Stanford CS224v Course

Conversational Virtual Assistants with Deep Learning

Lecture 9

Agentic Approach: Knowledge Base Queries

Monica Lam & Shicheng Liu & Sina Semnani

Can We Handle Real Knowledge Bases?

Study of 2 Hard Real-Life Datasets

The Agentic Approach

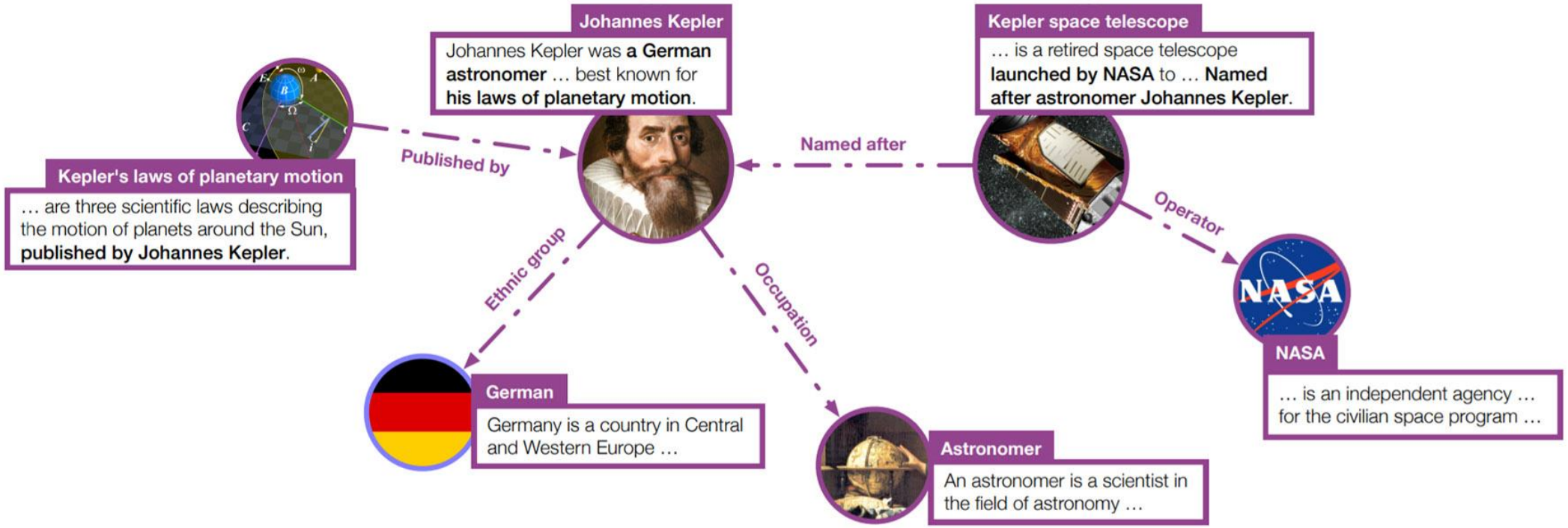
Lecture Goals

Teach the Agentic Approach so You Can Use It in Your Projects!

- Knowledge Graphs – aka Knowledge Bases (KB)
  - Why KBs?
  - Why is KBQA (KB Question Answering) Challenging?
    -- Previous Work
  - The Agentic Approach for KBs
  - How to Get a Good KBQA Dataset?
  - Evaluation
- Applying the Agentic Approach to SQL Databases

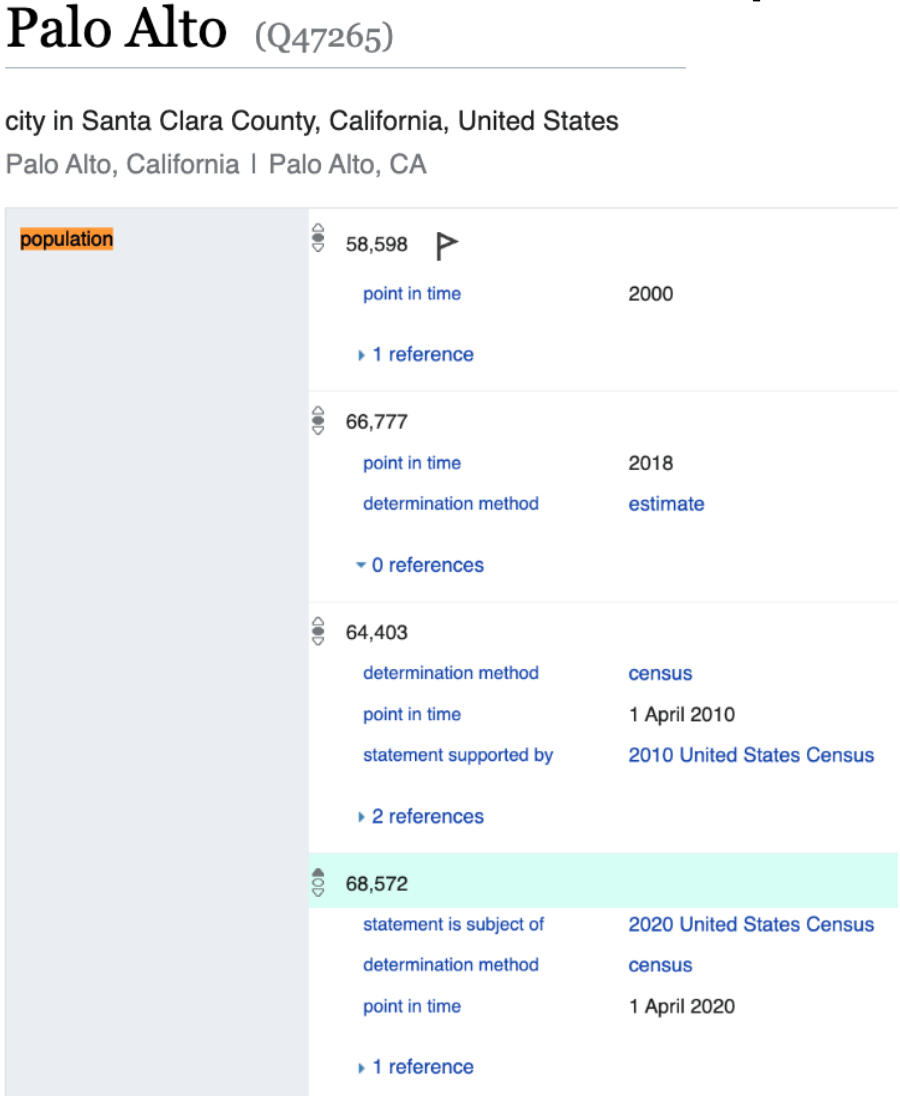
A Knowledge Graph

- A lot of information **cannot** be represented in tables
- Knowledge graph is also known as a semantic web
  - Nodes are entities
  - Edges are relationships



Wikidata: The Largest Live Knowledge Graph

- Stats:
  - 15B facts (1B more triples per year)
  - 100M entities
  - 10K properties
  - 25K contributors
- All entities in Wikipedia are in Wikidata
- Wikidata contains many more entities



Wikidata Representation – a RDF graph

- Representation: RDF triples
  - Nodes are entities (represented by unique QIDs)
    - There are many entities with exactly the same name
    - Same ID across all Wikipedia in all languages
  - Edges are properties (represented by unique PIDs)

Query with SPARQL

Who founded Stanford?

SELECT ?x WHERE

{ wd:Q41506 wdt:P112 ?x. }

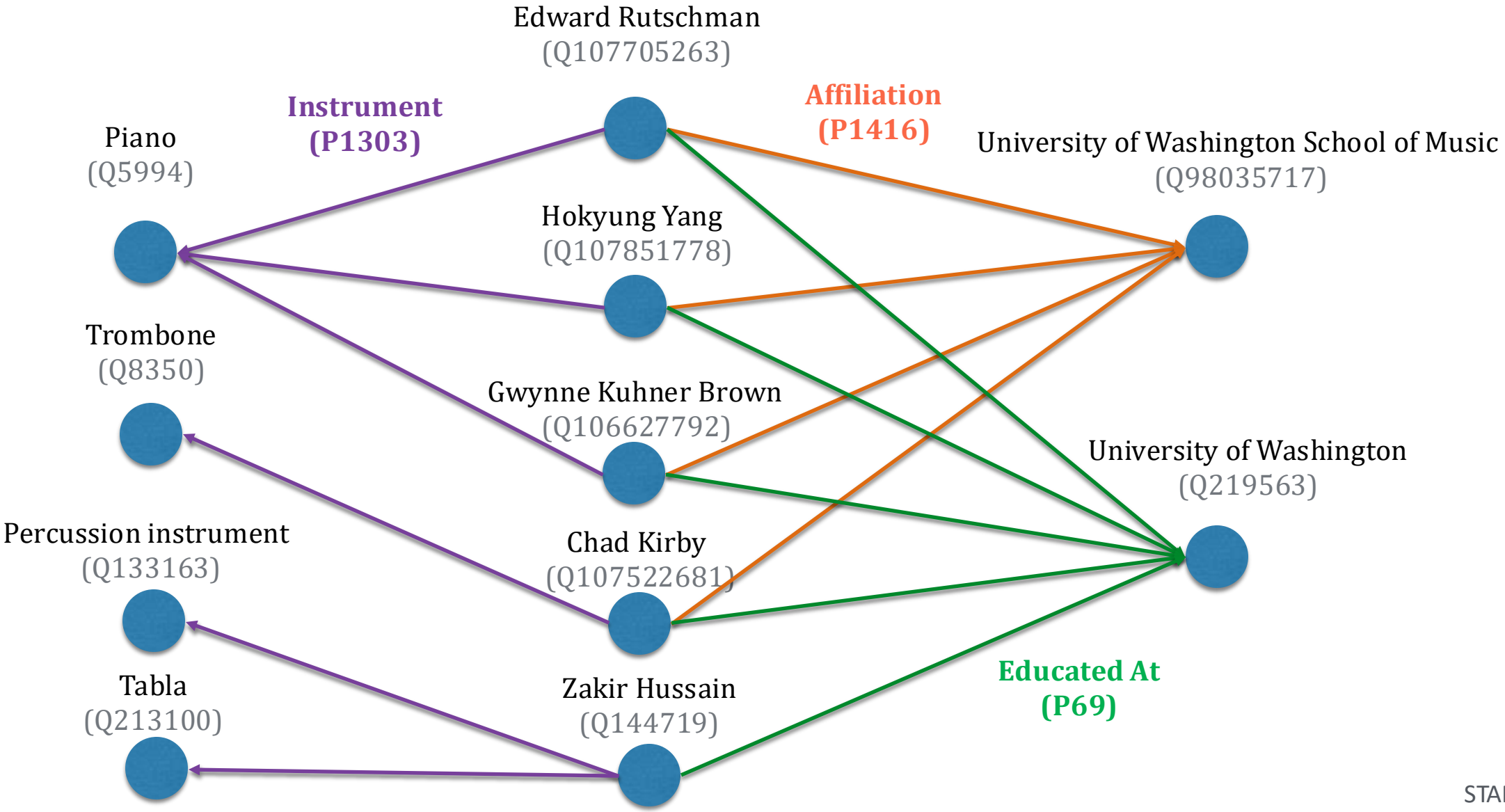
Stanford Founded by

A natural language interface can greatly expand access!c

Running Example: Music Instruments Played

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

A Subset of the Knowledge Graph in Wikidata



```
>> execute_sparql("""
SELECT ?instrument ?instrumentLabel (COUNT(?student) AS ?count) WHERE {
  ?student wdt:P1303 ?instrument;
    wdt:P1416 wd:Q98035717;
    wdt:P69 wd:Q219563.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
GROUP BY ?instrument ?instrumentLabel
""")
Observation:
   | instrument    | instrumentLabel  | count |
   |---------------|------------------|-------|
   | Q5994         | piano            | 99    |
   | Q1467960      | mbira            | 2     |
   | Q8350         | trombone         | 11    |
   | Q8338         | trumpet          | 8     |
   | Q17172850     | voice            | 32    |
   | ...           | ...              | ...   |
   | Q302497       | mandolin         | 1     |
   | Q187851       | recorder         | 1     |
   | Q185041       | cor anglais      | 1     |
   | Q83509        | piccolo          | 1     |

>> stop()
```

Writing the correct SPARQL query after correcting the mistake

P1303: instrument
P1416: affiliation
P69: educated at

Q98035717:
University of Washington
School of Music

Q219563:
University of Washington

The Power of WikiData

- SPARQL allows relational algebra operations across the entire knowledge graph
  - Running example needs: filters, projections, joins, counts …
  - Query optimizations
- Dataset for research in **Mathematics**, **Biology**, **Education**, **Social Sciences**, **Linguistics**, ..

