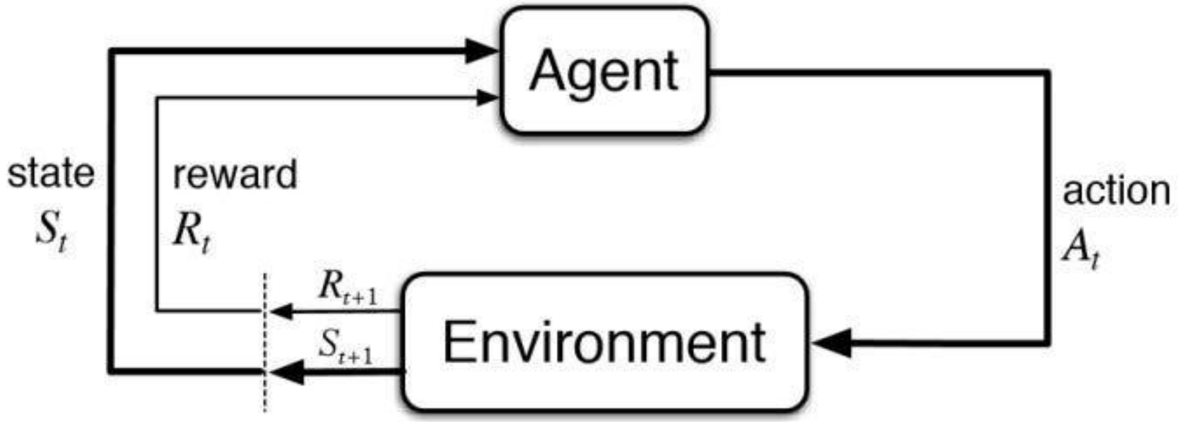
- But can we push further towards **autonomy**?
  - → What if an LLM could *understand the task itself*, and make alignment decisions dynamically — without relying on a rigid, pre-defined pipeline?

Outline

1. Ontology Alignment: Motivation & Challenges
2. Primer on Language Models
3. Methods for Ontology Alignment
4. Towards Agentic Workflow
5. Q&A / Discussion

LLMs as Agents

- With sufficient in-context learning and reasoning capabilities, an LLM can serve as an **agent** that **acts** according to its **state** in the **environment**



Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction," 2018

Agentic Ontology Alignment

- **Q:** What capabilities or tools do LLM agents need to perform ontology alignment ❓

- **A**: Equip the agent with tools to **access** and **reason** over ontology data, for example:
  - **SPARQL Engine** → query ontology databases
  - **Semantic Retriever** → fetch candidate entities
  - **Ontology Reasoner** → check logical consistency
  - *(… and other domain-specific tools)*

Agentic Ontology Alignment

👤 User: Please align two ontologies … [Attachment]: ontology_1, ontology_2

🤖 Assistant: Okay, first I will make a plan …

🤖 Assistant: Using SPARQL engine to query entity information …

🤖 Assistant: Using Semantic Retriever to find candidate matches …

🤖 Assistant: Detected inconsistency → applying repair strategy …

Thanks!

1. Ontology Alignment: Motivation & Challenges
2. Primer on Language Models
3. Methods for Ontology Alignment
4. Towards Agentic Workflow
5. Q&A / Discussion

Homepage: https://www.yuanhe.wiki/ | **LinkedIn**: www.linkedin.com/in/lawhy | **X**: @lawhy_X