北京航空航天大學
BEIHANG UNIVERSITY

Microsoft

Unleashing the Power of Pre-trained Language Models for Semantic Parsing

Qian Liu (刘乾)

qian.liu@buaa.edu.cn

Joint Ph.D. of BUAA and MSRA

Talk @ Shanghai Jiao Tong University 2022.01.19

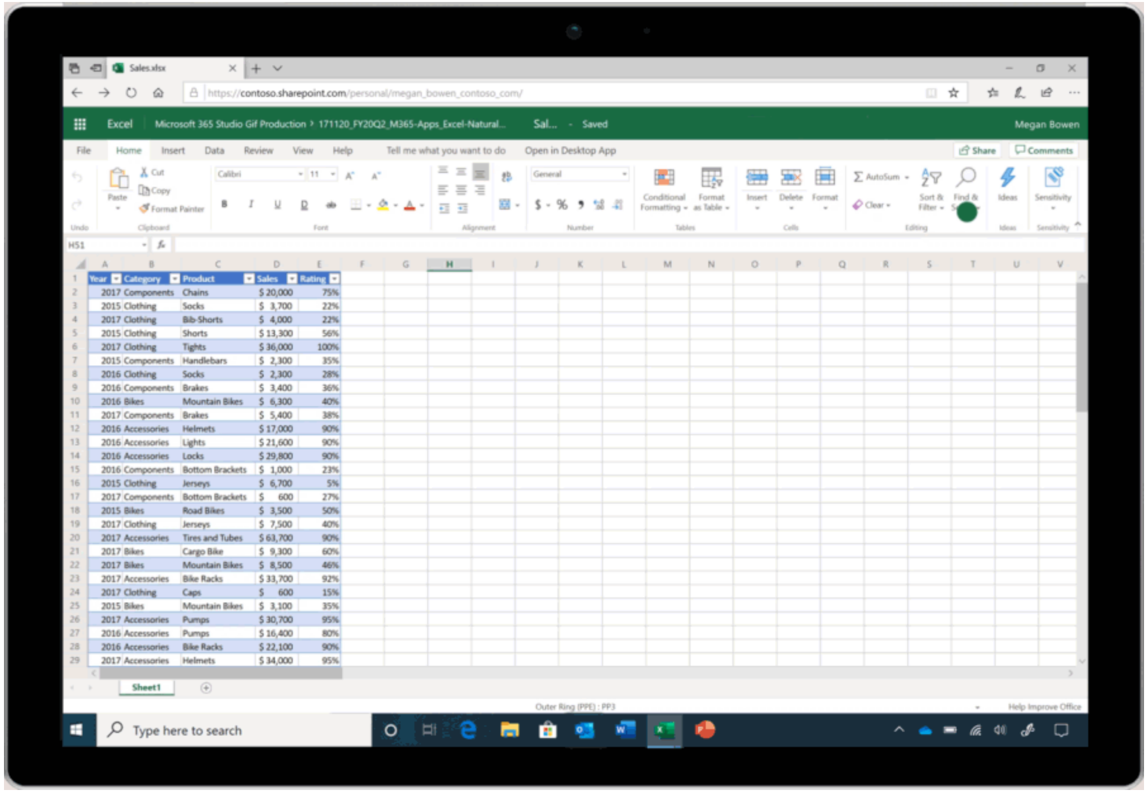
Self Introduction

Qian Liu Joint Ph.D. of BUAA and MSRA

- **Education Background**
  - B.S. and Ph.D. in Beihang University, Computer Science
  - Ph.D. Supervisor: Qinping Zhao & Jian-Guang Lou
  - Internship at MSRA from July 2016 to Now
- **Research Interest**
  - Natural Language Processing
  - Semantic Parsing

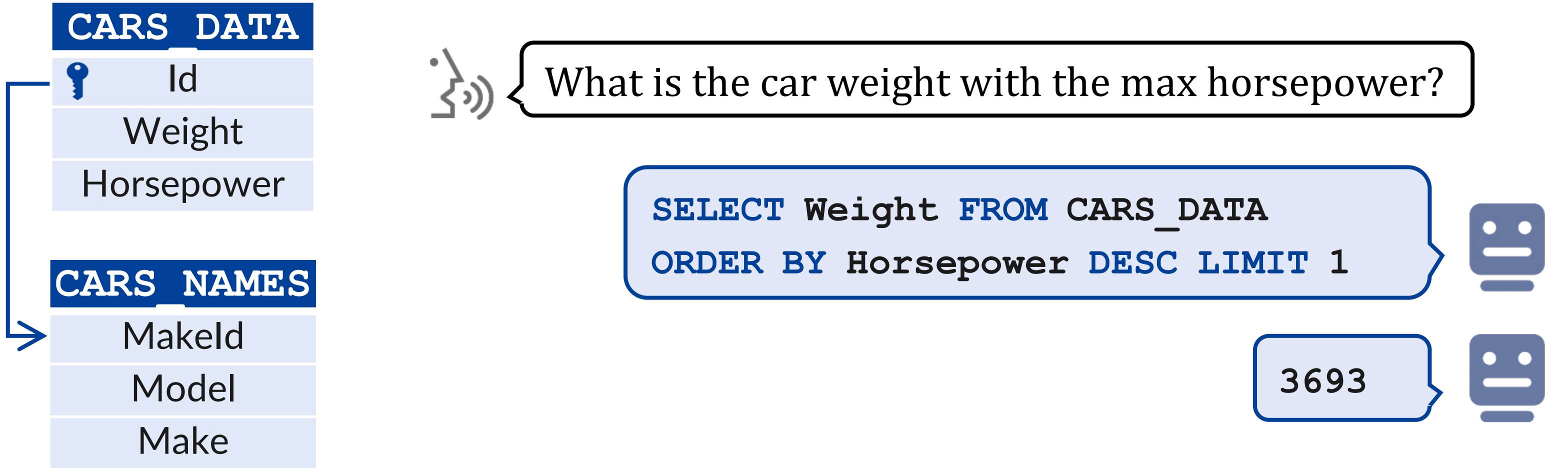
Research Problem: AI V.S. Programmer?

Research Vision: AI + Programming



Key Technology: Semantic Parsing

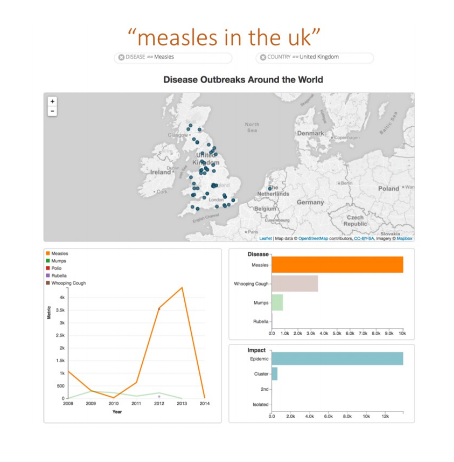
Translate a user's natural language sentences into machine-executable formal programming language (e.g., SQL) to accomplish relevant tasks.



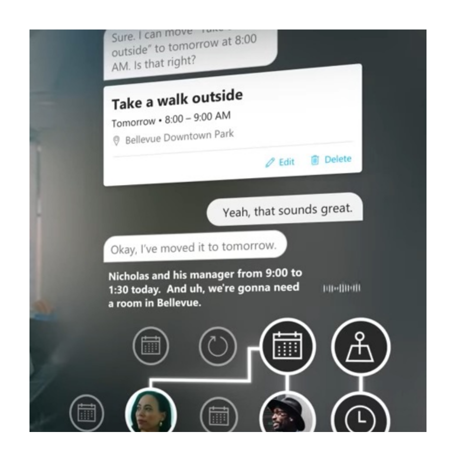
Key Technology: Semantic Parsing

Semantic parsing techniques has been widely applied in various applications and relieve users from the burden of learning programming.

Information Retrieval



Data Analysis



Virtual Assistants

Unleashing the Power of Pre-trained Language Models

Part 1: Towards Interpretable Parser

Grounding power of PLMs helps build interpretable semantic parsers.

Part 2: Towards Cost-Effective Parser

Learning latent program helps build cost-effective semantic parsers.

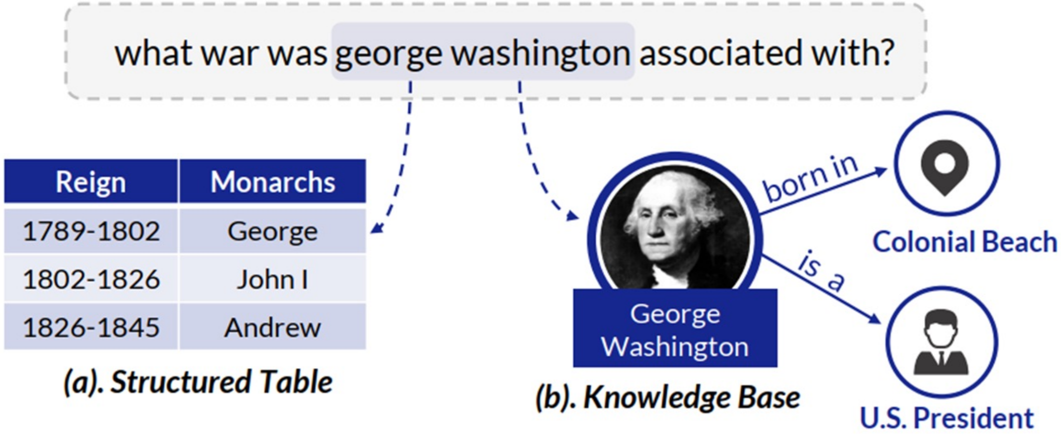
Part 3: Towards Synthesized Parser

Generalizing power of PLMs helps build synthesized semantic parsers.

Liu Q et al, Awakening Latent Grounding from Pretrained Language Models for Semantic Parsing. **ACL 2021 Findings**.

Background: Grounding

- Broadly speaking, Grounding means *"connect varying linguistic symbols to real-world perception"*. For example, george washington can be grounded into either a cell value in a structured table, or an entity in knowledge bases.



(a). Structured Table

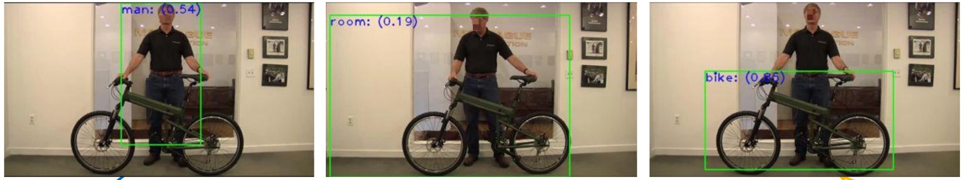
(b). Knowledge Base

[Roy. 2005]

Background: Grounding

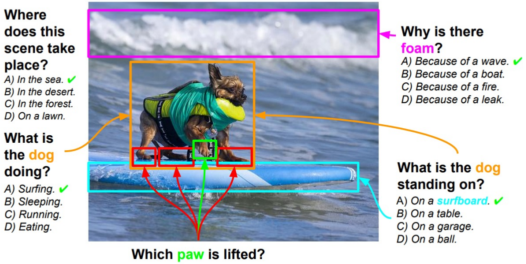
- Grounding improves interpretability and shows better performance empirically across different downstream tasks than without grounding.