Quiz: Can we represent the data as tables?

Lecture Goals

Teach the Agentic Approach so You Can Use It in Your Projects!

- Knowledge Graphs – aka Knowledge Bases (KB)
  - Why KBs?
  - **Why is KBQA (KB Question Answering) Challenging?**
    -- Previous Work
  - The Agentic Approach for KBs
  - How to Get a Good KBQA Dataset?
  - Evaluation
- Applying the Agentic Approach to SQL Databases

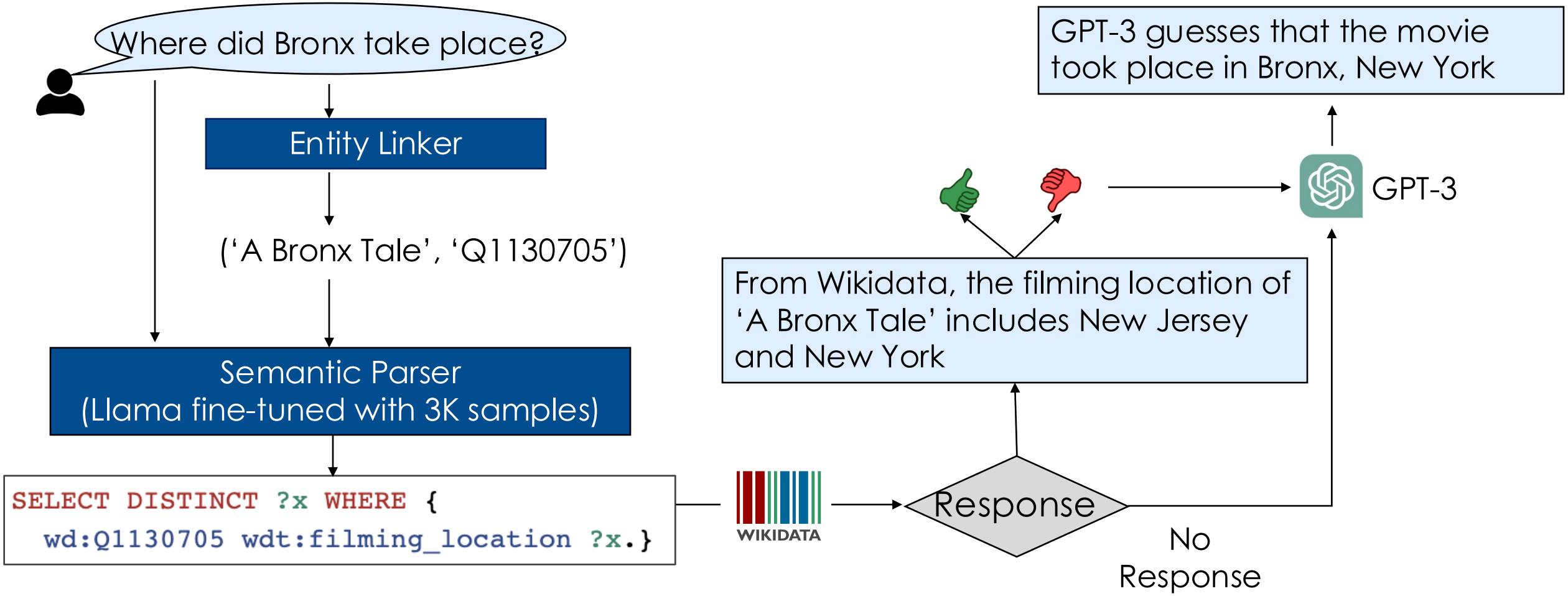
Previous Work

Knowledge Base Query Answering (KBQA)

SEMANTIC PARSER

SUBGRAPH TRAVERSAL

1. Semantic Parsing: NL → SPARQL



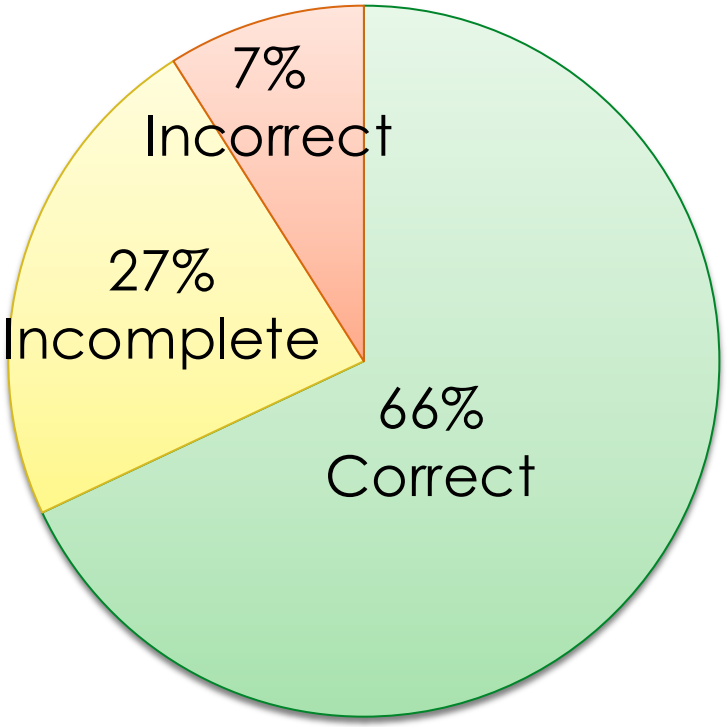
Silei Xu et al. Fine-tuned LLMs Know More, Hallucinate Less with Few-Shot Sequence-to-Sequence Semantic Parsing over Wikidata.. EMNLP 2023

New WikiWebQuestions Dataset

- Adapted from WebQuestions for FreeBase
  - Questions from Google Suggest API
  - Real-world popular questions
  - 2431 train, 438 dev, 1384 test
- GPT-3: trained on Wikipedia + Internet
  - Can GPT-3 answer WikiQebQuestions?

Silei Xu et al. Fine-tuned LLMs Know More, Hallucinate Less with Few-Shot Sequence-to-Sequence Semantic Parsing over Wikidata.. EMNLP 2023

GPT-3 on New Wiki-WebQuestions Dataset



GPT-3
Guesses

Question: "What does Obama have a degree in?"
GPT-3: "Political science degree"
Missing: "Law degree"

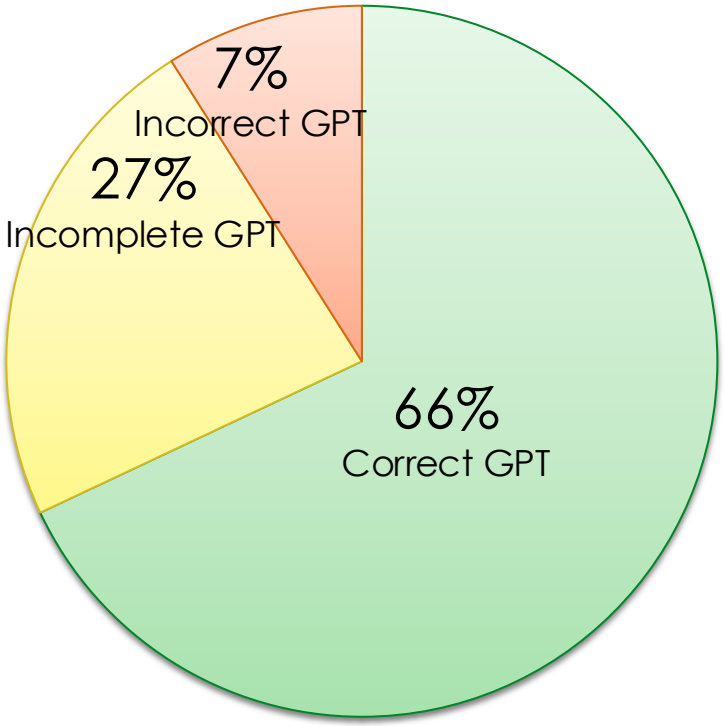
Question: "where does the name Melbourne come from?"
GPT-3: "Melbourne comes from the Latin word 'melburnum' meaning 'blackburn' or 'blackbird' "
Answer: "Melbourne is named after William Lamb, 2nd Viscount Melbourne"

Users must fact check all the answers!

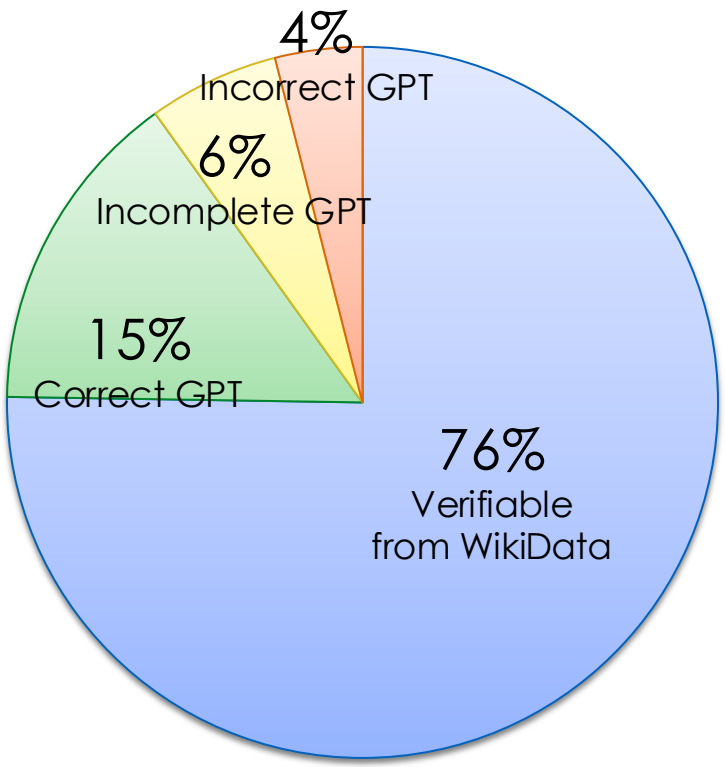
Silei Xu et al. Fine-tuned LLMs Know More, Hallucinate Less with Few-Shot Sequence-to-Sequence Semantic Parsing over Wikidata.. EMNLP 2023

Reduce GPT3 Hallucination with WikiData

Benchmark: WikiWebQuestions (Popular, Human-Generated)



Just GPT-3 Guesses



Wikidata (Genie semantic parser) + GPT3 Guesses

Weakness

- WikiWebQuestions are relatively simple, popular questions using mostly the same properties
- Wikidata has about 3000 knowledge-oriented properties

Dataset	#Unique Properties	Dataset Size
MCWQ	27	124187
QALD 9	158	507
QALD 10	177	394
RUBQ	251	2910
WikiWebQuestions	189	4316
→ SPINACH	298	323

Our test set for in-context learning

Why Querying Wikidata with SPARQL is Difficult

- **Many properties:** Hard to memorize all the properties
- **No fixed schema:** Wikidata does not have a fixed schema
  - We do not know what properties each node has
- **Many similar properties:**
  - Often unclear which property or entity should be used
  - Questions on locations: "Where did Isaac Newton live?" "Where is Salesforce?"
  - Possible properties – depends on what is available for the node
    - *administrative territorial entity*
    - *residence*, *state*, *country*, *place of birth*, *place of death*
    - *headquarters location*, *location of formation*
- **Need to look at a node's properties to determine the right SPARQL**

2. Subgraph Retrieval

Retrieve a part of the graph based on a question

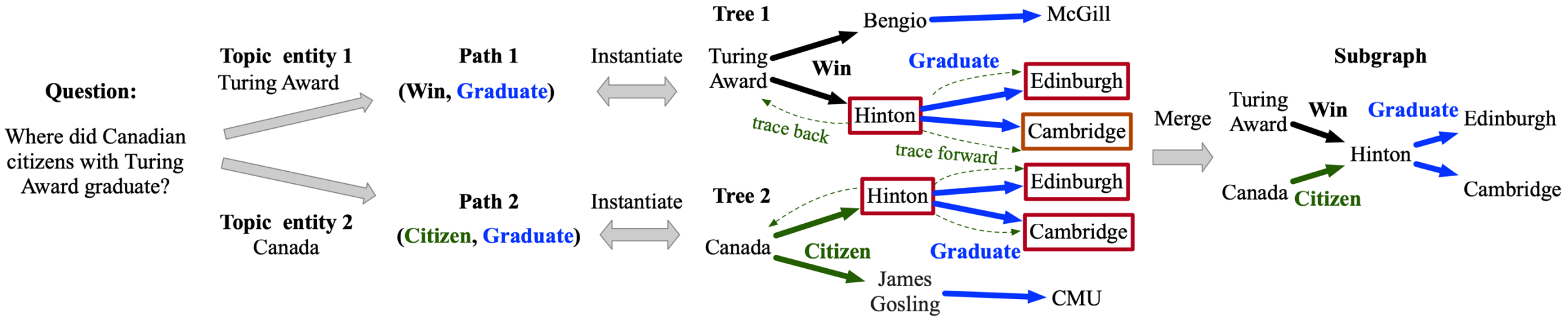


Figure 2: Illustration of the subgraph retrieving process. We expand a path from each topic entity as well as induce a corresponding tree, and then merge the trees from different topic entities to form a unified subgraph.

