Towards Scalable Schema Mapping using Large Language Models

Christopher Buss*, Mahdis Safari*,
Arash Termehchy, David Maier, Stefan Lee

*Equal Contributors

Portland State University

Oregon State University

[Rare Daily Staff 2023] & [Ashburn & Thor 2004]

Based on True Events: Drug Repositioning Saves Lives

Biomedical Researcher

Patients with Castleman's disease **(Rare disease)**

- Potentially fatal: causes severe inflammation
  - Shuts down major organs
- No effective treatments currently exist

Must do something!

Unfortunate reality:

Too rare: no financial incentive for companies to develop treatments

Alternative:

Find an existing drug to treat Castleman's disease

[Rare Daily Staff 2023] & [Ashburn & Thor 2004]

Consult a Reference Datasource

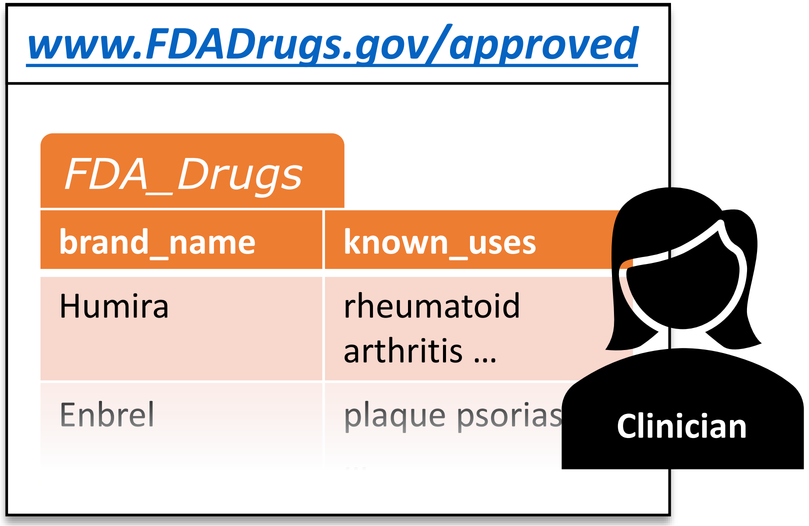
www.FDADrugs.gov/approved

FDA_Drugs

brand_name	known_uses
Humira	rheumatoid arthritis …
Enbrel	plaque psorias…

[Rare Daily Staff 2023] & [Ashburn & Thor 2004]

Identify a Candidate Drug



Castleman's causes severe inflammation…

Humira is used to treat conditions involving severe inflammation

Candidate drug: Humira

STOP: can't just give Humira to patients!
Will it help or hurt?

Next step: gather more information about Humira

- Without making patients wait too long!

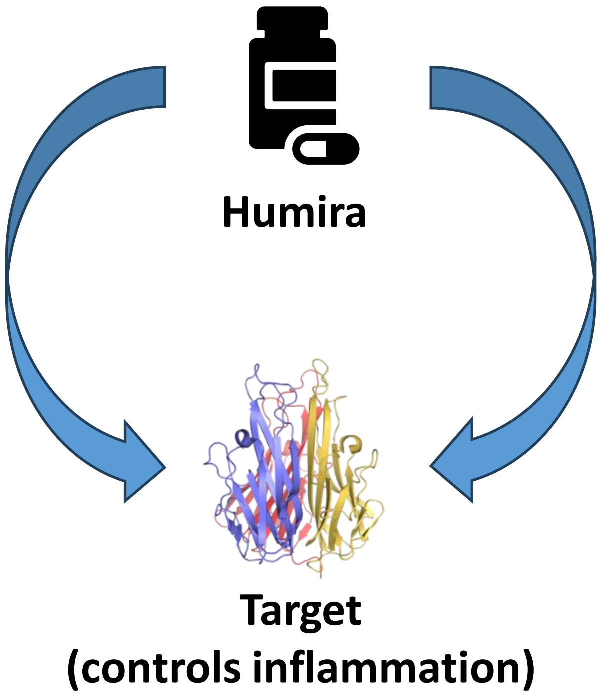
Need to connect data from many sources as quickly as possibly

- A lot of important things we need to know about Humira

Example: Humira’s Effects on Proteins?