Video Descriptions



A man is seen standing in a room speaking to the camera while holding a bike.

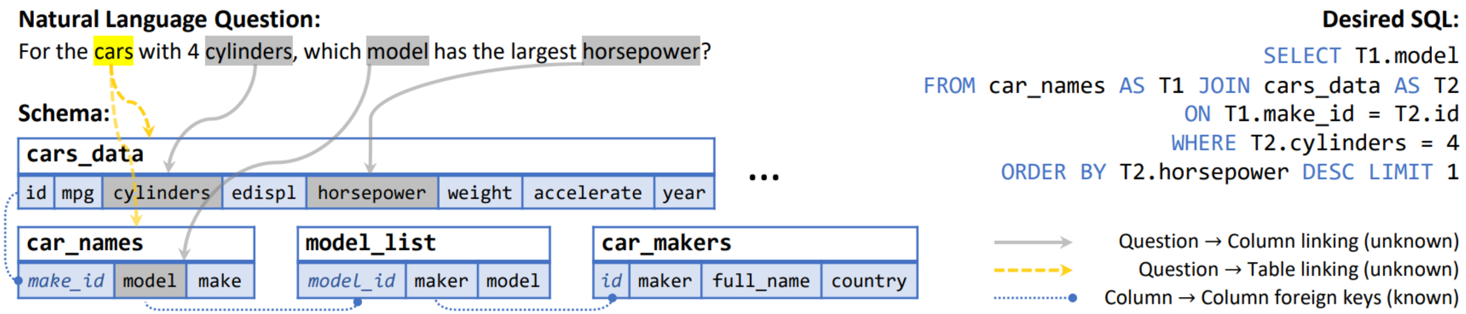
Visual Question Answering



Background: Grounding

- Grounding improves interpretability and shows better performance empirically across different downstream tasks than without grounding.

Semantic Parsing

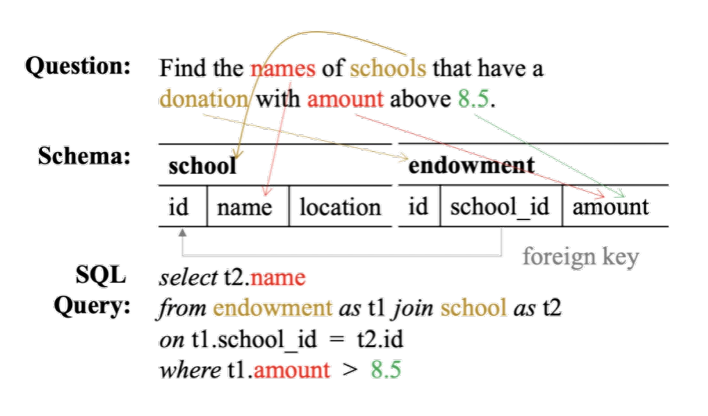


[Wang et al. 2020]

Previous Approaches

- **Heuristic** 🛠️: Rely on high-quality lexicons or ad-hoc heuristic rules like n-gram matching. It often suffers from poor flexibility.

Lexicon	Semantics
Nations	`nation`
Borders	`next_to_1`
Publication date	`publication_time`
Who	`author`



[Lei et al. 2020]

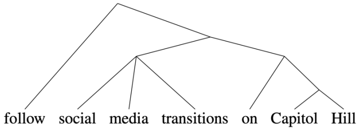
Previous Approaches

- **Supervised Learning** 💸: Collect grounding annotations as supervision. High modeling flexibility but requires expensive annotations of grounding.
- **Reinforcement Learning** ⏱️: Leverage rewards from semantic parsing models to train a grounding model. However, it is usually hard to train.

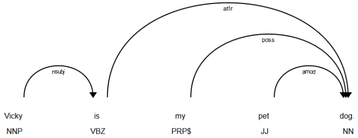
🧐 Can we learn a flexible grounding model end-to-end without grounding supervision?

When PLMs Meet Grounding

- Considering that PLMs have been shown to perform well in inducing syntactic structures, we hypothesize PLMs also excel at grounding.



Constituency Tree



Dependency Tree

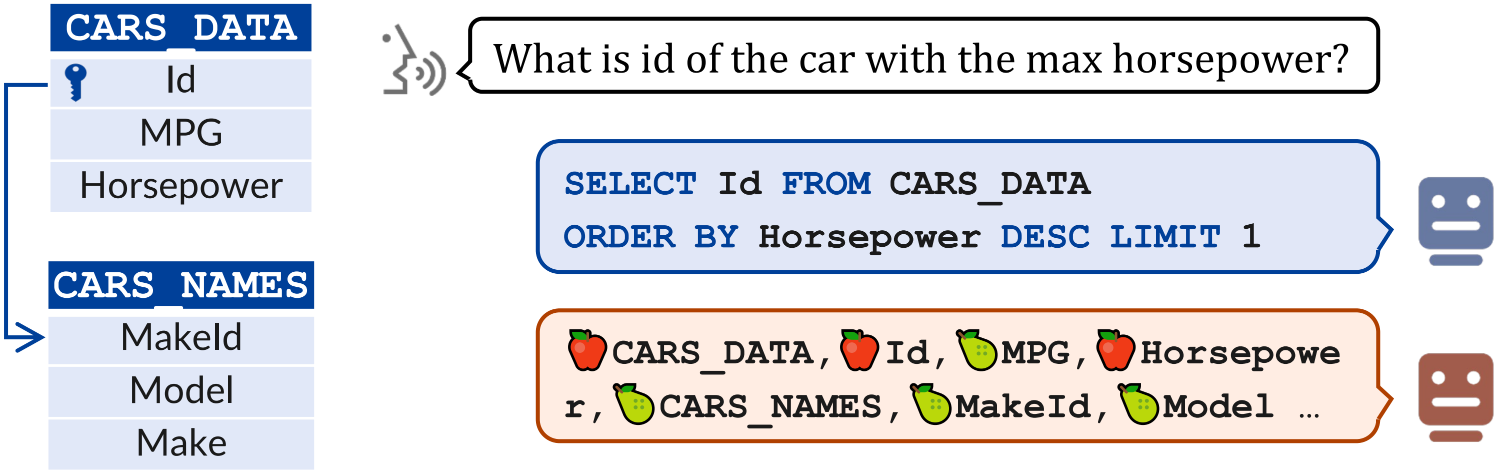
- Then the research question turns to be:

🔍 How to induce grounding from PLMs?

[Hewitt and Manning, 2019; Wu et al. 2020]

Step 1: Train An Auxiliary Module

- Train a PLM to identify whether a schema is mentioned in a question, whose supervision can be obtained from semantic parsing datasets.



[Liu et al. 2021]

Step 1: Train An Auxiliary Module

- Given a question and a schema, the auxiliary module can be used to give the model confidence of a schema being mentioned in a question.

Question	Schema	Confidence
How many total games were at braly stadium	Venue	0.92

Step 2: Erasing Tokens ✂️

- Erasing "How" and feed the erased question into the module to obtain the confidence, and calculate the confidence difference after erasing.

Question	Schema	Confidence
How many total games were at braly stadium	Venue	0.92

Erased Question	Schema	Confidence
~~How~~ many total games were at braly stadium	Venue	0.91

Difference
0.01

Step 2: Erasing Tokens✂️

- By erasing tokens in the question one by one, we can find which tokens are important for the module when predicting the schema "Venue".

Question

How many total games were at braly stadium

0.01	0.03	0.05	0.12	0.04	0.11	0.20	0.47

Schema

Venue

Step 3: Awakening Latent Grounding⏰

- We obtain a question-to-schema matrix after repeating the above over all schemas. Finally, the matrix can be used to train a standalone grounding model.

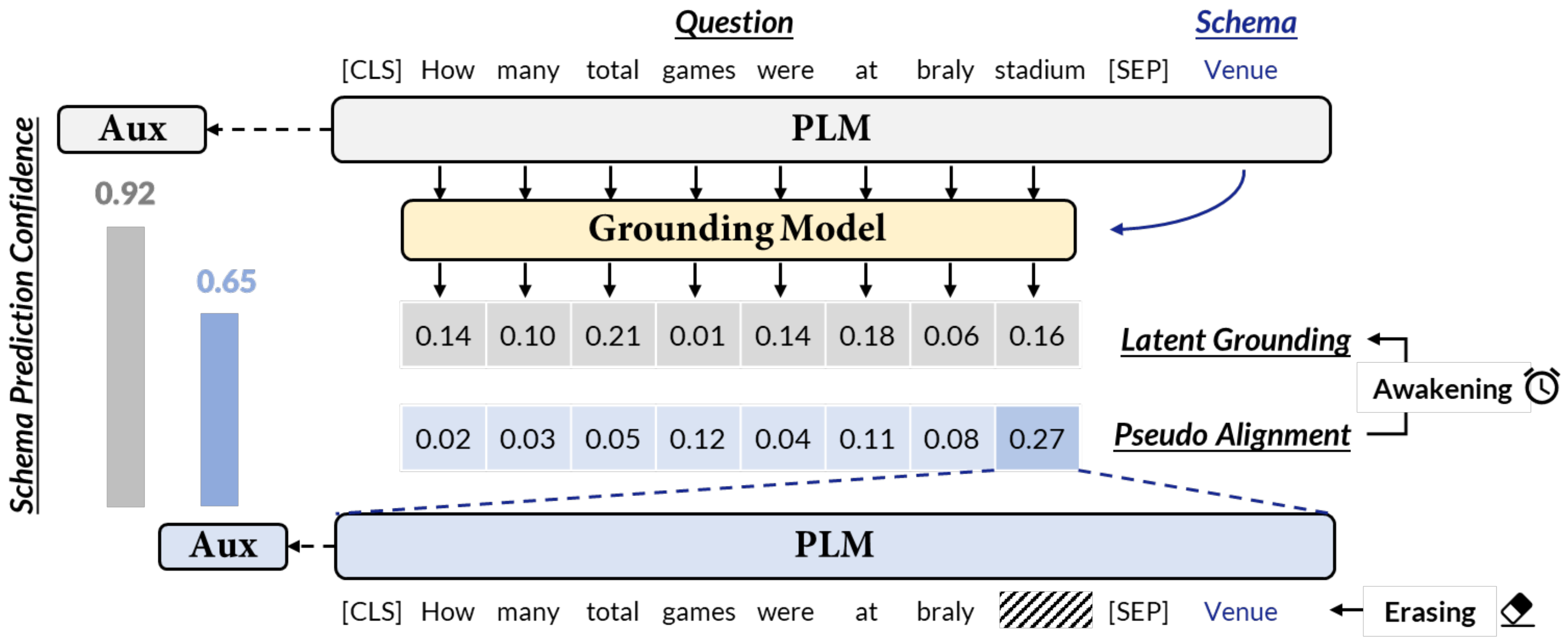
Question

How many total games were at braly stadium

Schema	How	many	total	games	were	at	braly	stadium
Venue	0.01	0.03	0.05	0.12	0.04	0.11	0.20	0.47
Game	0.01	0.01	0.02	0.64	0.00	0.02	0.01	0.04
Barly	0.01	0.02	0.01	0.02	0.01	0.12	0.74	0.32