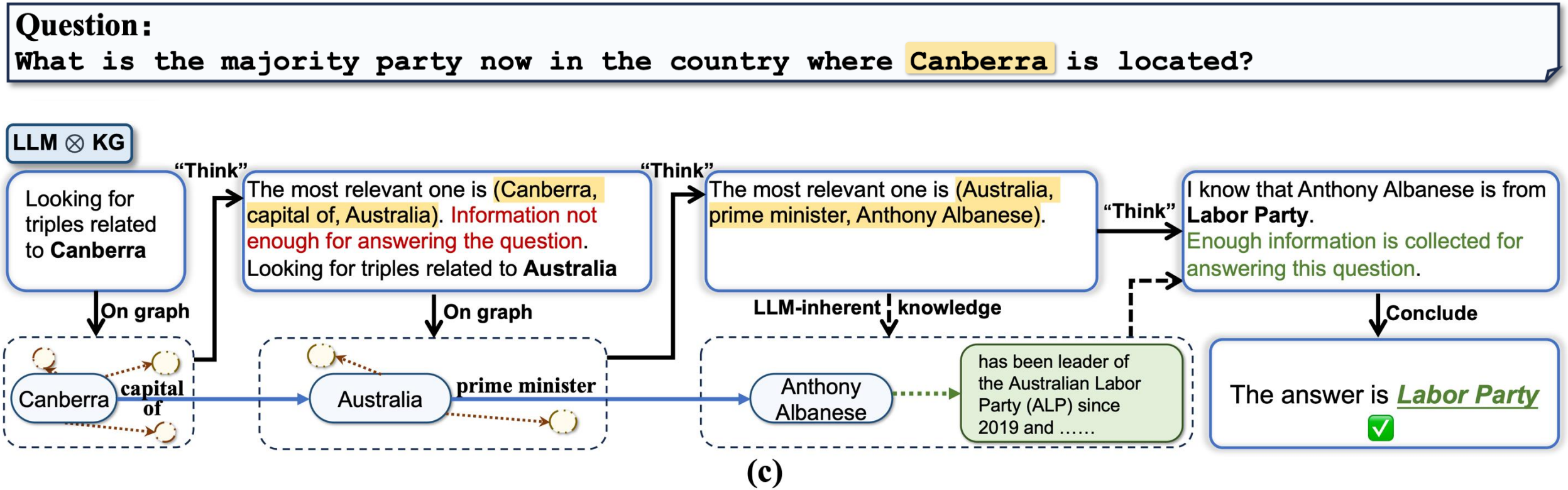
Quiz: Can this handle finding the tallest mountain?

Quiz: Can this handle our running example?

Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering (Zhang et al., ACL 2022)

3. LLM-Based Graph Exploration

Explore a sub-graph by walking the graph one edge at a time



Quiz: Is "walking the graph" enough?
Can it handle our running example?

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph, Sun et al, ICLR 2024

Lecture Goals

Teach the Agentic Approach so You Can Use It in Your Projects!

- Knowledge Graphs – aka Knowledge Bases (KB)
  - Why KBs?
  - Why is KBQA (KB Question Answering) Challenging?
    -- Previous Work
  - **The Agentic Approach for KBs**
  - How to Get a Good KBQA Dataset?
  - Evaluation
- Applying the Agentic Approach to SQL Databases

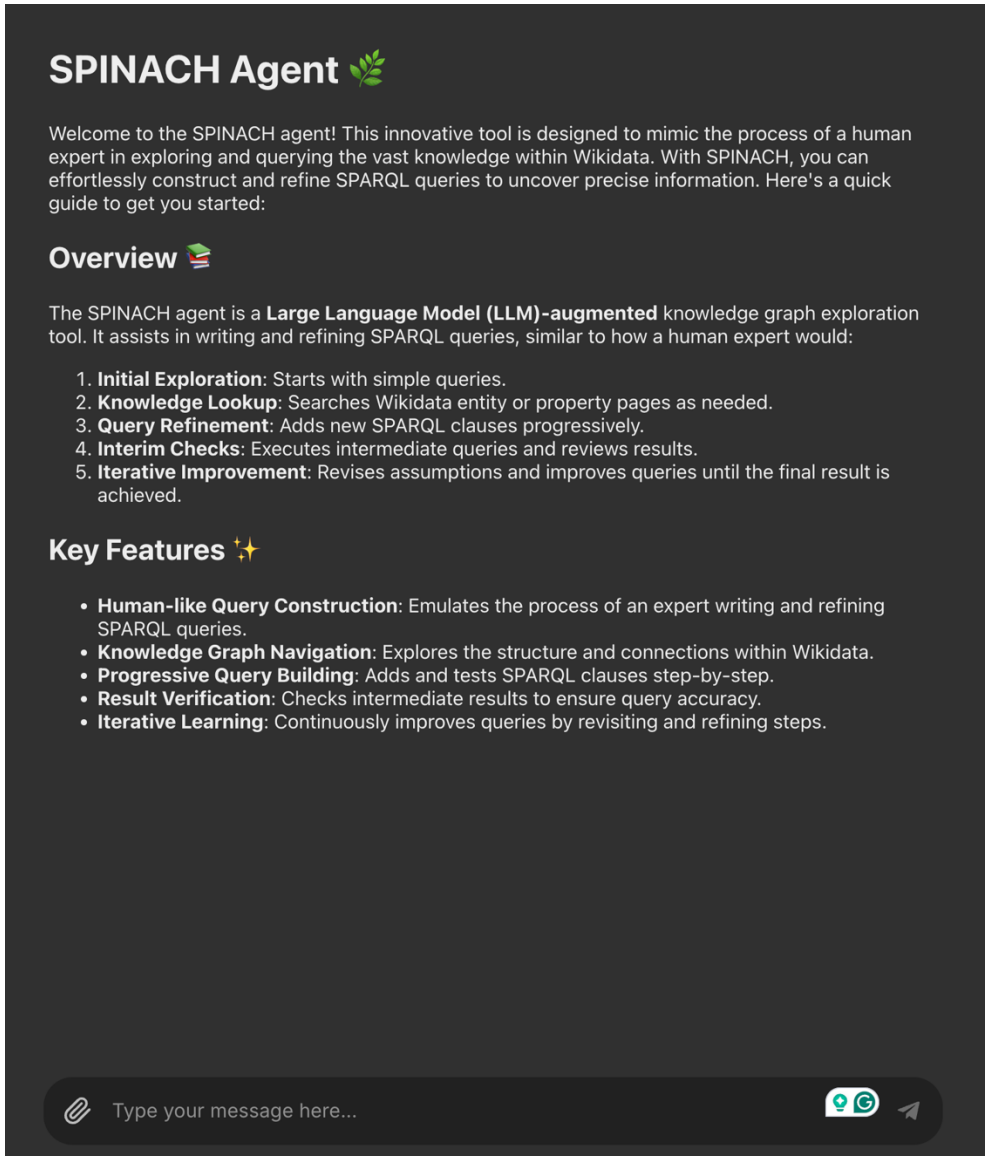
SPINACH:
SPARQL-Based Information Navigation for Challenging Real-World Questions

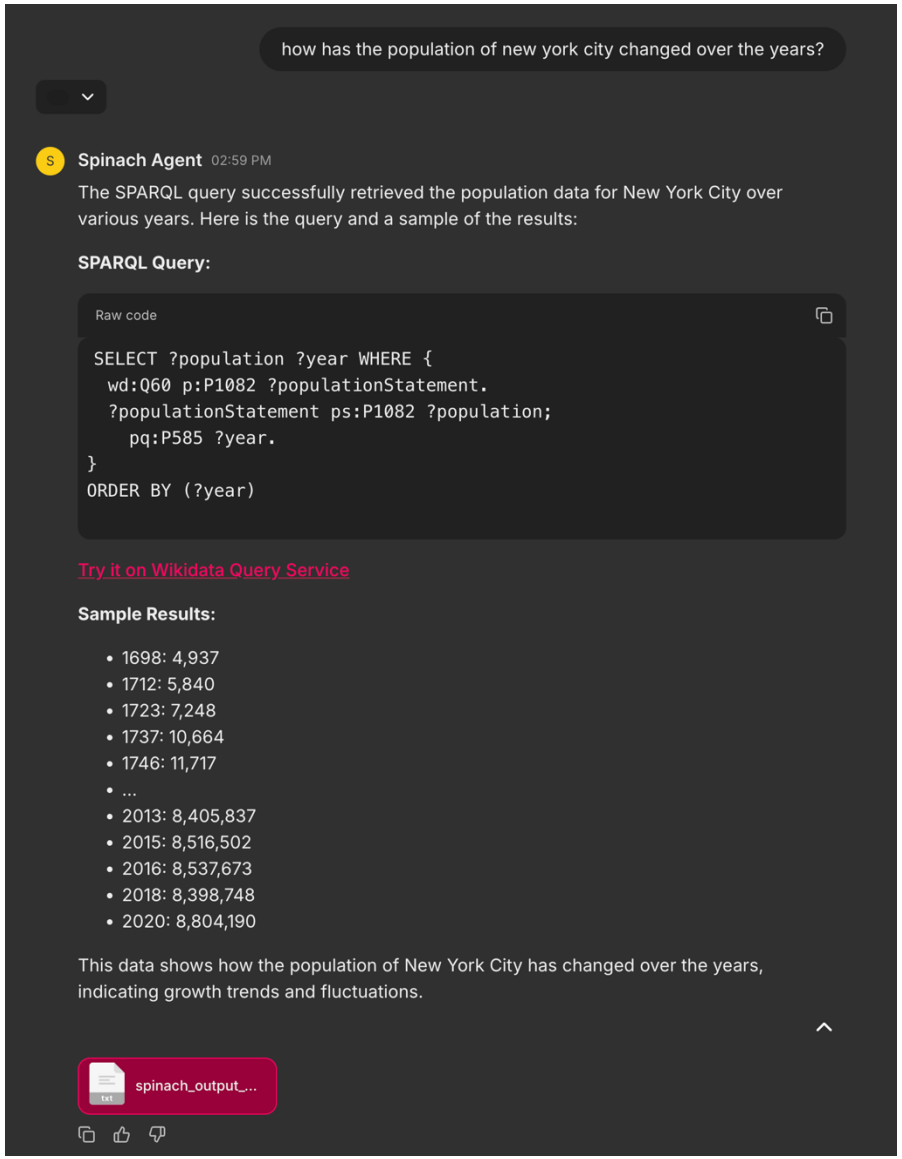
Shicheng Liu* Sina J. Semnani*
Harold Triedman[1] Jialiang Xu Isaac Dan Zhao Monica S. Lam
Stanford University

* Equal contribution
[1] Cornell Tech; Work conducted while at the Wikimedia Foundation

EMNLP 2024

Https://spinach.genie.stanford.edu





Combine the Best of SPARQL, Subgraph Retrieval, and LLMs

Key idea: combine

- SPARQL query:
  - Expressiveness, implementation/query optimizations
- Graph retrieval: exam actual properties in the data
- LLM:
  - If human experts can do it: we can do it!
  - Automate the human expert's approach
  - Humans don't memorize the nodes and properties

  → we don't need fine-tuning! Just use ICL

How a Human Expert Write a SPARQL Query

1. Start by writing **simple SPARQL queries**;
2. **Look up** Wikidata **entity or property pages** when needed
   - To understand the structure of the knowledge graph
   - Check what properties exist for a node
3. **Add new SPARQL clauses** to build towards the final SPARQL

Weaves together

- Knowledge inquiry
- Query writing
- Execution and evaluation of results (subgraph retrieval)

Actions Useful for Writing SPARQL

Provided by Wikidata

search_wikidata(string): http://wikidata.org (search bar):
text → returns QIDs and PIDs

get_wikidata_entry(QID): https://www.wikidata.org/wiki/QID:<QID>
QID → Wikidata page for entity

get_property_examples(PID): https://www.wikidata.org/wiki/Property:<PID>:
PID → examples of how property PID is used

execute_sparql(SPARQL): https://query.wikidata.org:
SPARQL → result

Running Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

Running Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

```
search_wikidata("musical instrument")
```

```
- musical instrument (Q34379)
  device created or adapted to make musical sounds
- heraldic musical instrument (Q56877088)
  category of heraldic charges
- Musical instrument (Q102413357)
  Oil of canvas by Alla Grigoryan
- Musical Instrument (Q102413736)
  painting by Musical instrument
```

This step performs Named Entity Disambiguation (NED) with LLM!