Proteins: fundamental to core mechanisms of body

- Make sure Humira affects *correct* proteins in *correct* way



A side effect of some drugs

Avoid these drugs!

Induce

=

more Inflammation

Inhibit

=

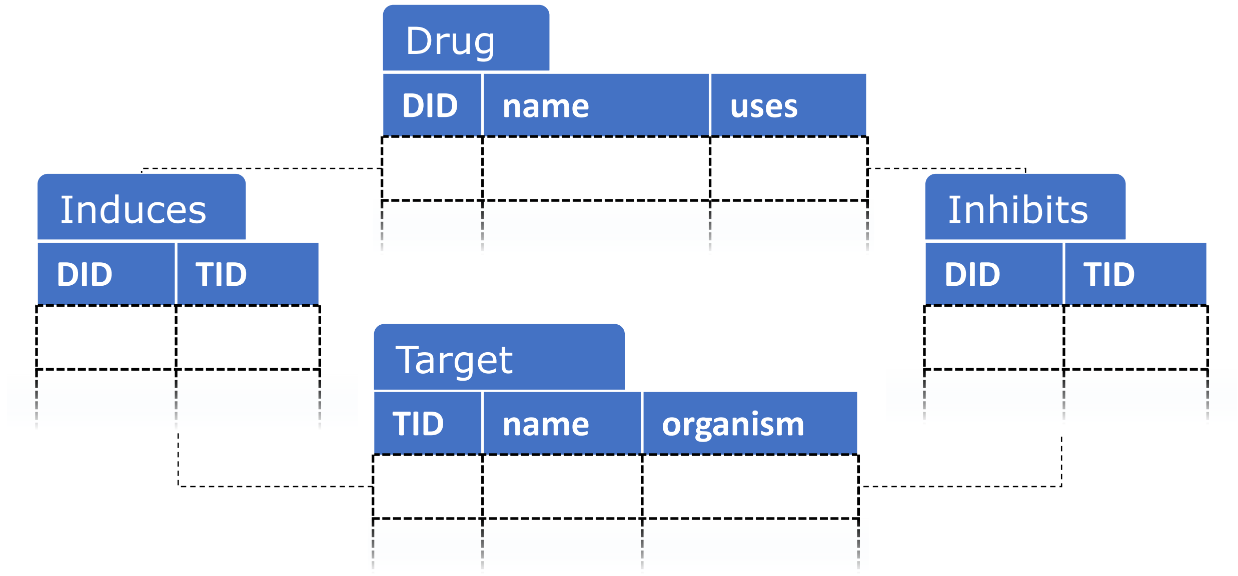
less Inflammation

What we want

Find these drugs!

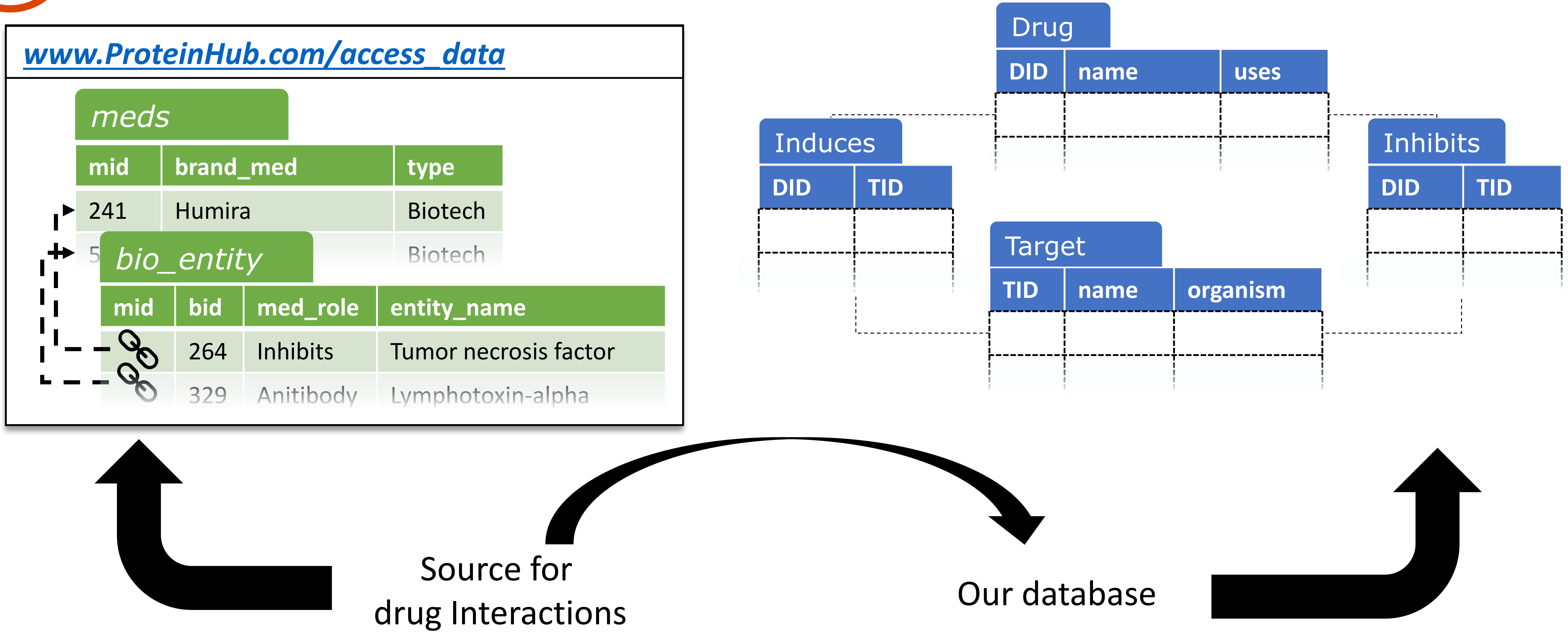
Create a database to <u>capture this information</u>