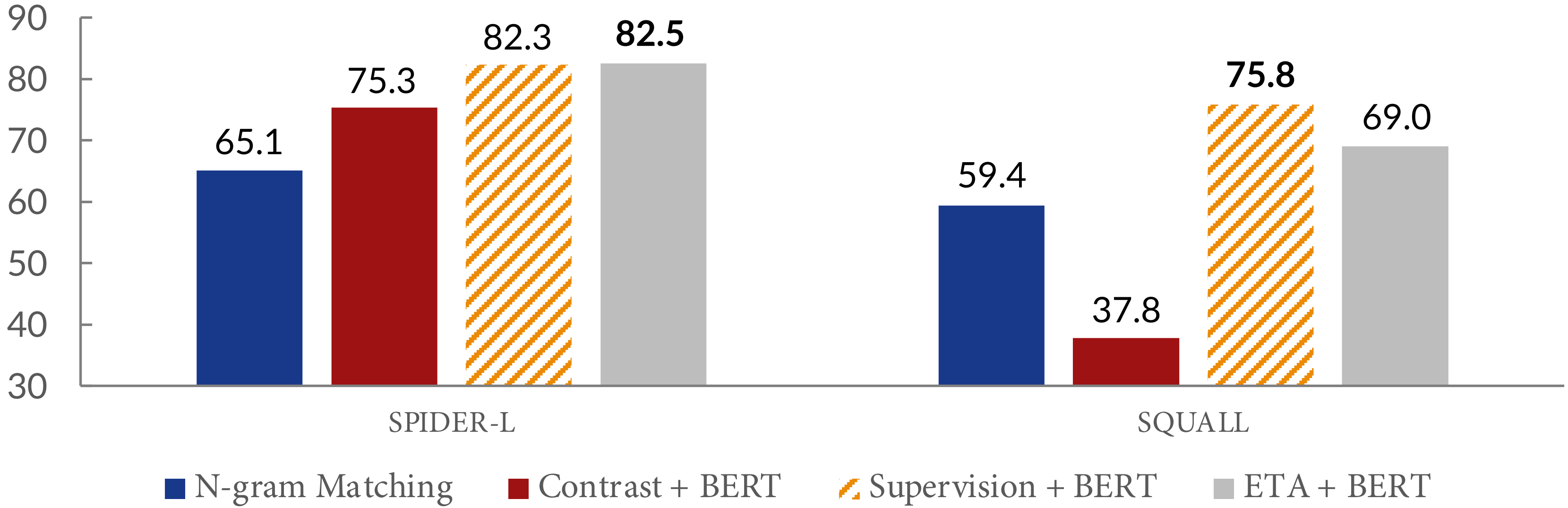
Supervise → **Grounding Model**

Overview: Erasing-then-Awakening (ETA)



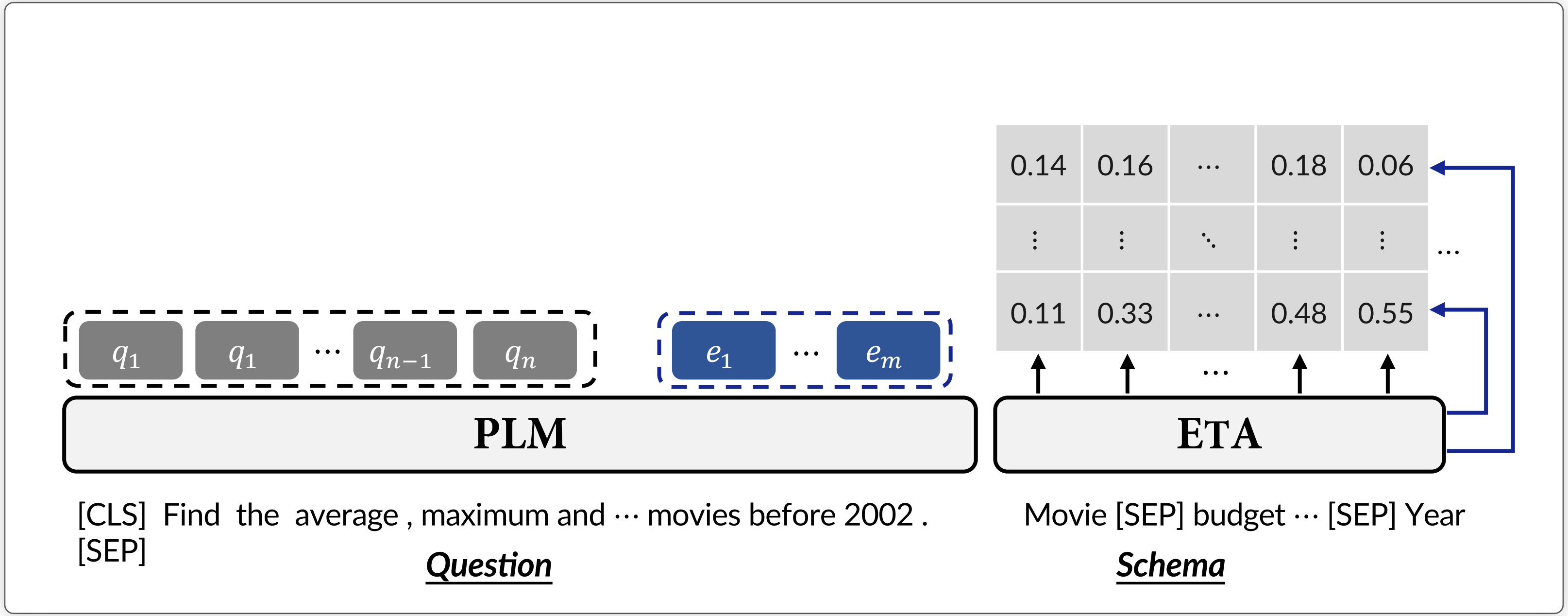
Results: Entity Grounding

- Although not trained under fine-grained grounding supervision, our model ( ) is comparable with or slightly worse than the supervised model ( ).



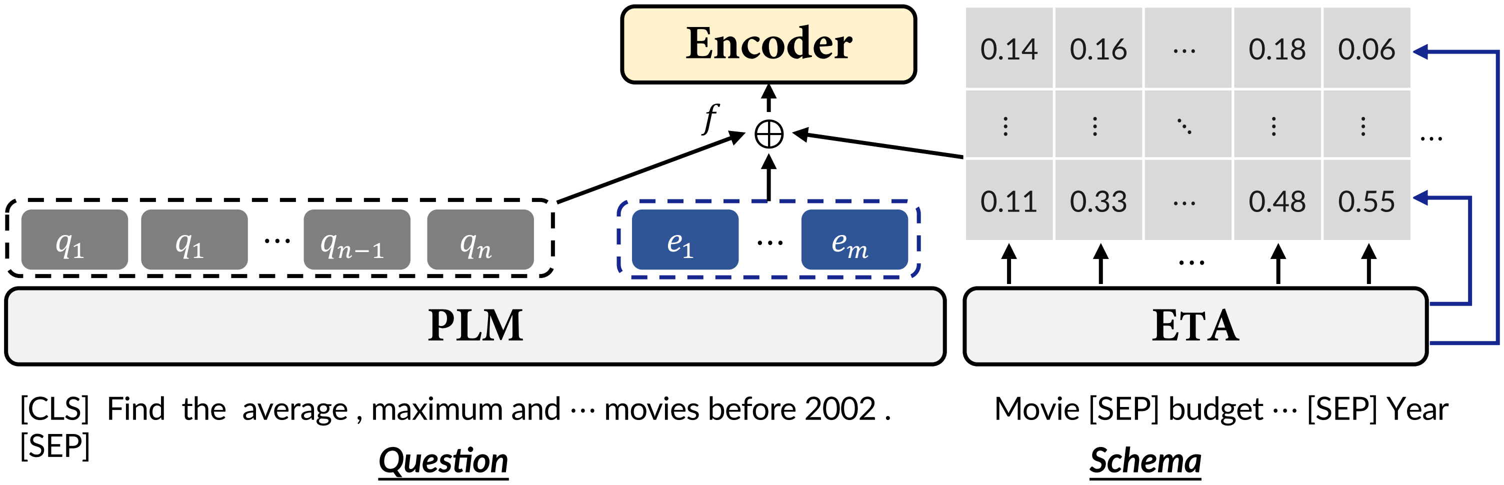
Couple with Semantic Parser

- The grounding model could couple with semantic parsers as the schema prior.