Running Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

```
search_wikidata("affiliation")
```

affiliation (P1416)

organization that a person or organization is affiliated with (not necessarily member of or employed by)

27 statements, 0 sitelinks - 18:16, 28 October 2024

affiliation string (P6424)

qualifier to provide the published string form of affiliation attached to an author

14 statements, 0 sitelinks - 16:31, 1 October 2024

Running Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

get_wikidata_entry
("Q98035717")

University of Washington School of Music (Q98035717)

```
{
  instance of (P31): {
    faculty (Q180958): {
      Qualifiers: [
        {
          of (P642): University of Washington (Q219563)
        }
      ]
    },
    music school (Q1021290): {}
  },
  field of work (P101): [
    higher education (Q136822),
    postgraduate education (Q141212),
    musicology (Q164204),
    music (Q638),
    research (Q42240),
    ethnomusicology (Q208365)
  ],
  located in the administrative territorial entity (P131): Seattle (Q5083),
  location (P276): Music Building (Q98690890),
  has subsidiary (P355): University of Washington Laboratory for Music
      Cognition, Culture & Learning (Q101157561),
  part of (P361): University of Washington College of Arts and Sciences
      (Q7896566),
  has part(s) (P527): University of Washington Laboratory for Music
      Cognition, Culture & Learning (Q101157561),
  parent organization (P749): University of Washington College of Arts and
      Sciences (Q7896566)
}
```

Running Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

```
get_property_examples("P1416")
```

```
Affiliation (P1416)

Raoul Bott -> Institute for Advanced Study
Hannah Sipe -> University of Washington School
of Environmental and Forest Sciences
```

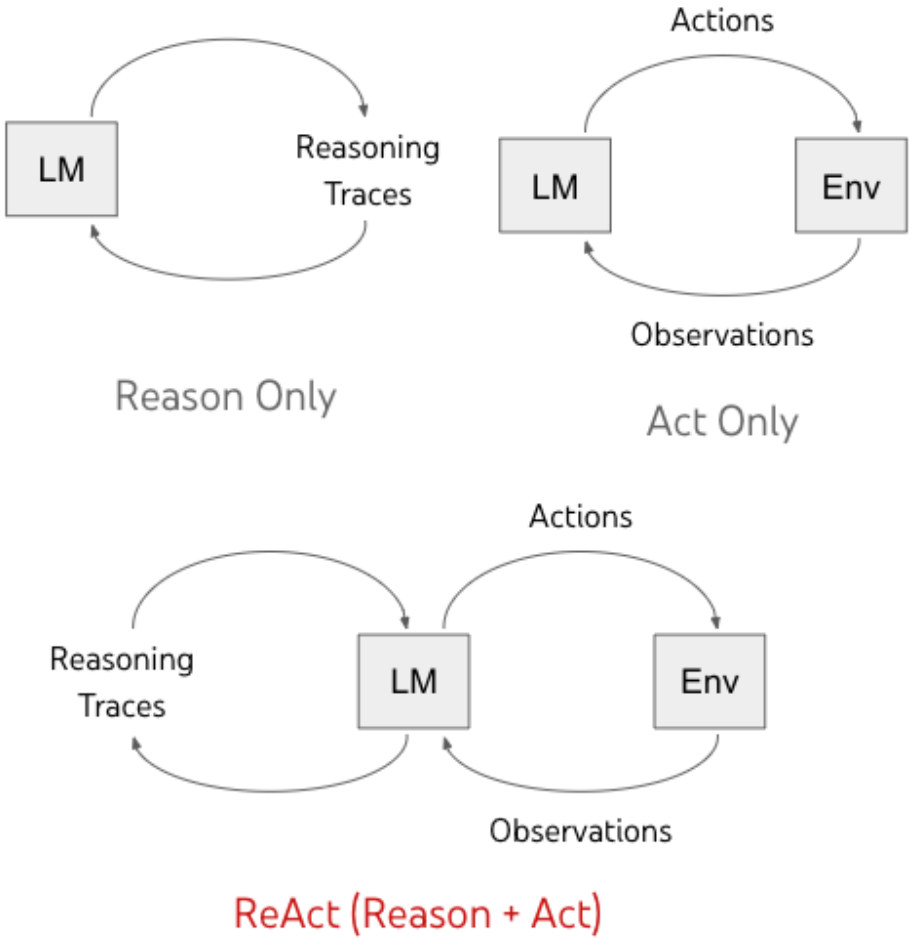
Running Example

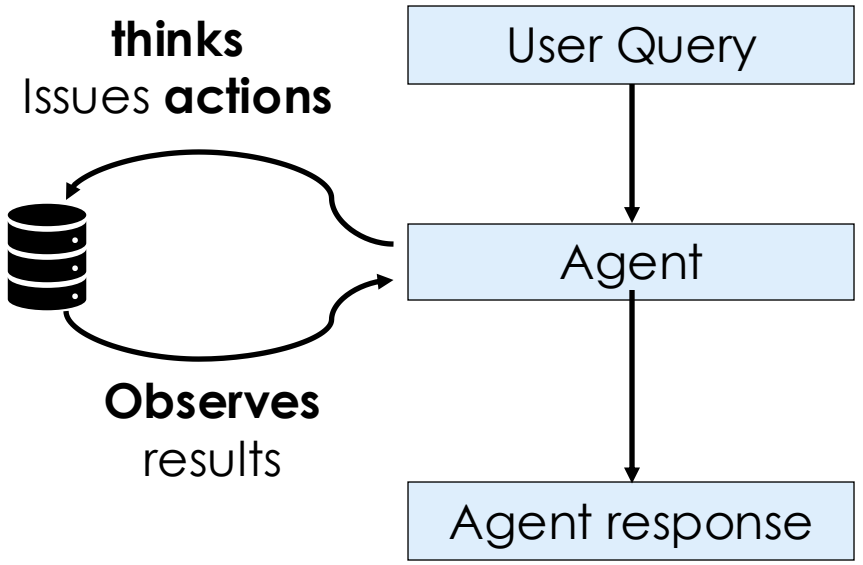
Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

```
execute_sparql("""
SELECT ?instrument ?instrumentLabel (COUNT(?student) AS ?count) WHERE {
  ?student wdt:P1303 ?instrument;
    wdt:P1416 wd:Q98035717;
    wdt:P69 wd:Q219563.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
GROUP BY ?instrument ?instrumentLabel
""")
```

instrument	instrumentLabel	count
Q5994	piano	99
Q1467960	mbira	2
Q8350	trombone	11
Q8338	trumpet	8
Q17172850	voice	32
...	...	...
Q302497	mandolin	1
Q187851	recorder	1
Q185041	cor anglais	1
Q83509	piccolo	1

Agentic Approach for Knowledge Bases





ReAct: Synergizing Reasoning and Acting in Language Models, Yao et al, ICLR 2023

The SPINACH Agent

- Imitates what the user does with an agentic approach
- Uses the full expressiveness of SPARQL for exploration

- For N steps:
  - Given the history of agent actions,
    - Prompt LLM to generate a **thought** and an **action**
    - Execute an action against the KG
    - Add an **observation** to the history

Main agent code available in this file,
implemented with LangGraph (part of LangChain) in Python

Zero-Shot LLM Policy Prompt

instruction

Your task is to write a Wikidata SPARQL query to answer the given question. Follow a step-by-step process:

1. Start by constructing very simple fragments of the SPARQL query.
2. Execute each fragment to verify its correctness. Adjust as needed based on your observations.
3. Confirm all your assumptions about the structure of Wikidata before proceeding.
4. Gradually build the complete SPARQL query by adding one piece at a time.
5. Do NOT repeat the same action, as the results will be the same.
6. The question is guaranteed to have an answer in Wikidata, so continue until you find it.
7. If the user is asking a True/False question with only one answer, use ASK WHERE to fetch a True/False answer at the very end.
8. In the final SPARQL projections, do not only ask for labels.
   Ask for the actual entities whenever needed (e.g. instead of doing `SELECT xLabel`, do `SELECT x`).
9. If the final result was contained in last round's `get_wikidata_entry` and you are ready to stop,
   use `execute_sparql` and generate a SPARQL to retrieve that results.

Form exactly one "Thought" and perform exactly one "Action", then wait for the "Observation".

Possible actions are:

- get_wikidata_entry(QID): Retrieves all outgoing edges (linked entities, properties, and qualifrs) of a specified Wikidata entity using its QID.
- search_wikidata(string): Searches Wikidata for entities or properties matching the given string.
- get_property_examples(PID): Provides a few examples demonstrating the use of the specified property (PID) in Wikidata.
- execute_sparql(SPARQL): Runs a SPARQL query on Wikidata and returns a truncated result set for brevity.
- stop(): Marks the last executed SPARQL query as the final answer and ends the process.

```
# input
{% for i in range(0, conversation_history|length) %}
User Question: {{ conversation_history[i]["question"] }}
{% for j in range(0, conversation_history[i]["action_history"]|length) %}
{{ conversation_history[i]["action_history"][j] }}
{% endfor %}
--
{% endfor %}

User Question: {{ question }}

{% if action_history %}
{% for i in range(0, action_history|length) %}

{{ action_history[i] }}
{% endfor %}
{% endif %}

Output one "Thought" and one "Action":
```

Templates written in jinja syntax

Looping in SPINACH

- SPINACH needs to explore different queries during iteration
  - Run policy prompt with temperature 1.0
- LLM can fall into a loop of executing the same action

```
execute_sparql("SELECT ?x WHERE {wd:Q1 wdt:P1 ?x.}")
Did not find any results
execute_sparql("SELECT ?x WHERE {wd:Q1 wdt:P1 ?x.}")
Did not find any results
execute_sparql("SELECT ?x WHERE {wd:Q1 wdt:P1 ?x.}")
…
```

Quiz: Why would this happen?
Quiz: How to solve this?