A Database for Drug-Protein Interaction



Populate **database** with information

Add Drug-Target Information



Write **mapping** to move data from **source** to our **database**

Map Drug Information

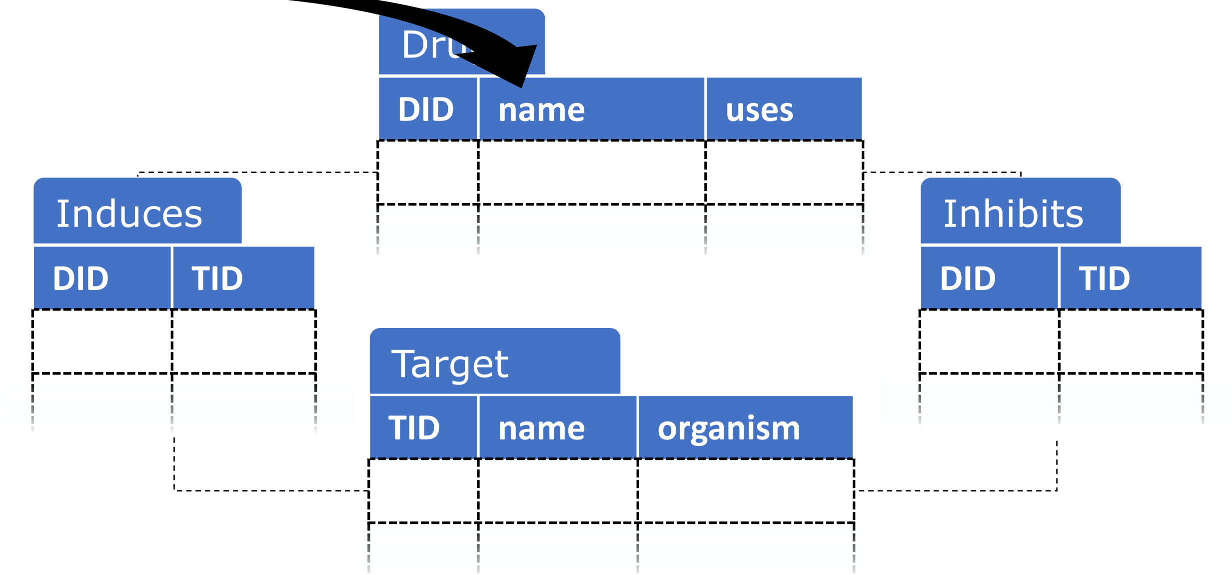
www.ProteinHub.com/access_data

meds

mid	brand_med	type
241	Humira	Biotech
5		Biotech

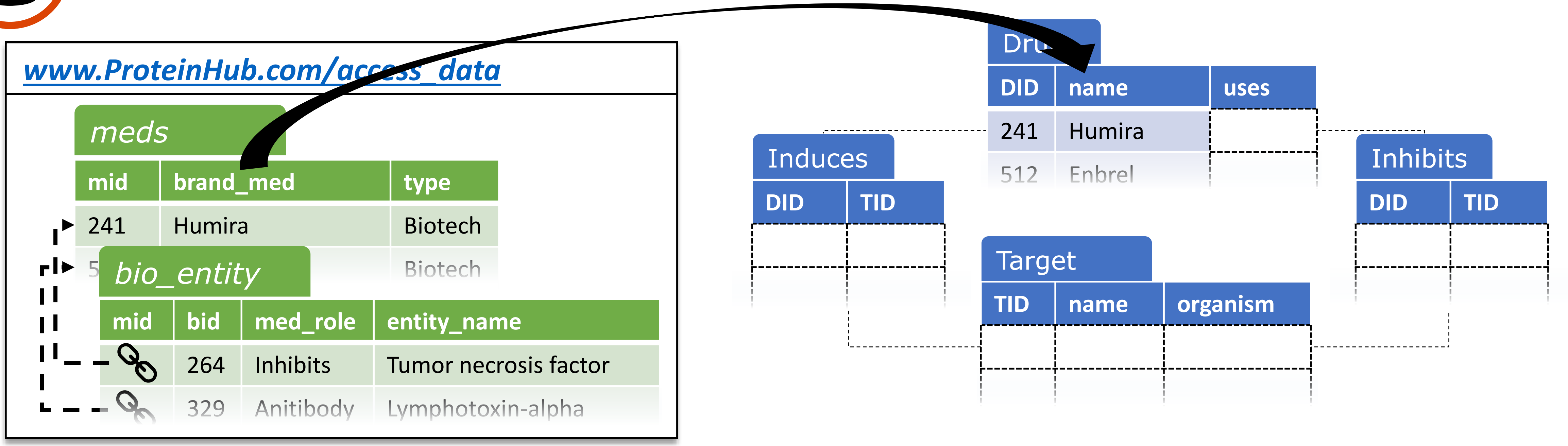
bio_entity

mid	bid	med_role	entity_name
	264	Inhibits	Tumor necrosis factor
	329	Anitibody	Lymphotoxin-alpha



Mapping:

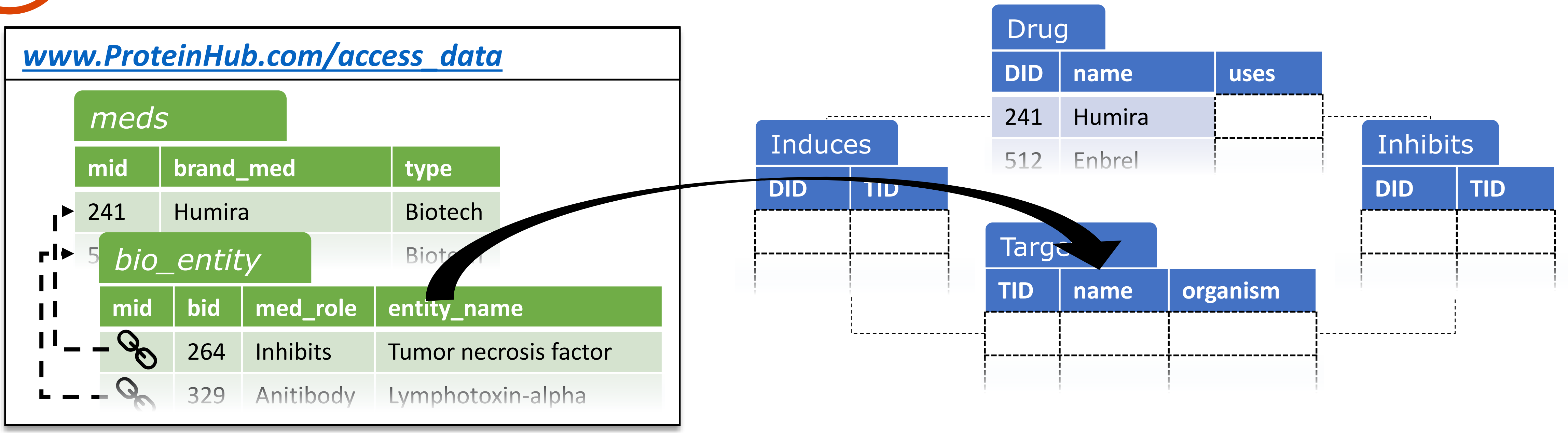
Map Drug Information



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).
```

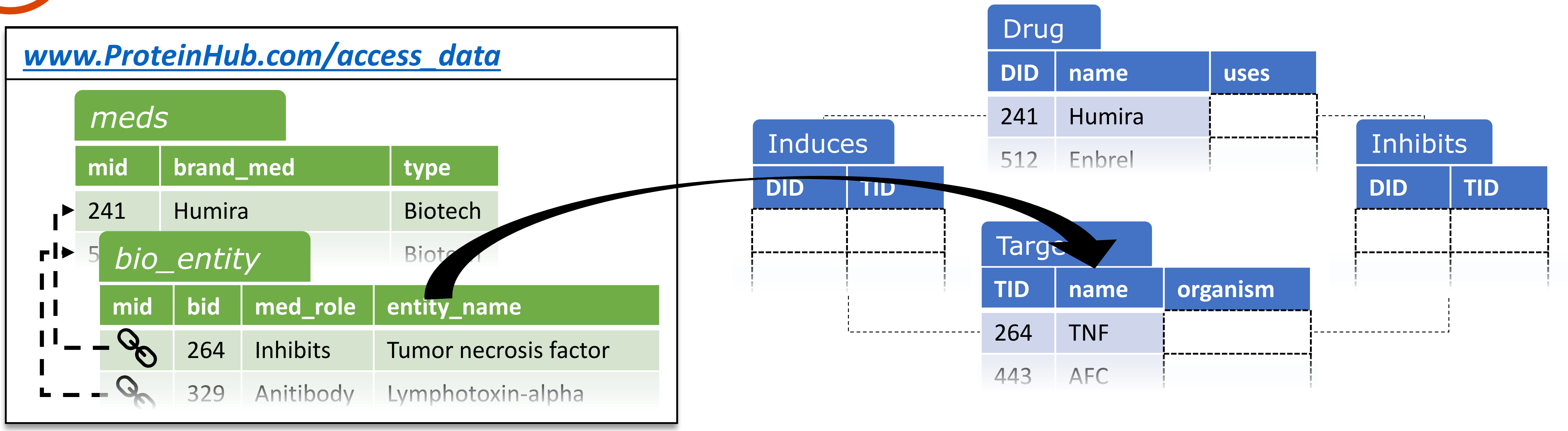
Map Target Information



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).
```

Add Target Information



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).
```

Finally, Connect Drugs and Targets

www.ProteinHub.com/access_data

meds

mid	brand_med	type
241	Humira	Biotech
5		Biotech

bio_entity

mid	bid	med_role	entity_name
	264	Inhibits	Tumor necrosis factor
	329	Anitibody	Lymphotoxin-alpha

Drug

DID	name	uses
241	Humira	
512	Enbrel	

Induces

DID	TID

Inhibits

DID	TID

Target

TID	name	organism
264	TNF	
443	AFC	

Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).
```