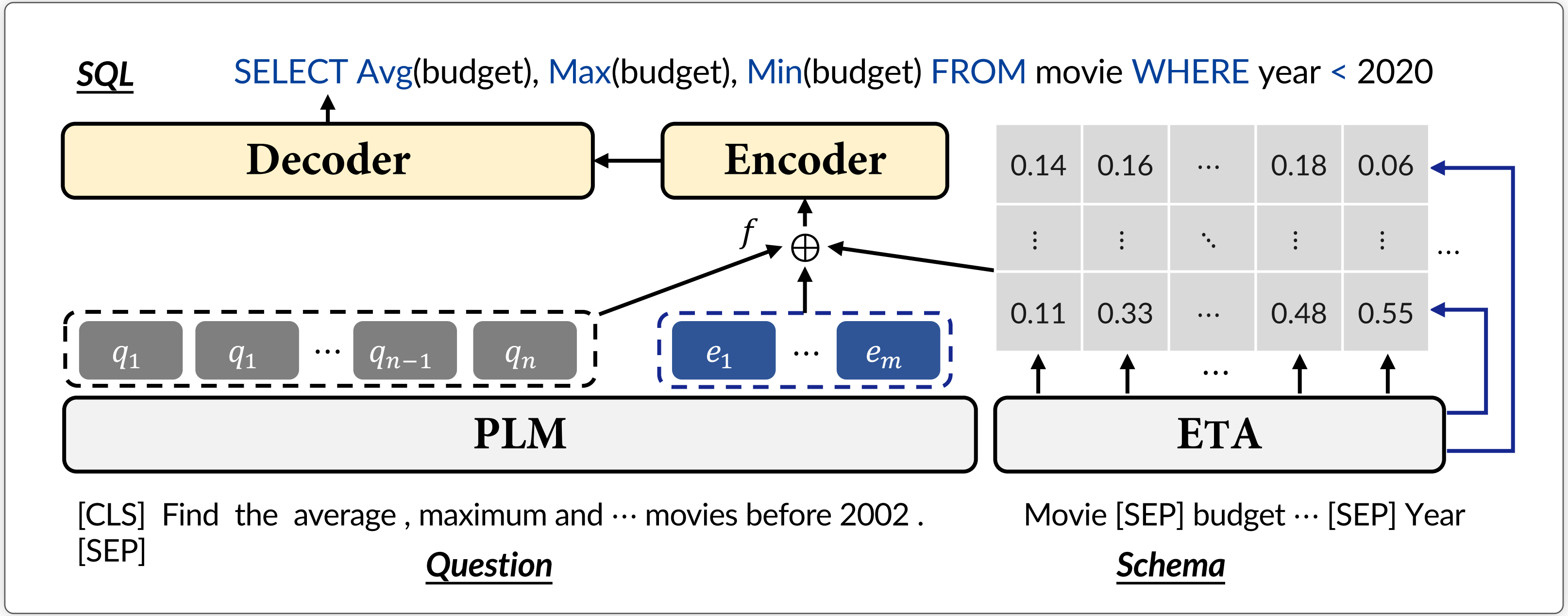
Couple with Semantic Parser

- Concretely, the parser leverages grounding to produce schema-enhanced question representations.



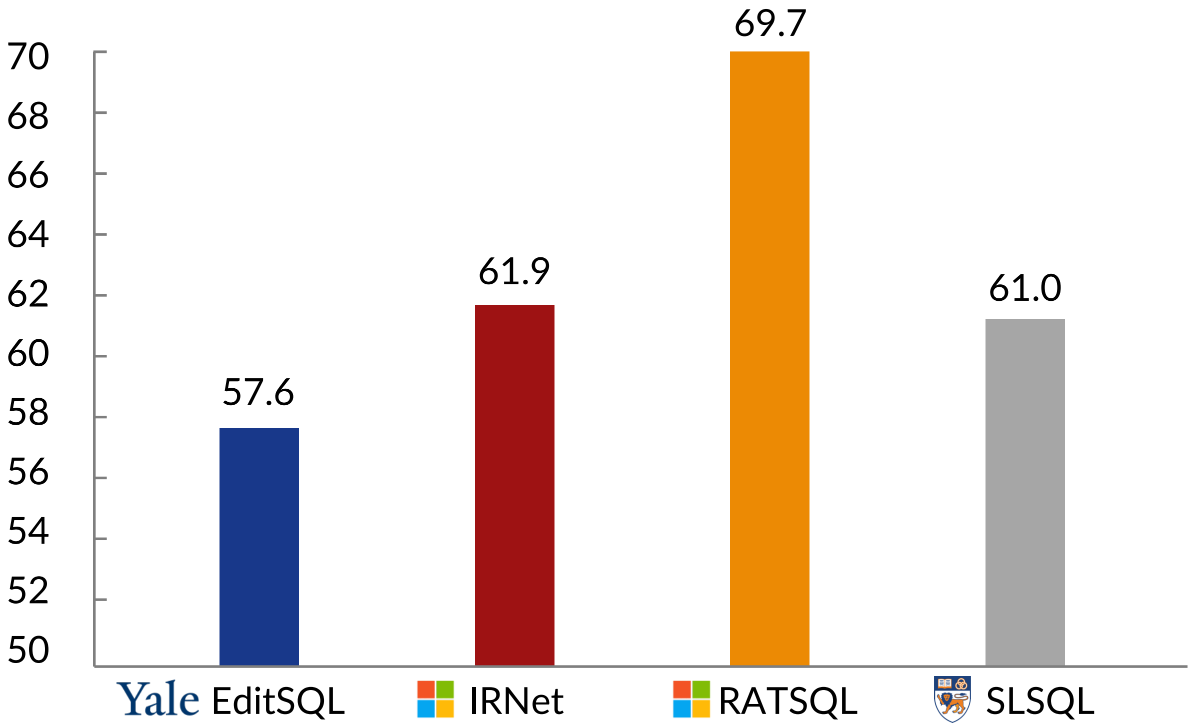
Couple with Semantic Parser

- The decoder predicts the SQL by attending on the schema-enhanced states.



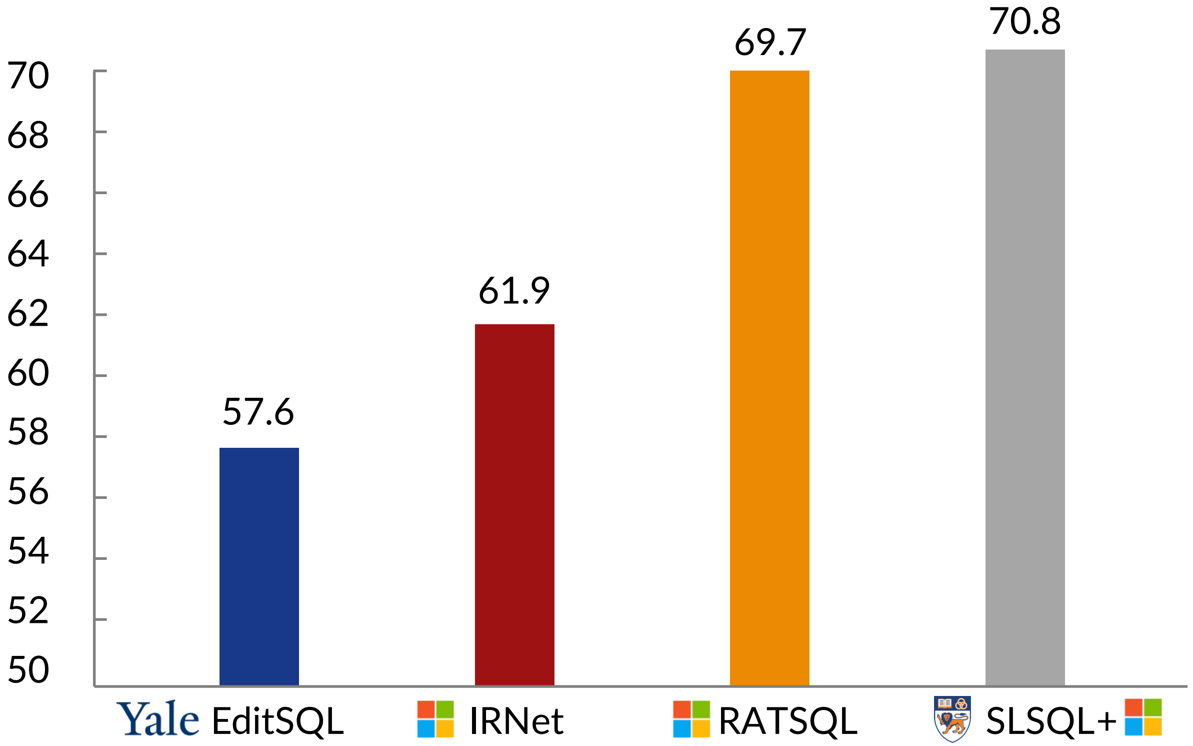
Result: Semantic Parsing

- ETA boosts existing parsers significantly (Spider, WTQ) with good interpretability.



Result: Semantic Parsing

- ETA boosts existing parsers significantly (Spider, WTQ) with good interpretability.



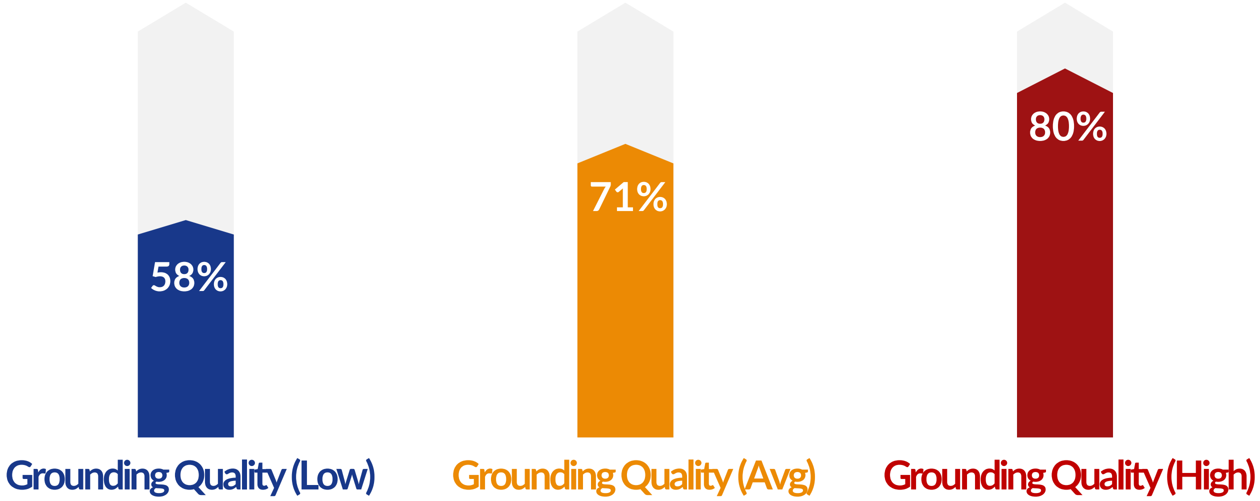
Towards Interpretable Parser

- Grounding provide us a useful guide to understand a semantic parser.