Looping in SPINACH

- If repeated actions are found,
  - Agent resets exploration state to the one before the repetition
  - Continues from there

- We cap number of actions by either:
  - 15 actions after taking “rollbacks” into account
  - a total of 30 actions including “rollbacks”

The SPINACH Agent - Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

Thought: First, I need to identify the properties and entities related to people affiliated with the University of Washington School of Music ...

```
>> search_wikidata("University of Washington School of Music")
```

Observation: University of Washington School of Music (Q98035717)

[3 more time steps omitted]

```
>> search_wikidata("musical instrument")
```

Looking up relevant PIDs and QIDs

The SPINACH Agent - Example

Question: What are the musical instruments played by people who are affiliated with the University of Washington School of Music and have been educated at the University of Washington, and how many people play each instrument?

```
>> execute_sparql("""
SELECT ?instrument (COUNT(*) AS ?count) WHERE {
  ?person wdt:P69 wd:Q1063349;
    wdt:P1416 wd:Q98035717;
    wdt:P1303 ?instrument.
}
GROUP BY ?instrument
""")
Obesrvation: Empty result.
```

Attempting to directly write the SPARQL fails due to the hallucinated Q1063349

The SPINACH Agent - Example

```
Thought: Since my previous queries returned empty results, I need to
verify if there are any individuals in Wikidata who have been educated at
the University of Washington. This will help me confirm if I am using the
correct properties and entities.
>> get_wikidata_entry("Q1063349")
Observation:
  Spasskoye (Q1063349) {
    country (P17): Russia (Q159),
    ... more entries ...
  }

Thought: It seems like there is an issue with the QID I used for the
University of Washington. I should search for the correct QID.
>> search_wikidata("University of Washington")
Observation:
  - University of Washington (Q219563)
    public research university in Seattle, Washington, United States
```

Attempting to debug the issue by running simpler SPARQL queries

```
>> execute_sparql("""
SELECT ?instrument ?instrumentLabel (COUNT(?student) AS ?count) WHERE {
  ?student wdt:P1303 ?instrument;
    wdt:P1416 wd:Q98035717;
    wdt:P69 wd:Q219563.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
GROUP BY ?instrument ?instrumentLabel
""")
Observation:
   | instrument    | instrumentLabel  | count |
   |---------------|------------------|-------|
   | Q5994         | piano            | 99    |
   | Q1467960      | mbira            | 2     |
   | Q8350         | trombone         | 11    |
   | Q8338         | trumpet          | 8     |
   | Q17172850     | voice            | 32    |
   | ...           | ...              | ...   |
   | Q302497       | mandolin         | 1     |
   | Q187851       | recorder         | 1     |
   | Q185041       | cor anglais      | 1     |
   | Q83509        | piccolo          | 1     |

>> stop()
```

Writing the correct SPARQL query after correcting the mistake

P1303: instrument
P1416: affiliation
P69: educated at

Q98035717:
University of Washington
School of Music

Q219563:
University of Washington

Lecture Goals

Teach the Agentic Approach so You Can Use It in Your Projects!

- Knowledge Graphs – aka Knowledge Bases (KB)
  - Why KBs?
  - Why is KBQA (KB Question Answering) Challenging?
    -- Previous Work
  - The Agentic Approach for KBs
  - **How to Get a Good KBQA Dataset?**
  - Evaluation
- Applying the Agentic Approach to SQL Databases

KBQA Data Sets (Crowdsourcing)

- **Datasets with natural questions originally collected through search engines or crowdsourcing** ← Quiz: Is this OK?
  - WebQuestionSP (Yih et al., 2016)
  - QALD datasets (Usbeck et al., 2017, 2018, 2023; Perevalov et al., 2022)
  - RuBQ (Korablinov and Braslavski, 2020)
  - SimpleQuestions (Bordes et al., 2015)

Simple Queries

KBQA Data Sets (Synthesized)

- **Datasets with synthetically generated logical forms & questions**
  - ComplexWebQuestions (Talmor and Berant, 2018)
  - GrailQA (Gu et al., 2021)
  - KQA Pro (Cao et al., 2022a)
  - CFQ (Keysers et al., 2020)
  - CWQ (Talmor and Berant, 2018)
  - LC-QuAD2 (Dubey et al., 2019)

← Quiz: Is this OK?

Limited NL variety
& Unique query patterns

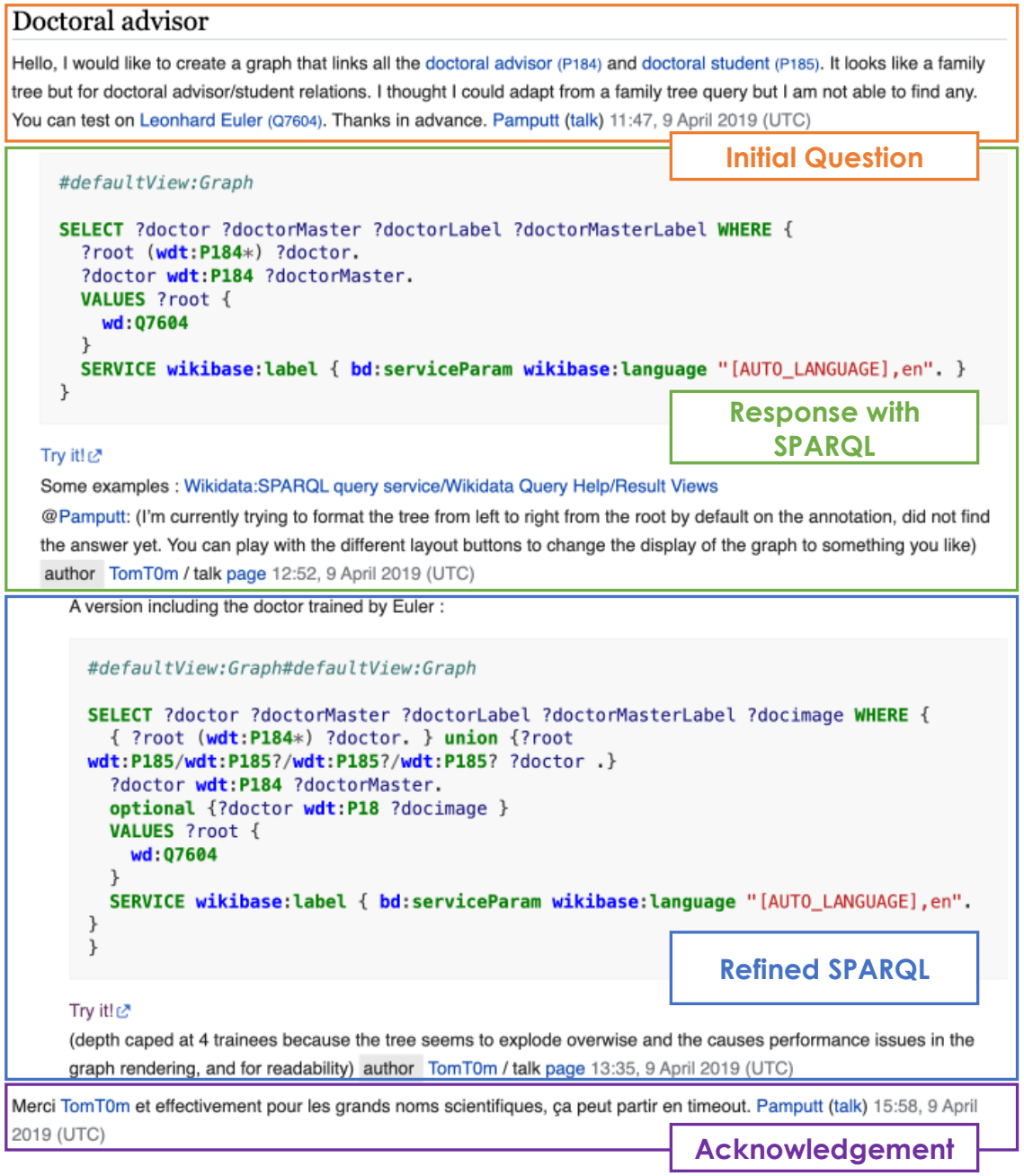
Dataset	Size	UQPs	UQPs / Size
GrailQA (train+dev)	51100	116	0.227%
KQA-Pro	117970	1689	1.432%
CWQ	34689	402	1.159%
WikiWebQuestions	4316	176	0.041%
SPINACH	320	320	100.0%

Table 6: Comparison of Unique Query Patterns (UQPs) in SPINACH and prior works.

Wikidata SPARQL Forum

https://m.wikidata.org/wiki/Wikidata:Request_a_query

- To help Wikidata users write SPARQL queries
- People exchange conversations on how to write SPARQLs
- The queries are real, but difficult



The SPINACH dataset

- Based on discussions on the Wikidata Request Query forum from July 2016 – May 2024
- Categories of clauses removed:
  - Wikimedia presentation queries
  - Questions on complex SPARQL code
  - Queries obscured by optimizations
  - Formatting clauses
- Annotating Natural Questions:
  - Disambiguate entities and properties
  - Natural verbalizations
  - Accurately capturing optional clauses and projections
- 155 validation examples and 165 test examples ← Quiz: This number seems small. Is this enough?

SPINACH dataset:
natural questions + complex logical forms

	Avg. Clauses	Avg. Projs	Avg. Rels	Avg. Subjs	Avg. Preds	Avg. Objs	Avg. Lits
Natural questions w/ annotated logical forms							
WikiWebQuestion (Xu et al., 2023)	2.63	1	1.53	1.25	1.52	1.53	0.04
QALD-9 Plus (Perevalov et al., 2022)	3.14	1	1.77	1.26	1.70	1.78	0.05
QALD-10 (Usbeck et al., 2023)	2.38	1	1.27	1.19	1.17	1.32	0.05
RuBQ (Korablinov and Braslavski, 2020)	2.17	1	1.12	1.03	1.11	1.07	0.01
SimpleQuestionsWikidata (Diefenbach et al., 2017b)	2.00	1	1.00	1.00	1.00	1.00	0.00
Synthetic logical forms w/ synthetic or paraphrased questions							
CWQ (Talmor and Berant, 2018)	5.19	1	2.80	1.87	2.62	3.38	0.11
GrailQA (Gu et al., 2021)	7.10	1	3.02	1.97	2.43	3.90	0.08
KQA Pro (Cao et al., 2022a)	6.34	1	5.01	2.77	3.94	2.43	2.37
MCWQ (Cui et al., 2022)	6.34	1	5.09	2.67	3.53	3.37	0.00
LC-QuAD-2 (Dubey et al., 2019)	3.65	1	2.07	1.51	2.05	2.07	0.22
Natural logical form w/ annotated questions							
SPINACH **(Ours)**	8.89	2.50	4.03	1.76	3.55	4.53	0.46

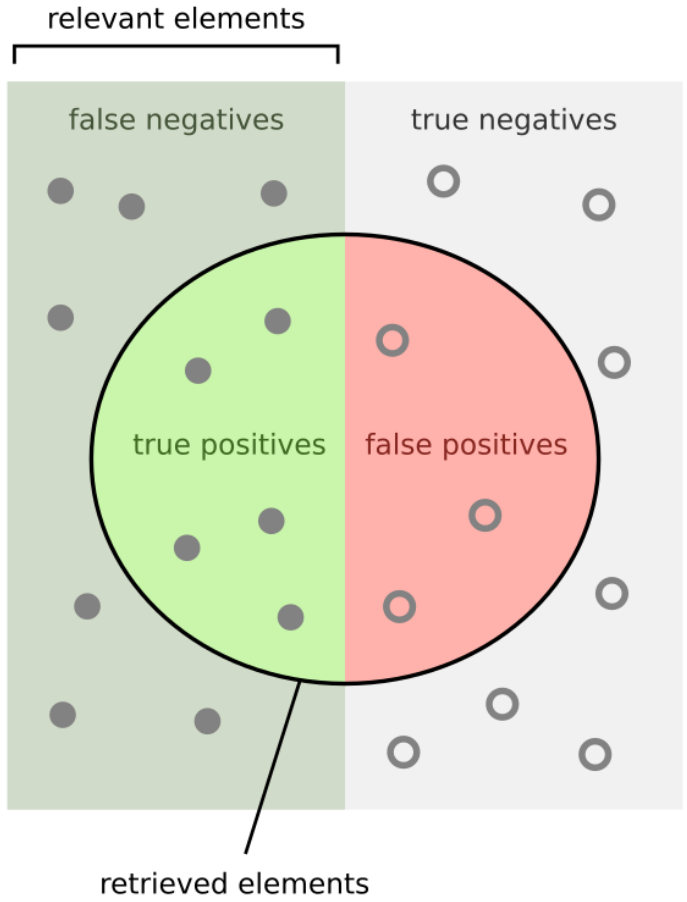
Lecture Goals

Teach the Agentic Approach so You Can Use It in Your Projects!

- Knowledge Graphs – aka Knowledge Bases (KB)
  - Why KBs?
  - Why is KBQA (KB Question Answering) Challenging?
    -- Previous Work
  - The Agentic Approach for KBs
  - How to Get a Good KBQA Dataset?
  - **Evaluation**
- Applying the Agentic Approach to SQL Databases

The F1 metric

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}.$$



How many retrieved items are relevant?