Consider Value of *bio_entity.med_role*

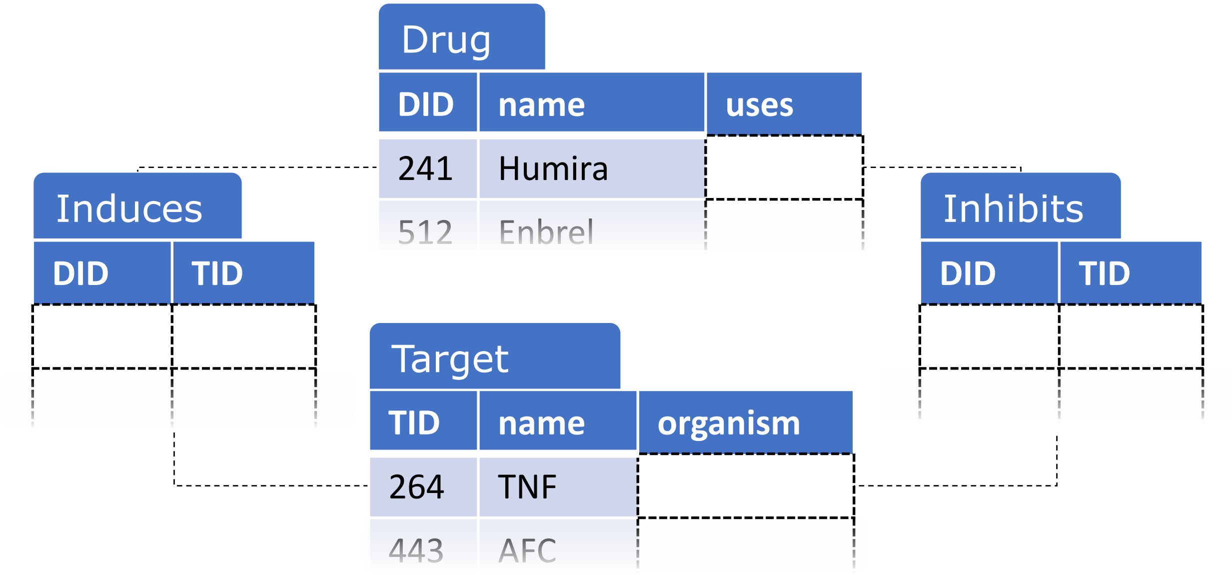
www.ProteinHub.com/access_data

meds

mid	brand_med	type
241	Humira	Biotech
5		Biotech

bio_entity

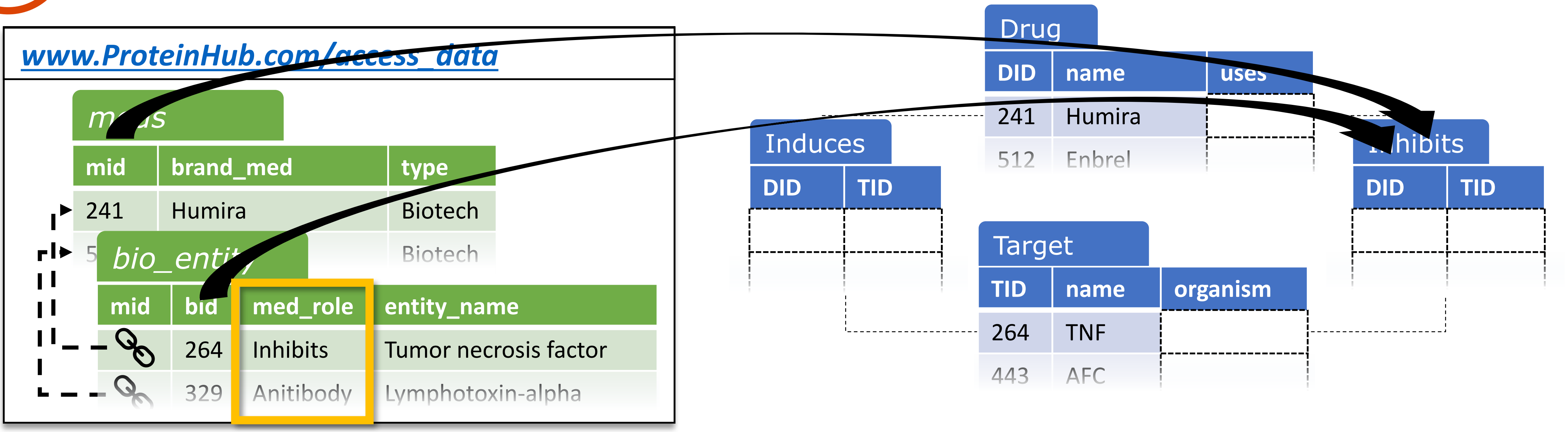
mid	bid	med_role	entity_name
	264	Inhibits	Tumor necrosis factor
	329	Anitibody	Lymphotoxin-alpha