Question with Alignment	SQL with Alignment
1. Show name_1, country_2, age_3 for all singers_4 ordered by age_5 from the oldest_5 to the youngest_5.	SELECT name_1, country_2, age_3 FROM singer_4 ORDER BY age_5 DESC
2. Where_1 is the youngest_2 teacher_3 from?	SELECT hometown_1 FROM teacher_3 ORDER BY age_2 ASC LIMIT 1
3. For each semester_1, what is the name_2 and id_3 of the one with the most students registered_4?	SELECT semester_name_2, semester_id_3 FROM semesters_1 JOIN $\text{student\_enrolment}_4$ ON semesters.semester_id = student_enrolment.semester_id GROUP BY semester_id_3 ORDER BY COUNT(*) DESC LIMIT 1

Towards Interpretable Parser

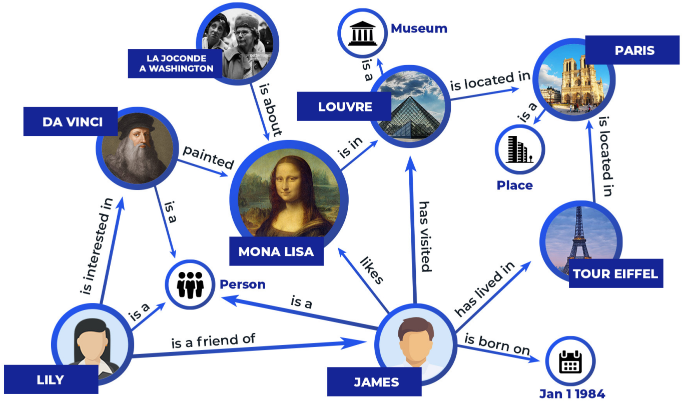
- The parsing performance highly depends on the grounding. When the grounding quality is high, the predicted SQL correctness can be up to 80.17%.
- It opens a way to perform hot fixes on semantic parsers via lexicons.



Insight: Knowledge from Pre-trained Language Model

Pre-trained language models may be good knowledge bases, and we can try to exploit these knowledge in a more explicit way.

≈



Unleashing the Power of Pre-trained Language Models(PLMs)

Part 1: Towards Interpretable Parser

Grounding power of PLMs helps build interpretable semantic parsers.

Part 2: Towards Cost-Effective Parser

Learning latent program helps build cost-effective semantic parsers.

Part 3: Towards Synthesized Parser

Generalizing power of PLMs helps build synthesized semantic parsers.

Liu Q et al, TAPEX: Table Pre-training via Learning a Neural SQL Executor. **Under Review.**

Strong Supervision ⚔️ Weak Supervision

Strong Supervision

What is id of the car with the max horsepower?

```
SELECT Id FROM CARS_
DATA ORDER BY Horsep
ower DESC LIMIT 1
```

Weak Supervision

Greece held its last Summer Olympics in which year?

```
SELECT Year FROM Oly
mpics ORDER BY Year
DESC LIMIT 1
```

2004

[Pasupat and Liang et al. 2015]

Strong Supervision ⚔️ Weak Supervision

Strong Supervision

- **Annotation**: domain experts who can write programs.

- **Cost of Data**: 1700 USD for <3K python code [Yin et al. 2018].
- **Performance**: programs as prior for understanding questions.

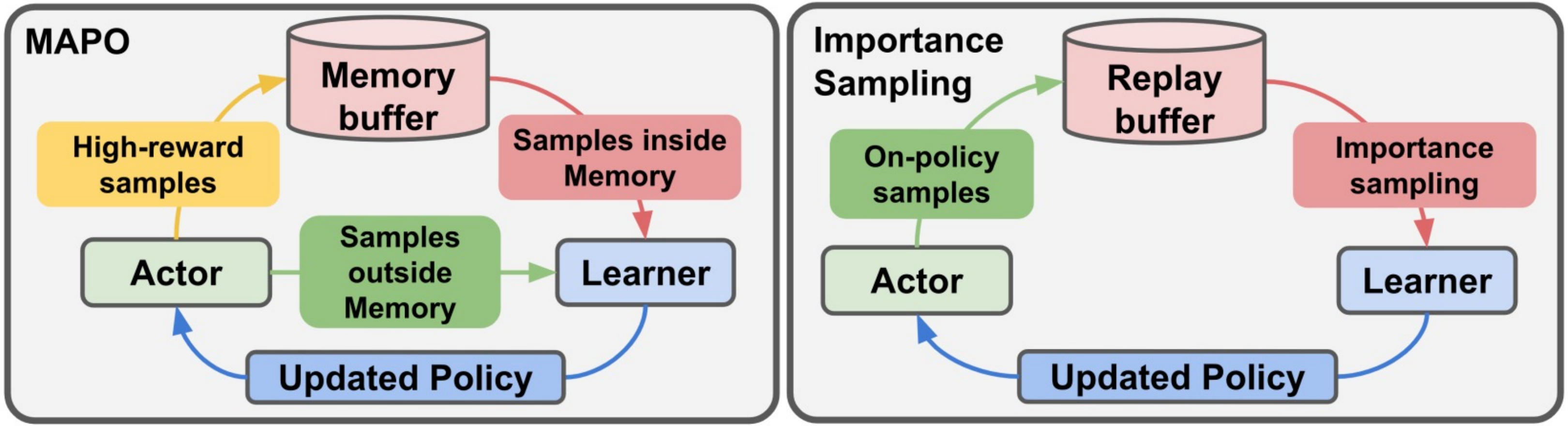
Weak Supervision

- **Annotation**: users who want to seek information.

- **Cost of Data**: <0.1 USD per answer [Reddy et al.2018].
- **Performance**: suffer from the issue of spurious programs.

Previous Approaches

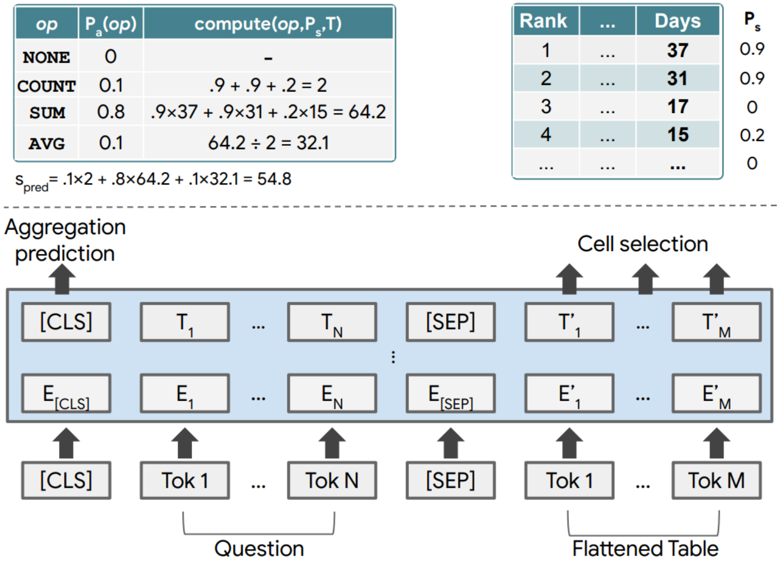
- **Reinforcement Learning**: Obtain rewards by comparing execution results with golden answers to train a semantic parser. Hard to scale to complex scenarios.



[Chen et al. 2018]

Previous Approaches

- **Table Parsing**: Predict answer by selecting table cell values and optionally applying an aggregation operator to the selected region. Flexibility is limited.

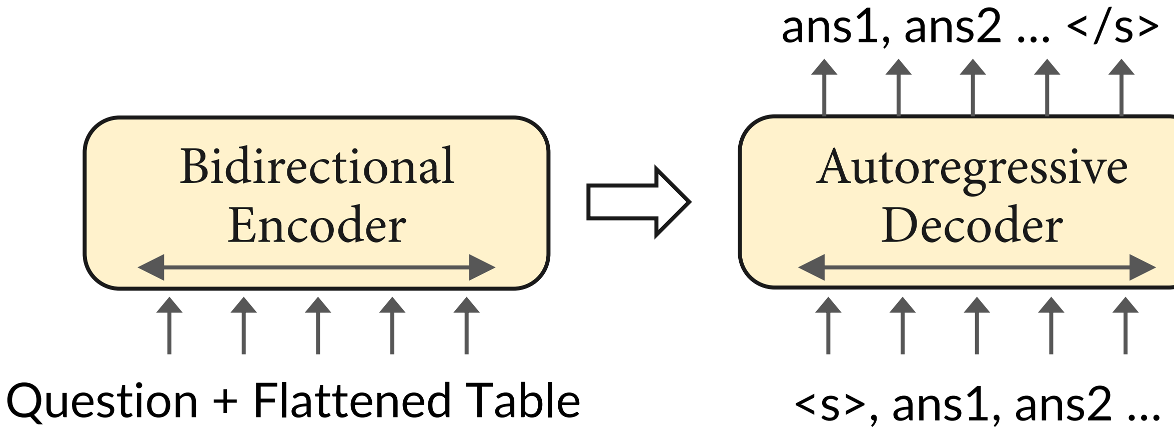


[Herzig et al. 2020]

Method: Generative Weakly Semantic Parsing

We formulate the task of weakly-supervised semantic parsing as answer generation, and leverages generative PLMs (e.g., BART) to output autoregressively.

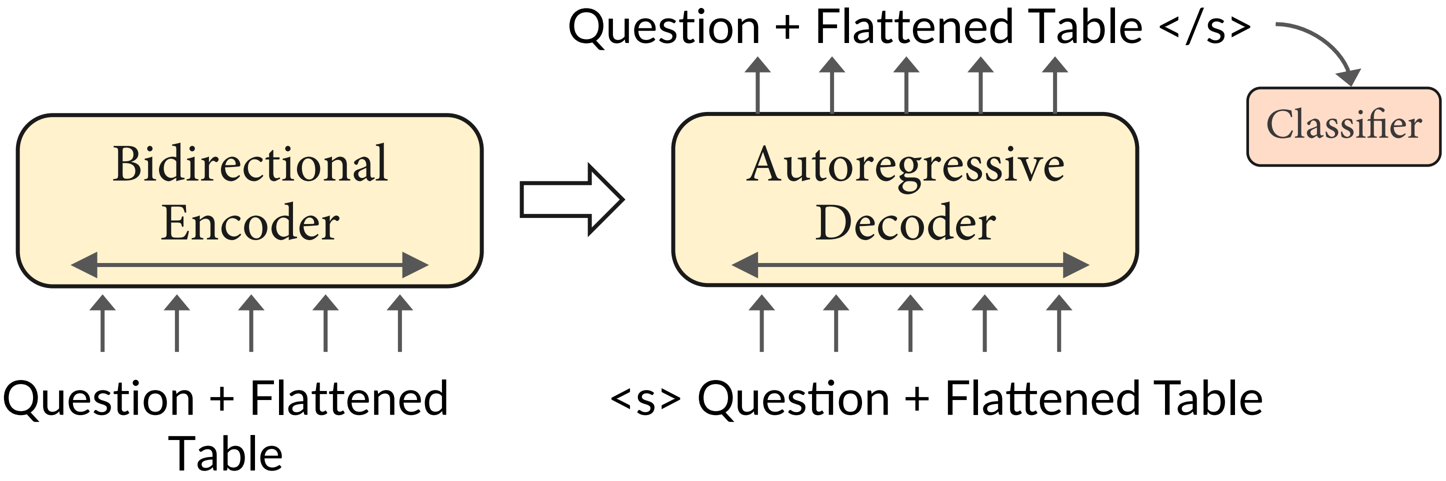
Table-based Question Answering



Method: Generative Weakly Semantic Parsing

We formulate the task of weakly-supervised semantic parsing as answer generation, and leverages generative PLMs (e.g., BART) to output autoregressively.