Precision =

How many relevant items are retrieved?

Recall =

But: traditional EM & F1does not work

Predicted

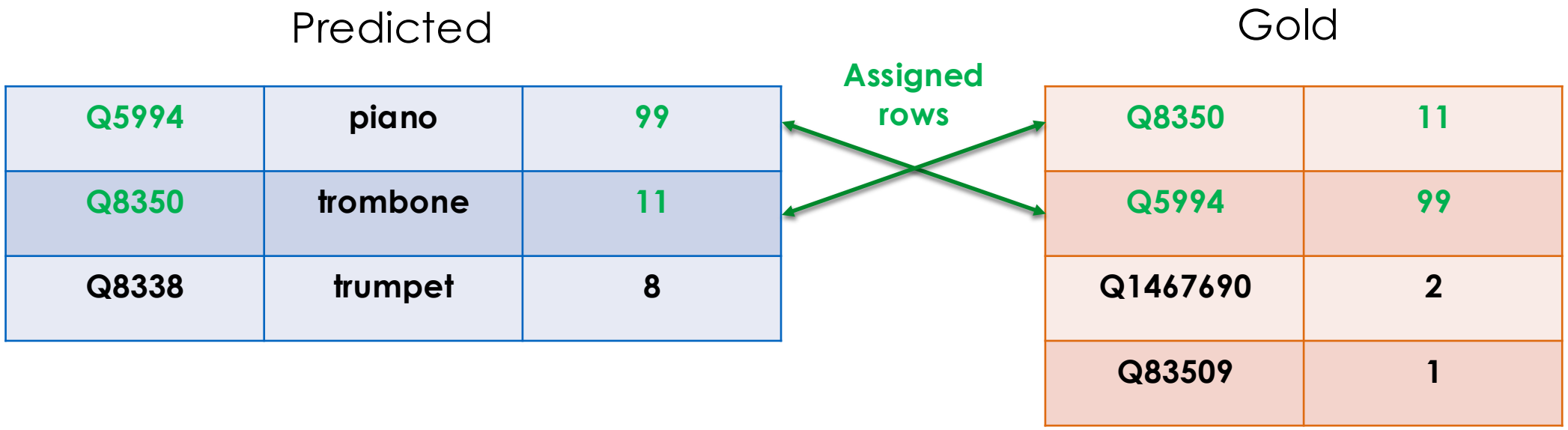
Minnehaha County	**206,930**
Pennington County	**115,903**
Lincoln County	**73,238**

Gold

Minnehaha County
Pennington County
Lincoln County

Question: Do we want to penalize this column?

New Metric: **row-major** EM & F1



Step 1: Run an **assignment algorithm** to maximize **total overlap** (recall)

Without penalizing extra columns in prediction

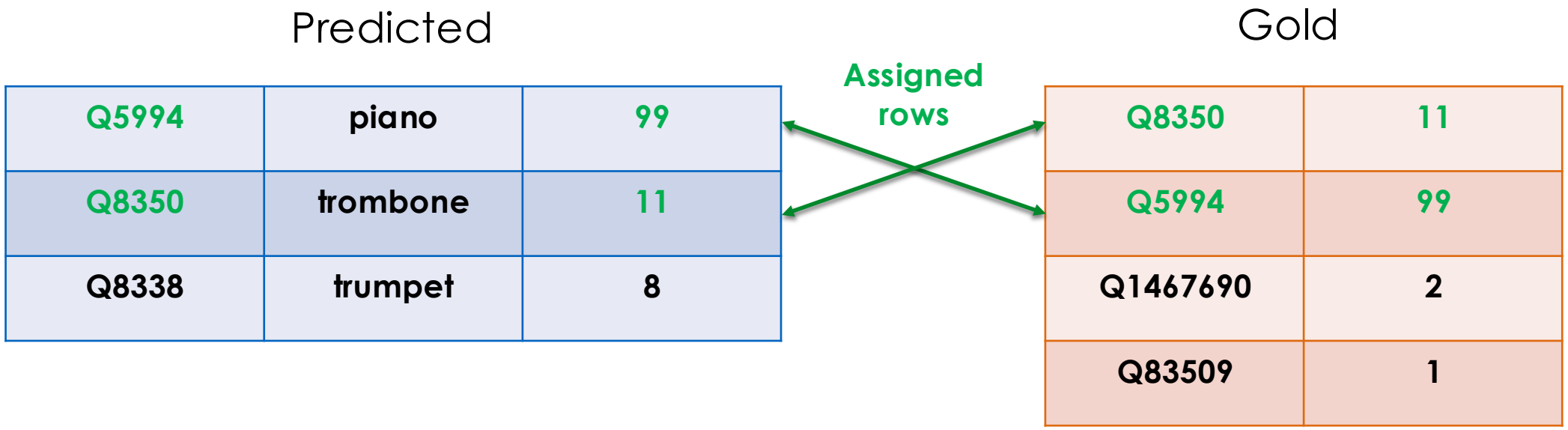
y_i a row in gold

y'_i a row in prediction

$$\text{recall}(y_i, y'_j) = \frac{|y_i \cap y'_j|}{|y_i|}$$

Matching rows with 0 recall is not allowed

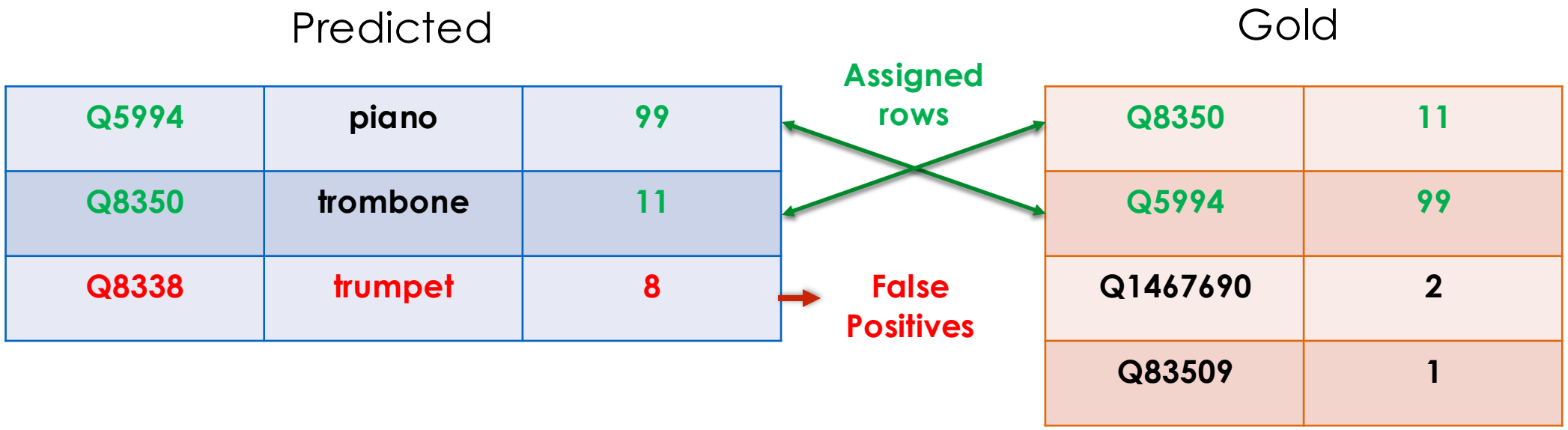
New Metric: **row-major** EM & F1



Step 2: Calculate **true positives** as sum of recalls in assigned rows

$$tp = \sum_{(i,j)\in A(\mathbf{y},\mathbf{y}')} \text{recall}(y_i, y'_j) \qquad = 2$$

New Metric: **row-major** EM & F1



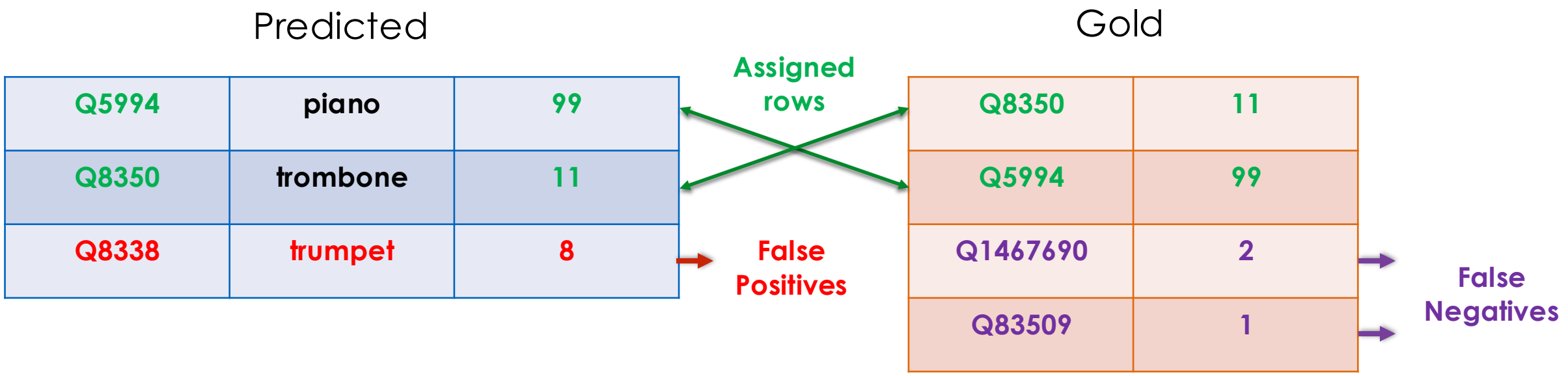
Step 3: Each unassigned row in prediction counts as a **false positive**

n': number of rows in prediction

r: number of rows in matching

$$fp = n' - r \qquad = 1$$

New Metric: **row-major** EM & F1



Step 4: Each unassigned row in gold counts as a **false negative**

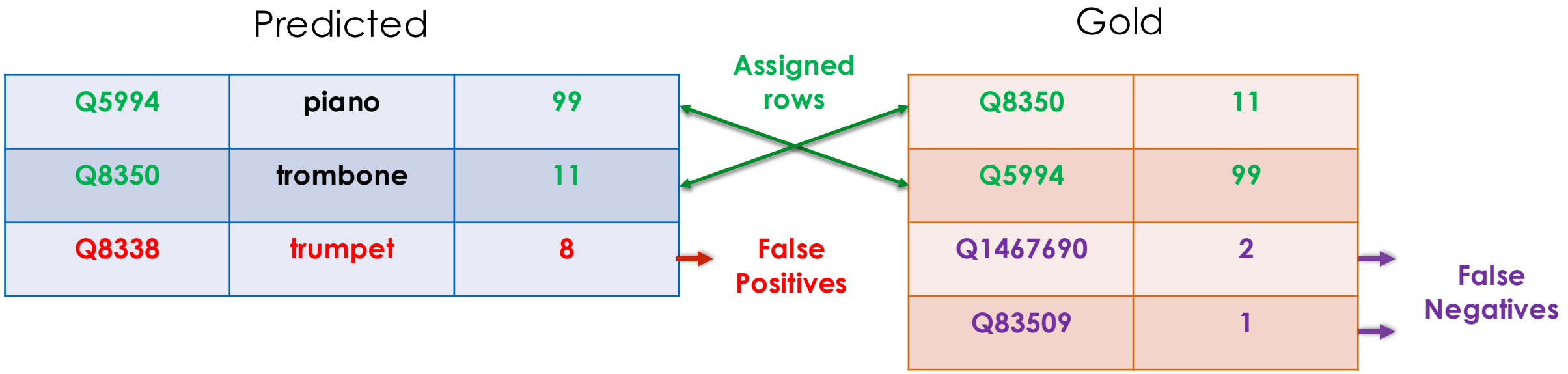
plus sum of 1– recall in the matching (0 in this case)

n: number of rows in gold

r: number of rows in matching

$$fn = n - r + \sum_{(i,j)\in A(\mathbf{y},\mathbf{y}')} 1 - \text{recall}(y_i, y'_j) \quad = 2$$

New Metric: **row-major** EM & F1



Step 5: Calculate F1 with the usual formula

$$F_1 = \frac{2tp}{2tp+fn+fp} = 0.66$$

Results on the SPINACH Dataset

Fine-tuned model ->
SOTA. Ask LLM to walk the graph ->

	Dev		Test	
	EM	F1	EM	F1
Direct GPT-4o Question Answering	0.0	3.9	0.0	4.0
GPT-4o Generating SPARQL	1.3	5.4	0.6	3.9
Fine-tuned WikiSP (Xu et al., 2023)	1.3	3.5	1.2	7.1
0-shot ToG (GPT-4) (Sun et al., 2024a)	3.9	9.8	1.8	7.2
0-shot SPINACH agent (GPT-4o) **(Ours)**	**21.4**	**46.4**	**16.4**	**45.3**

SPINACH agent achieves considerable gain over prior approaches!