Table-based Fact Verification



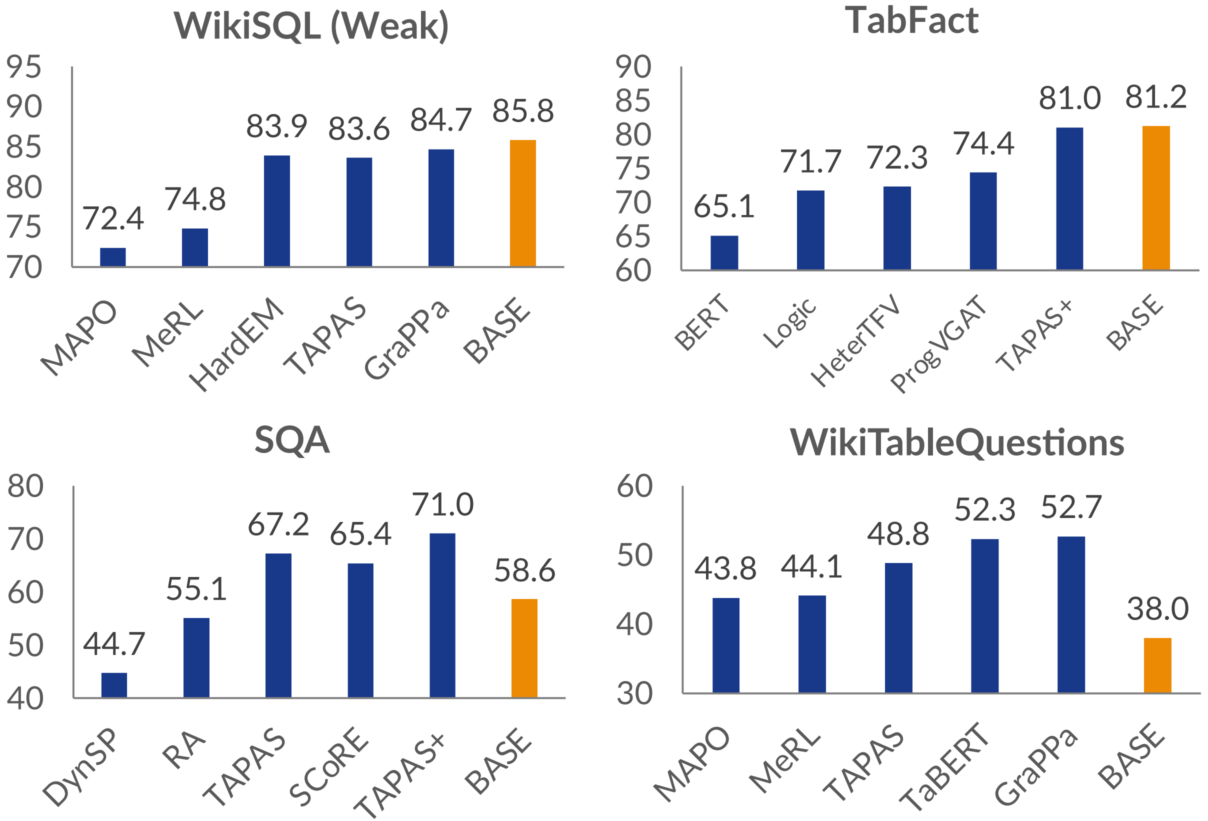
Preliminary: Models Are Still Data-Hungry

Flexibility

Flexible to adapt to different tasks, including simple QA, complex QA, dialogue QA, table fact verification.

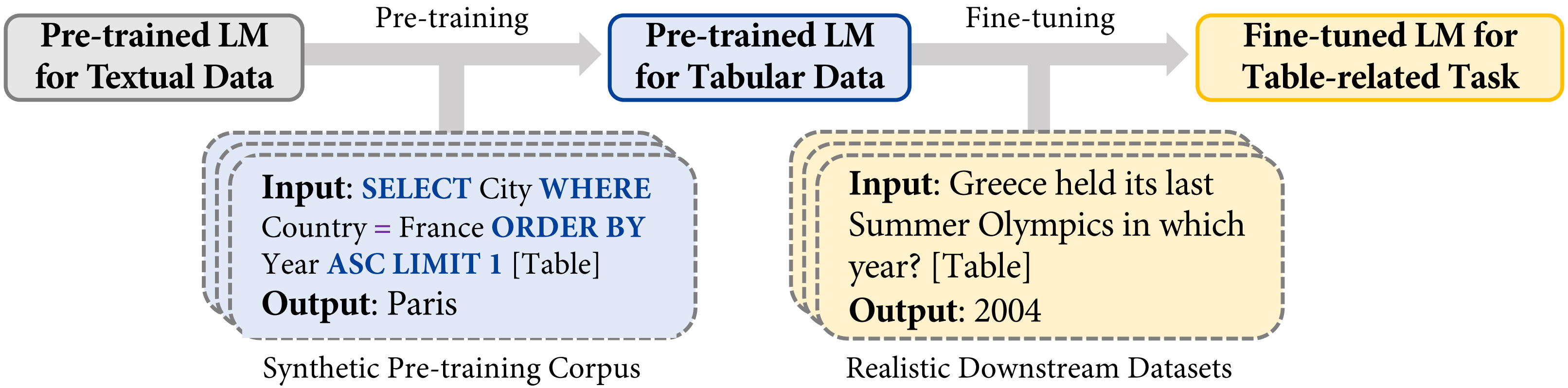
Data-Hungry

Relatively small data amount (SQA and WikiTableQuestions) result in large performance degradations.



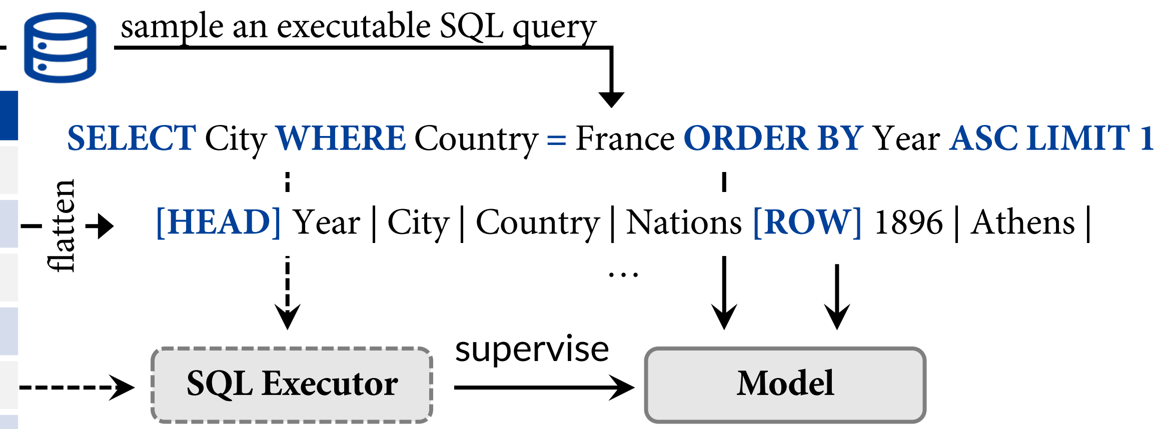
TAPEX: Execution 🚀 Continual Pre-training

Continually pre-train a model to mimic the behavior of a symbolic execution engine.



Method: Latent Program Learning via Execution

If we train a model to mimic the SQL query execution procedure over databases, we believe it learns latent program from the execution engine.



Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
…	…	…	…
2004	Athens	Greece	201
2008	Beijing	China	204

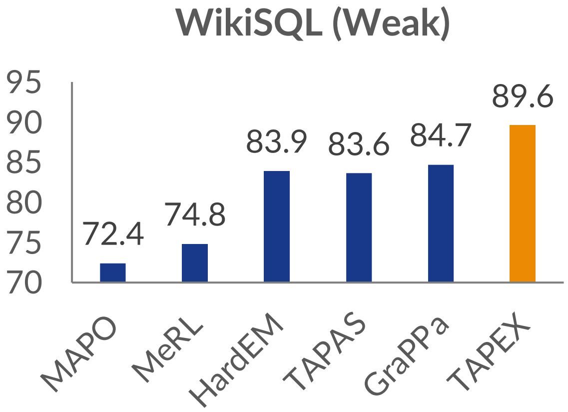
Result: Execution Unleashes the Power of Task Data

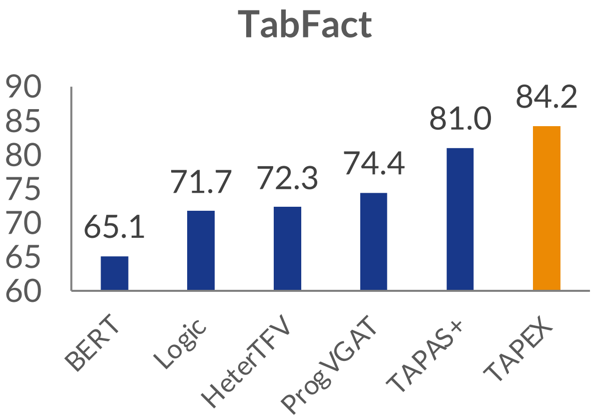
Generality

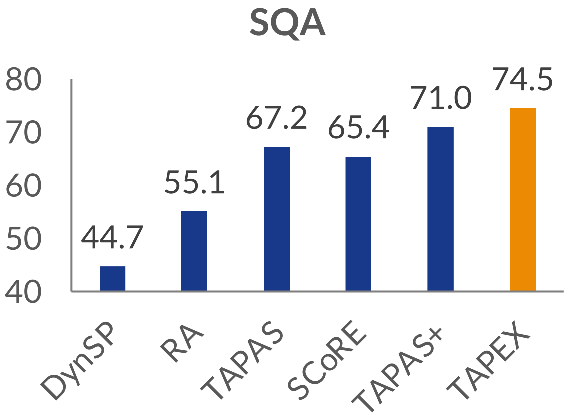
TAPEX broadly improves the model under the only task supervision.