Fine-tuned LLMs Know More, Hallucinate Less with Few-Shot Sequence-to-Sequence Semantic Parsing over Wikidata, Xu et al, EMNLP 2023
Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph, Sun et al, ICLR 2023

Results on Other Datasets

	QALD-7 (Task 4) Test		QALD-9 Plus (en) Test		QALD-10 (en) Full Test Set		QALD-10 (en) Subset in ToG		WikiWebQuestions Dev		WikiWebQuestions Test	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
STAGG (Yih et al., 2016)	-	19.0	-	-	-	-	-	-	-	-	-	-
GGNN (Sorokin and Gurevych, 2018)	-	21.3	-	-	-	-	-	-	-	-	-	-
LingTeQA (To and Reformat, 2020)	-	34.0	-	-	-	-	-	-	-	-	-	-
Baramiia et al. (2022)	-	-	-	-	-	42.8	-	-	-	-	-	-
Shivashankar et al. (2022)	-	-	-	-	-	49.1	-	-	-	-	-	-
QAnswer (Diefenbach et al., 2017a)	-	40.0	-	44.6	-	57.8	-	-	-	-	-	-
SPARQL-QA (Borroto et al., 2022)	-	-	-	-	-	59.5	-	-	-	-	-	-
Liu et al. (2024)	-	-	-	-	56.5	-	-	-	-	-	-	-
0-shot ToG (GPT-4) (Sun et al., 2024a)	-	-	-	-	-	-	54.7	-	-	-	-	-
Fine-tuned WikiSP (Xu et al., 2023)	38.0	43.6	-	-	-	-	-	-	**75.6**	**76.9**	**65.5**	**71.9**
0-shot SPINACH agent (GPT-4o) **(Ours)**	**62.2**	**74.6**	**58.3**	**71.6**	**63.1**	**69.5**	**64.7**	**72.4**	61.2	72.3	59.9	70.3

Zero-shot ICL (in-context learning) achieves new SOTA on QALD Wikidata datasets

Comes within 1.6 F1 on WikiWebQuestions to WikiSP, fine-tuned on WikiWebQuestions

The Importance of Agent Actions

	EM	F1
SPINACH agent	**21.4**	**46.4**
w/o **`get_wikidata_entry`**	11.7	36.4
w/o **`get_property_examples`**	10.4	29.4
w/o **`search_wikidata`**	4.6	25.3

All actions make meaningful contribution to the agent performance

Live Demo:

https://spinach.genie.stanford.edu/

Code:

https://github.com/stanford-oval/spinach

As a bot on Wikidata:

https://www.wikidata.org/wiki/User:SpinachBot

Error Analysis

40%: **Property-related problems**:
Fails to fetch the correct property or incorrectly uses a property

30%: **Complicated SPARQL**:
Fails to write complex SPARQL to fetch results.

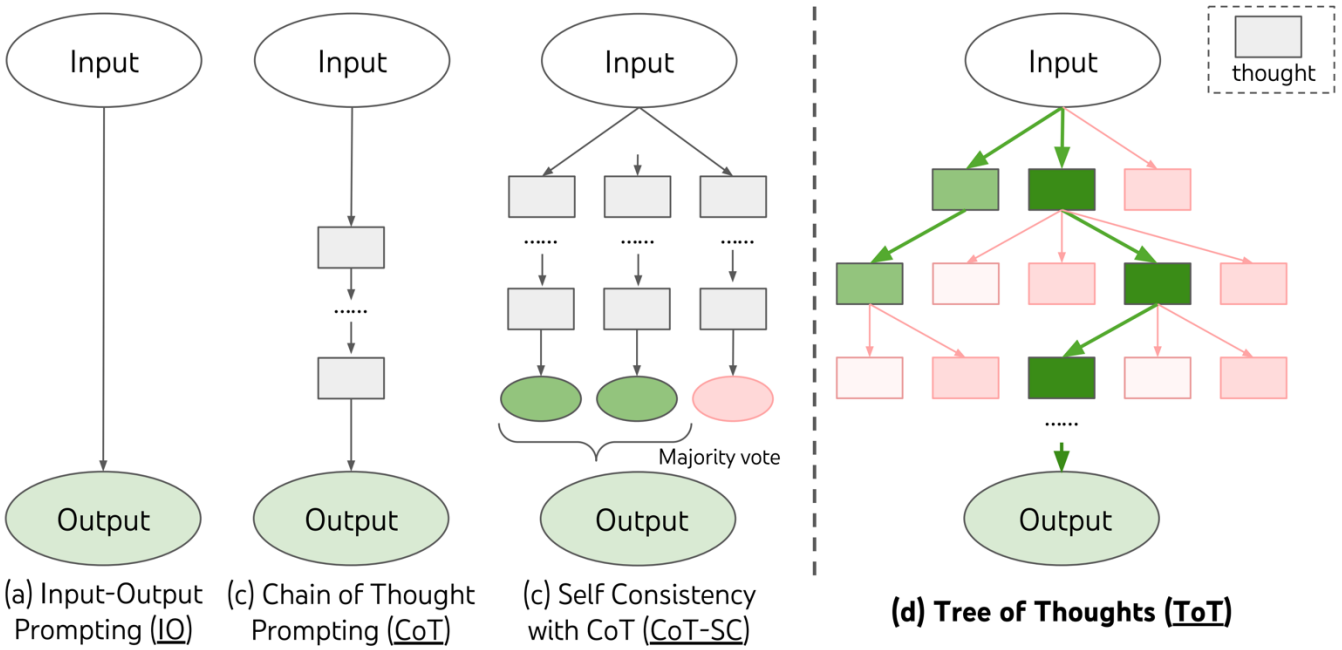
15%: **Not enough exploration:**
Insufficient exploration within limit of actions allowed.

10%: **Inaccurate semantic parsing**:
LLM injecting an extra clause.

5%: **Formatting issues**

Potential Solution #1

- Expand navigations further. Use a tree to explicitly keep track of different actions one can take at a step.



Tree of Thoughts: Deliberate Problem Solving with Large Language Models, Yao et al, NeurIPS 2023

SPINACH Deployed on Wikidata Forum

https://m.wikidata.org/wiki/Wikidata:Request_a_query

- 600+ conversations in the wild: all real and hard queries!
- 198 randomly selected conversations
  - Success rate: 78% (154 cases) A higher success rate in practice!
  - Failures: 22% (44 cases)
    - 50% (22 cases) similar to queries in the dataset
    - 50% (22 cases) are not similar:
      underspecified queries, query correction/modification,
      string manipulation
- Next step: Can we fine-tune an open-source model?

Lecture Goals

Teach the Agentic Approach so You Can Use It in Your Projects!

- Knowledge Graphs – aka Knowledge Bases (KB)
  - Why KBs?
  - Why is KBQA (KB Question Answering) Challenging?
    -- Previous Work
  - The Agentic Approach for KBs
  - How to Get a Good KBQA Dataset?
  - Evaluation
- **Applying the Agentic Approach to SQL Databases**

The New York Times

Inside the Late-Night Parties Where Hawaii Politicians Raked In Money

After the state passed a law barring government contractors from donating to politicians, fund-raising parties showed just how completely the reform effort failed.

By Blaze Lovell, Eric Sagara and Irene Casado Sanchez

The reporters examined campaign contributions and government contracts for this article, part of a series about loopholes in Hawaii's pay-to-play laws, for The Times's Local Investigations Fellowship.

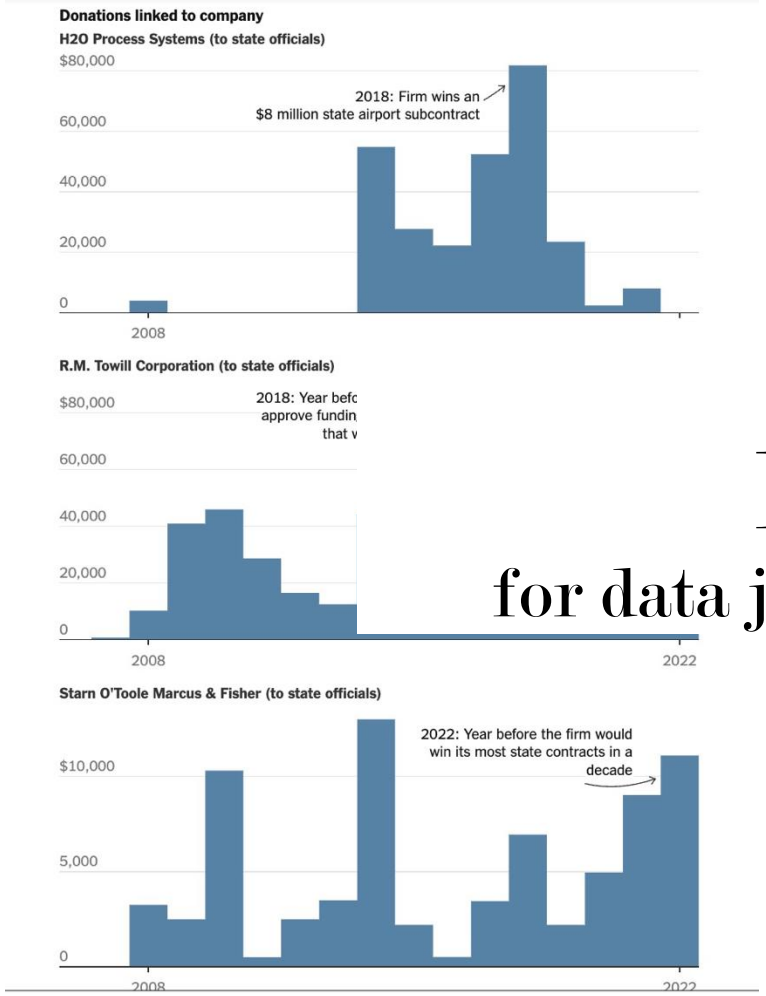
April 17, 2024

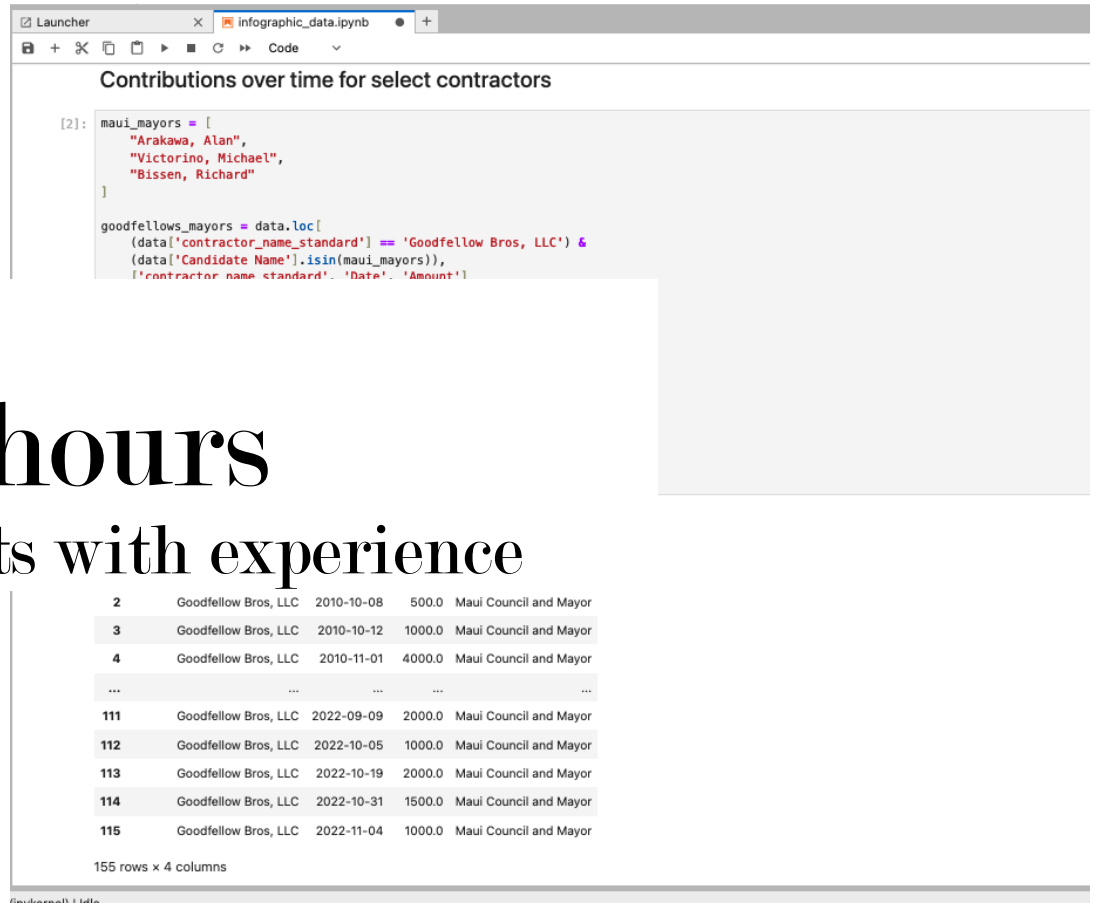
This article was reported in partnership with Big Local News at Stanford University.