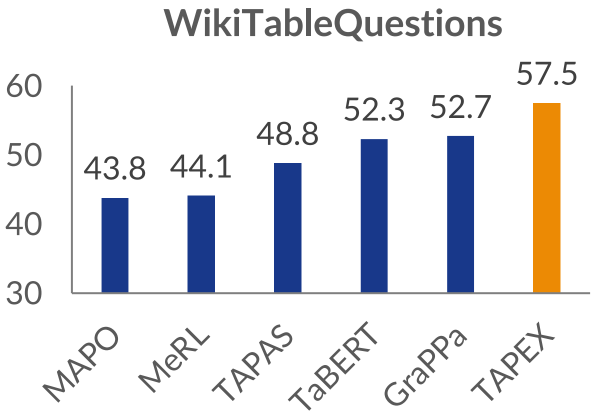
Competitiveness

Our model beats against those baselines with program supervision on two datasets.

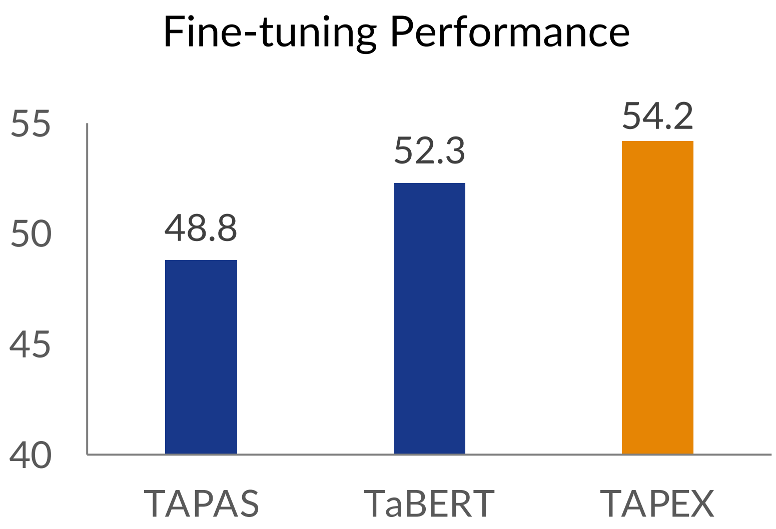


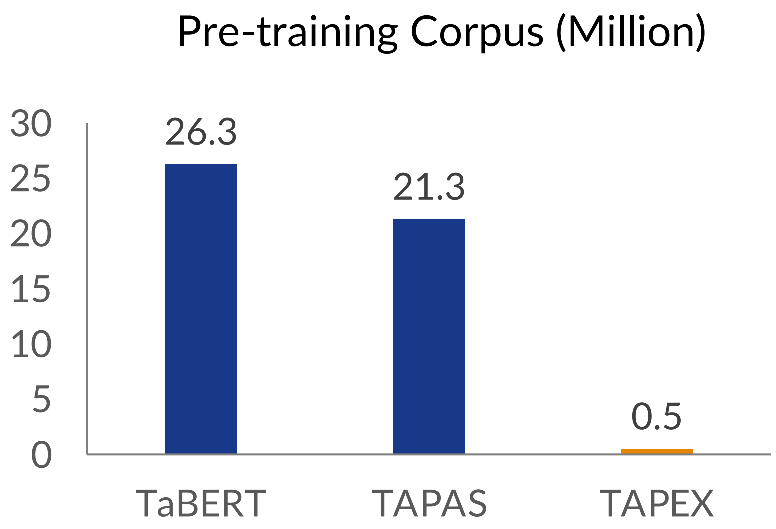






Result: Cost-Effective Semantic Parsing





Compared with TaBERT, **2%** of corpus yields **2%** improvement!

Insight: The Choice of Pre-training Corpus

When performing continual pre-training, instead of mining a large noisy corpus, we can also try to synthesize an accurate and small corpus.

Insight: The Choice of Pre-training Corpus

When performing continual pre-training for NLP tasks, instead of natural language, we can also try to leverage programs.

> We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

```
def start_link(name, opts \\ []) do
  GenServer.start_link(__MODULE__, {name}, opts)
end

def init({name}) do
  require Logger
  Logger.log :debug, "Started channel #{name}!"
  :pg2.join(:channels, self)
  :ets.insert(:channels, {name, self})
  users = :ets.new(:users, [:set, :protected])
  {:ok, {name, users, []}}
end

def handle_call({:send, message}, _from, {name, _users, _buffer} = state
  Kaguya.Util.sendPM(name, message)
  {:reply, :ok, state}
end

def handle_call({:rename_user, {old_nick, new_nick}}, _from, {_name, use
  case :ets.lookup(users, old_nick) do
    [{^old_nick, user}] ->
      new_user = %{user | nick: new_nick}
      :ets.delete(users, old_nick)
      :ets.insert(users, {new_nick, new_user})
    [] -> :ok
  end
  {:reply, :ok, state}
end
```

Unleashing the Power of Pre-trained Language Models(PLMs)

Part 1: Towards Interpretable Parser

Grounding power of PLMs helps build interpretable semantic parsers.

Part 2: Towards Cost-Effective Parser

Reasoning power of PLMs helps build cost-effective semantic parsers.

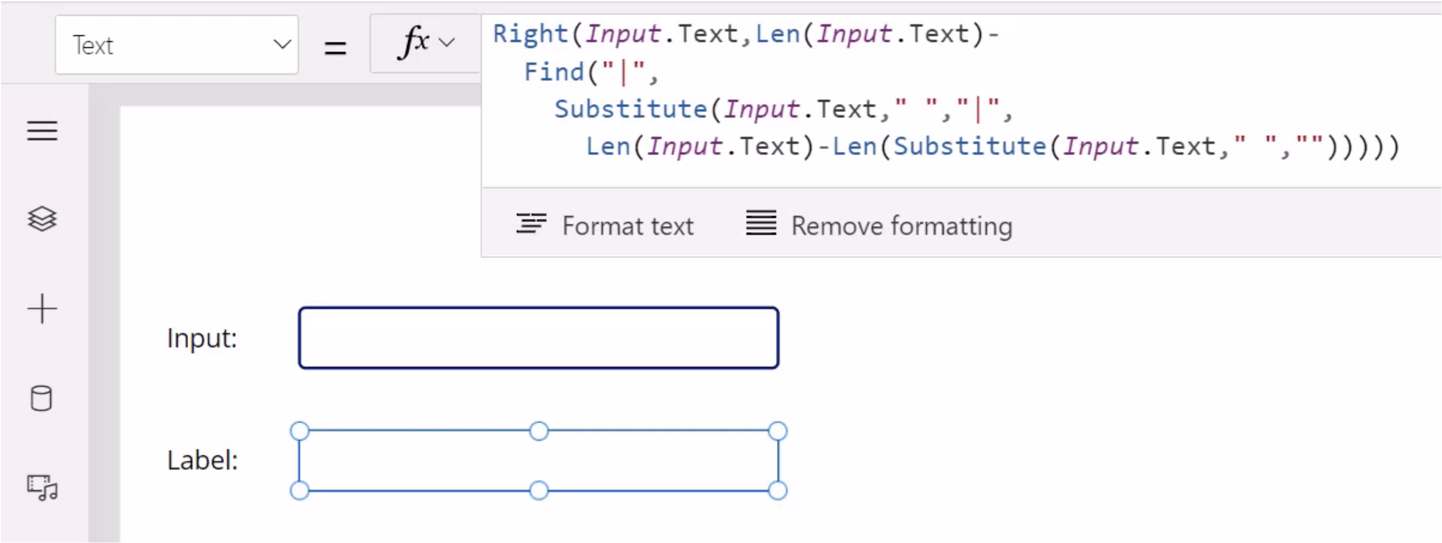
Part 3: Towards Synthesized Parser

Generalizing power of PLMs helps build synthesized semantic parsers.

Power Apps Ideas: AI-Powered Assistance Now helps Anyone Create Apps Using Natural Language. **Microsoft Build 2021.**

Background: Task-specific Datasets

- For popular tasks (e.g., text-to-SQL), there are a lot of public datasets.
- For specific tasks (e.g., text-to-Formula), data collection is difficult 😵 and costly 💰.



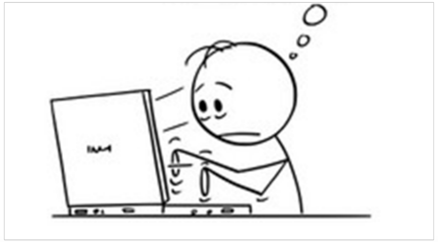
Background: Task-specific Datasets

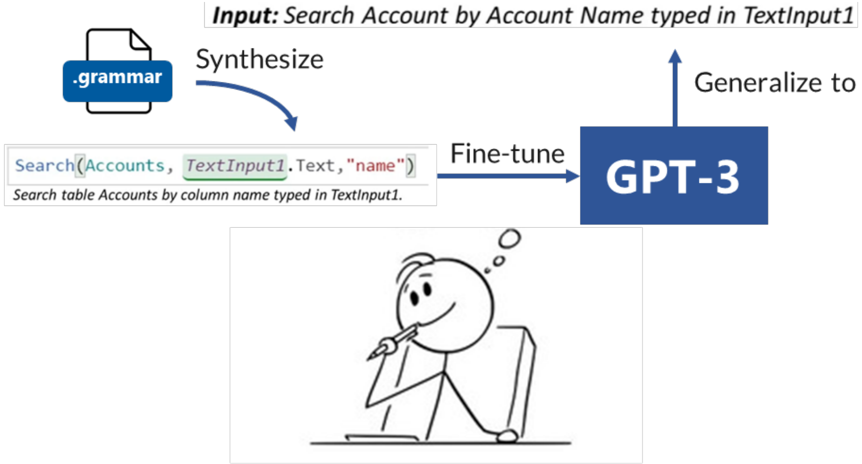
- How about fine-tuning PLMs with only synthetic data?

Input: *Search Account by Account Name typed in TextInput1*

Output:

```
Search(Accounts, TextInput1.Text,"name")
```



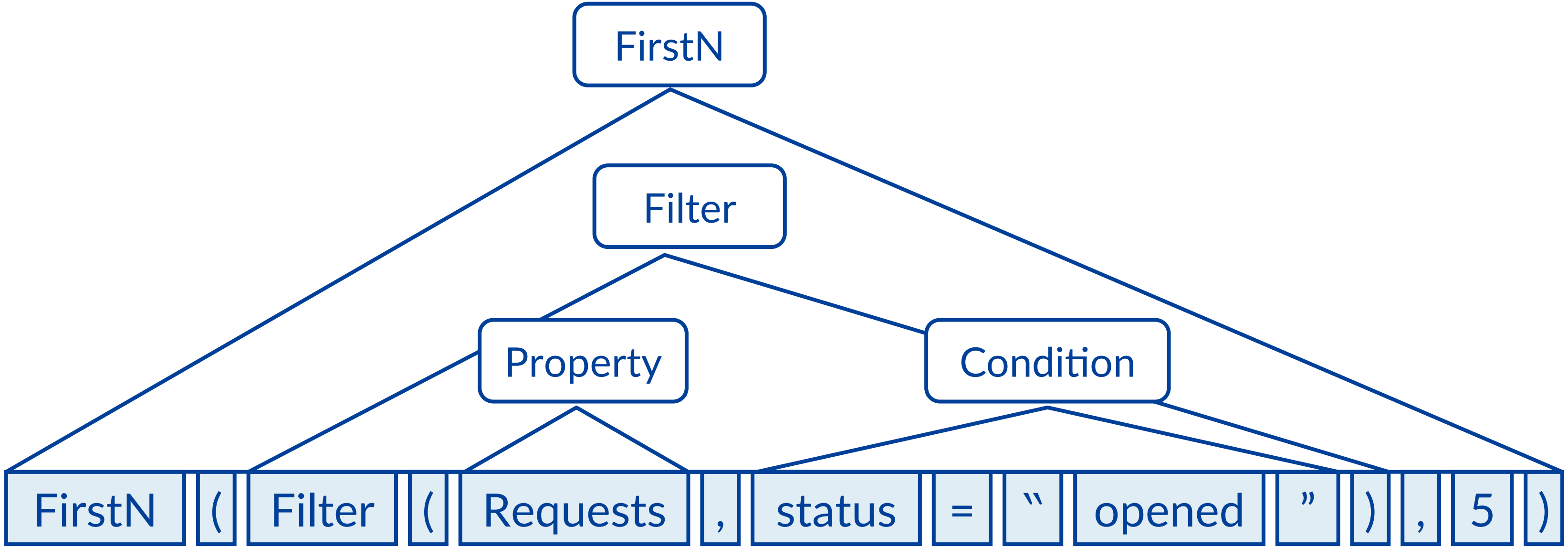


Data annotation is extremely costly.

PLMs bridge the gap between synthetic and real data.

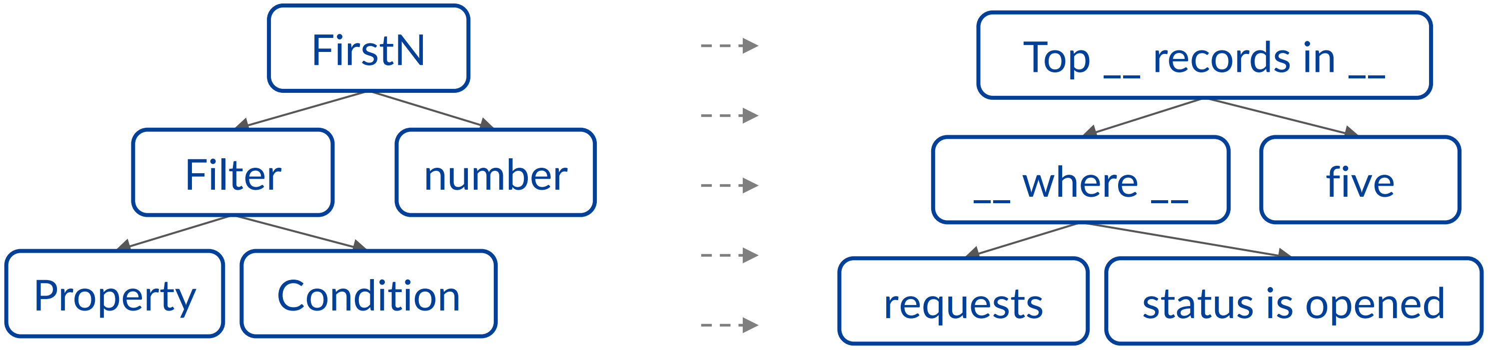
Motivation: Program Modularity

Program: FirstN (Filter (Requests, status="opened"), 5)



Motivation: Program Modularity

Program: FirstN (Filter (Requests, status="opened"), 5)



Sentence: Top five records in requests where status is opened

Towards Synthesized Parser

Compositional SCFG

- ❑ Filter ({0:DataSource},{1:Condition})
  ↔ {0} where {1}
- ❑ FirstN ({0:DataSource},{1:Number})
  ↔ first {1} records in {0}
- ❑ …

Synthetic MRs

- ❑ Filter(IceCream, OnOrder > 0)
- ❑ FirstN(Filter(Requests, status = “opened”),5)
- ❑ …

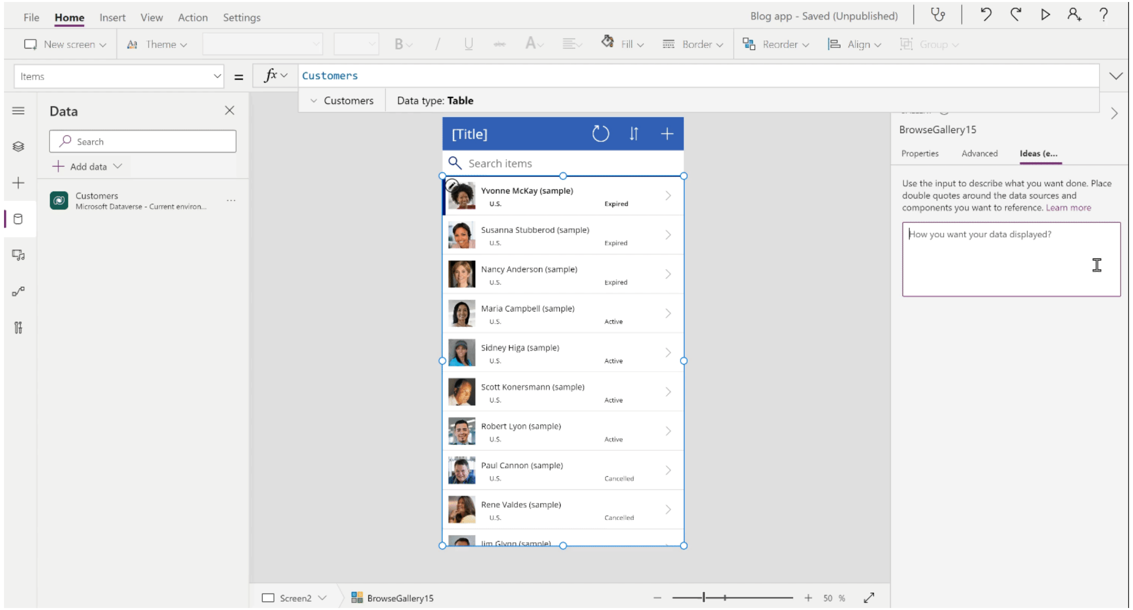
Synthetic Data

Filter(IceCream, OnOrder > 0)
=> IceCream where OnOrder is larger than 0

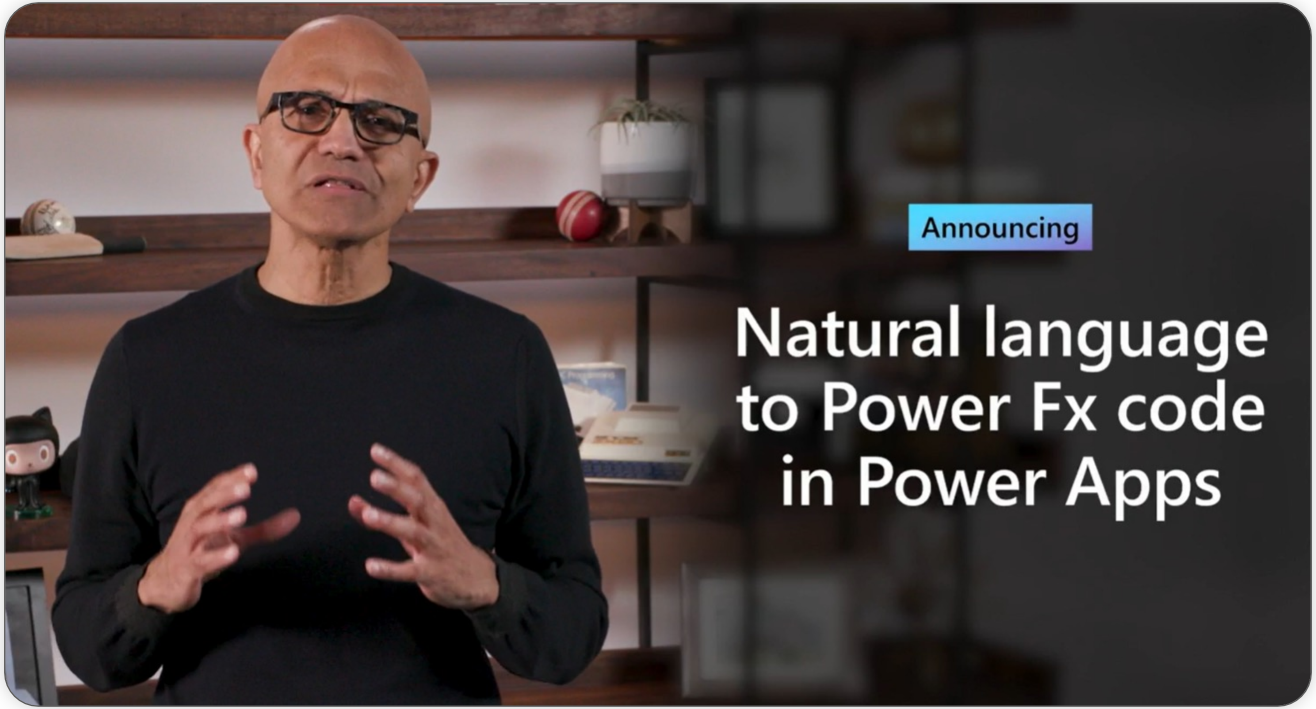
FirstN(Filter(Requests, status = “opened”), 5)
=> First 5 records in Requests where status is “opened”

Result: Synthetic Data Proxy for Real Data

XY.z% test accuracy on natural queries in **1** week



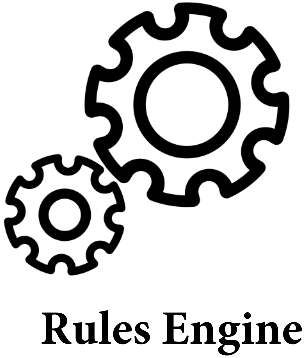
Result: Integration Into PowerApps



Insight: Injecting Rules into BIG Models

Rules can be injected into BIG models in a data-driven way, with (probably) not hurting its linguistic generalization capability.

+



Take Away

Part 1: Towards Interpretable Parser

We can awaken grounding from PLMs to build interpretable semantic parsers.

Part 2: Towards Cost-Effective Parser

We can learn latent program inside models via execution pre-training.

Part 3: Towards Synthesized Parser

We can leverage PLMs to perform unsupervised semantic parsing.

Thanks & QA

Qian Liu (刘乾)
qian.liu@buaa.edu.cn
Joint Ph.D. of BUAA and MSRA