https://www.nytimes.com/2024/04/17/us/haw
contractors-campaign-donations.html

The New York Times

Inside the Late-Night Parties Where Hawaii Politicians Raked In Money





100+ hours

for data journalists with experience

https://www.nytimes.com/2024/04/17/us/hawaii-contractors-campaign-donations.html

Increase Truthful Reporting in Journalism

Can Journalists Just *TALK* to Data

Without Needing a Data Scientist?

Stanford | ENGINEERING
Computer Science

THE BROWN INSTITUTE FOR MEDIA INNOVATION

Get started

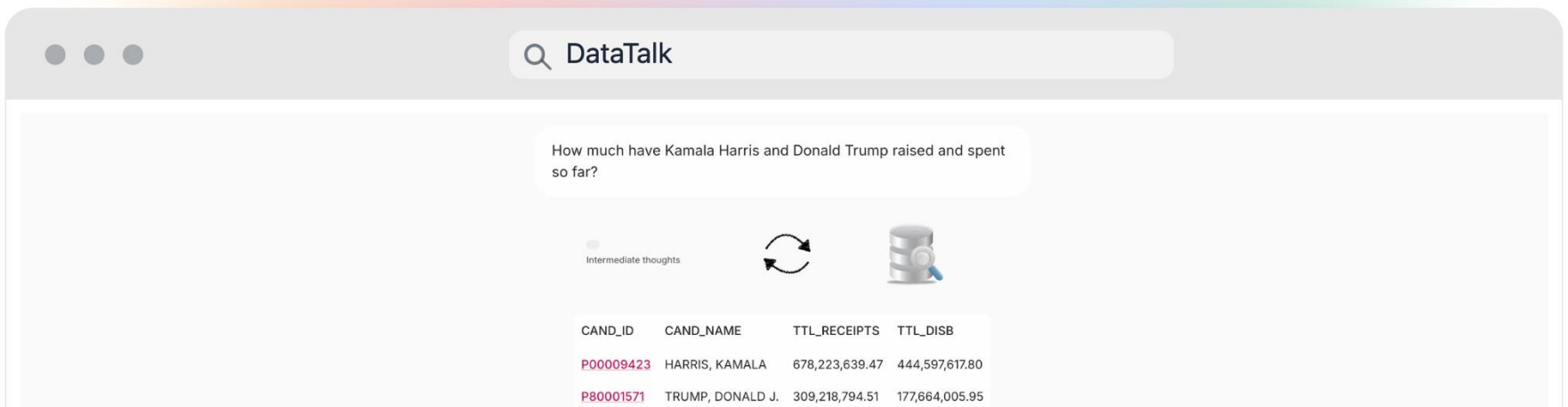
✨ Supported by a Magic Grant from the Brown Institute, the Verdant Foundation, and Microsoft Azure Credits

DataTalk

A Campaign Finance Agent (Beta)

Chat with publicly available campaign finance data from
the Federal Election Commission (FEC) and
PAC classification data from OpenSecrets.org

Get Started →



DataTalk

A Campaign Finance Agent (Beta)

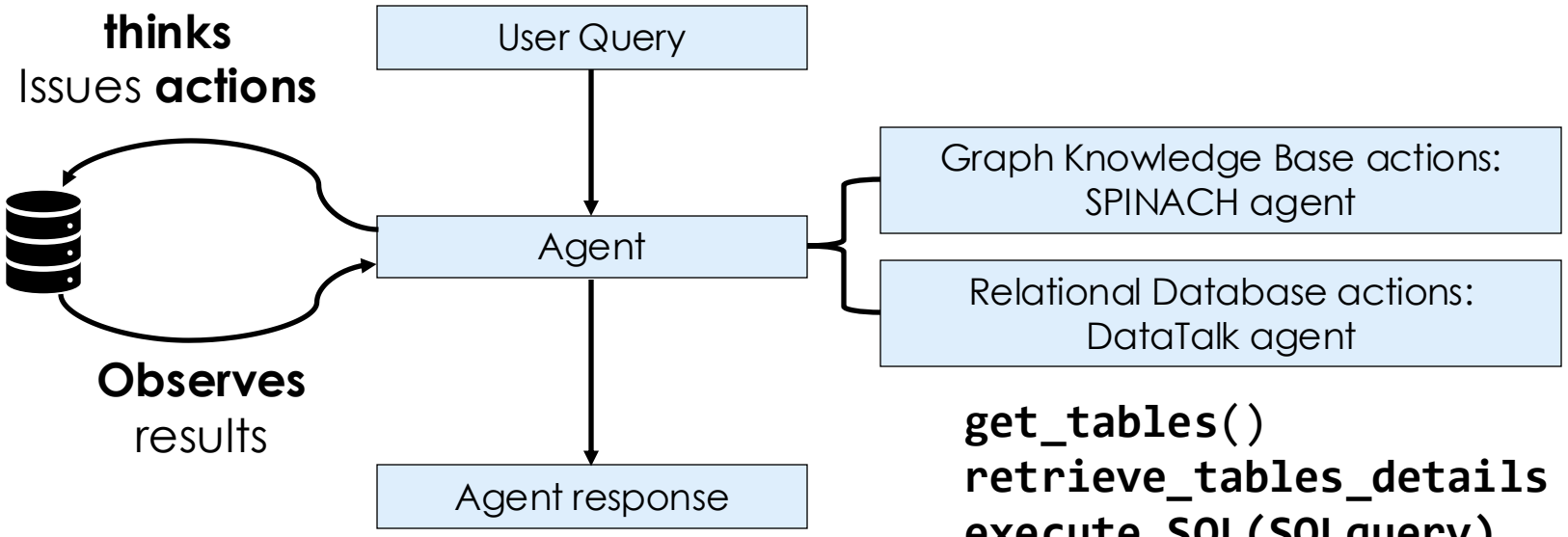
Shicheng Liu[1] Harold Triedman[4] Leah Harrison[3] Eryn Davis[3]
Sajid Omar Farook[1] Alexander Spangher[5] Cheryl Phillips[2] Derek Willis[6]
Serdar Tumgoren[2] Monica S. Lam[1]

[1] Stanford CS [2] Stanford Big Local News [3]Columbia Journalism School
[4] Cornell Tech [5] USC [6] University of Maryland

DataTalk: Campaign Finance

- Chat with publicly available campaign finance data
- Based on
  - Federal Election Commission (FEC)
  - OpenElections.org data
- 36 Large, relational databases

Agentic Approach for Knowledge Bases




```
get_tables()
retrieve_tables_details ([table1, table2, …])
execute_SQL(SQLquery)
stop()
```

Spoiler: This is not enough yet!

The DataTalk Agent - Actions

get_tables()

```
{'table_name': 'committee_type_codes', 'comments': ’’},
{'table_name': 'party_codes', 'comments': ’’},
{'table_name': 'pac_details', 'comments': 'based on OpenSecrets data'},
{'table_name': 'candidate_committee_linkage_2023_2024', 'comments':
\"The candidate-committee linkage file contains information linking the
candidate's information to information about his or her
committee.\\n\\nThis file shows the candidate’s identification number,
candidate’s election year, FEC election year, committee identification
number, committee type, committee designation, and a linkage
identification number."},
…
```

The DataTalk Agent - Actions

```
retrieve_tables_details(['contributions_by_individuals_2023_2024', 'committee_master_2023_2024'])
```

```
CREATE TABLE "contributions_by_individuals_2023_2024" (
    "CMTE_ID" VARCHAR(9), -- Filer identification number. A 9-character alpha-numeric code assigned to a committee by the Federal Election Commission
    "AMNDT_IND" VARCHAR(1), -- Amendment indicator. Indicates if the report
    …
);
CREATE TABLE "committee_master_2023_2024" (
    "CMTE_ID" VARCHAR(9) PRIMARY KEY, -- Committee identification. A 9-character alpha-numeric code assigned to a committee by the Federal Election Commission. Committee IDs are unique and an ID for a specific committee always remains the same.
    "CMTE_NM" VARCHAR(200),
    …
)
```

The DataTalk Agent - Actions

execute_sql(

```sql
SELECT p."CMTE_ID", p."name", s."TTL_RECEIPTS", s."CVG_END_DT", m."CMTE_FILING_FREQ"
FROM "pac_details" p
JOIN "pac_and_party_summary_2023_2024" s ON p."CMTE_ID" = s."CMTE_ID"
JOIN "committee_master_2023_2024" m ON p."CMTE_ID" = m."CMTE_ID"
WHERE 'cryptocurrency' = ANY(p."industry")
ORDER BY s."TTL_RECEIPTS" DESC
LIMIT 10;
```

)

CMTE_ID	name	TTL_RECEIPTS	CVG_END_DT	CMTE_FILING_FREQ
C00835959	FAIRSHAKE	2.0407e+08	2024-10-16	M
C00836221	DEFEND AMERICAN JOBS	5.9075e+07	2024-10-16	M
C00848440	PROTECT PROGRESS	3.6786e+07	2024-10-16	M
C00804179	COINBASE, INC. INNOVATION PAC (COINBASE INNOVATION PAC)	209856	2024-10-16	M
C00809020	AMERICAN DREAM FEDERAL ACTION	84031.8	2024-09-30	Q
C00824896	BLOCKCHAIN ASSOCIATION POLITICAL ACTION COMMITTEE (BA PAC)	60290.9	2024-10-16	M
C00845651	PAYWARD VENTURES, INC. PAC (KRAKEN PAC)	59459.2	2024-09-30	Q
C00801514	PROTECT OUR FUTURE PAC	37891.5	2024-10-16	M
C00567412	CHAMBER OF DIGITAL COMMERCE POLITICAL ACTION COMMITTEE	17563.8	2024-09-30	Q
C00804732	CRYPTO INNOVATION	16102.8	2024-09-30	Q

Real-World Experience with Journalists

- **Used by journalists and journalism students**
- **Agentic approach is necessary**
  - Too many tables: need to retrieve knowledge about the tables
- **Interpretation of results are difficult**
  - Lots of caveats on the data
    (e.g. contributions below $200 not included in some tables)
  - Requires experts on the data
- **Need to capture the expertise just like an apprentice**
  - More in the next class

Conclusions

- **The Agentic Approach**: effective for complex knowledge base queries
  - **Acquire knowledge** and to react to results
- Wikidata
  - SPINACH dataset is based on hard real queries
  - Achieves 45.3 F1 with the SPINACH dataset
  - **Achieves 78% accuracy in Wikidata Request-a-Query forum**
- FEC data
  - **The same agentic approach handles the large tables**
  - **Discover the problem of tacit expert knowledge (next lecture)**