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).
```

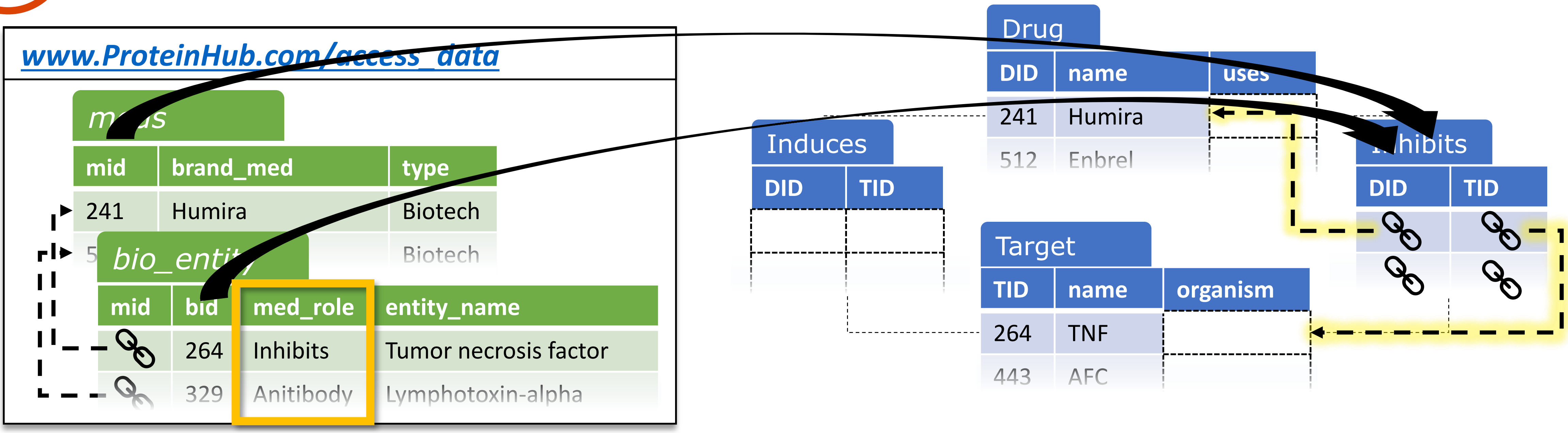
Add Drug-Inhibits-Target Information



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).
```

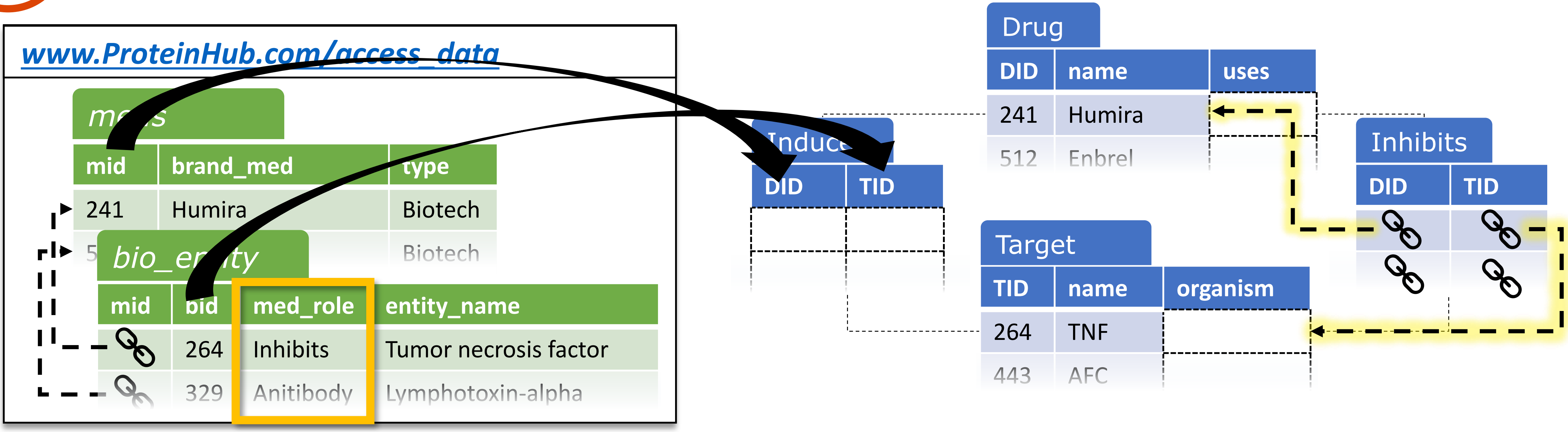
Add Drug-Inhibits-Target Information



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).

Inhibits(mid, bid) :- bio_entity(mid, bid, "Inhibits", _).
```

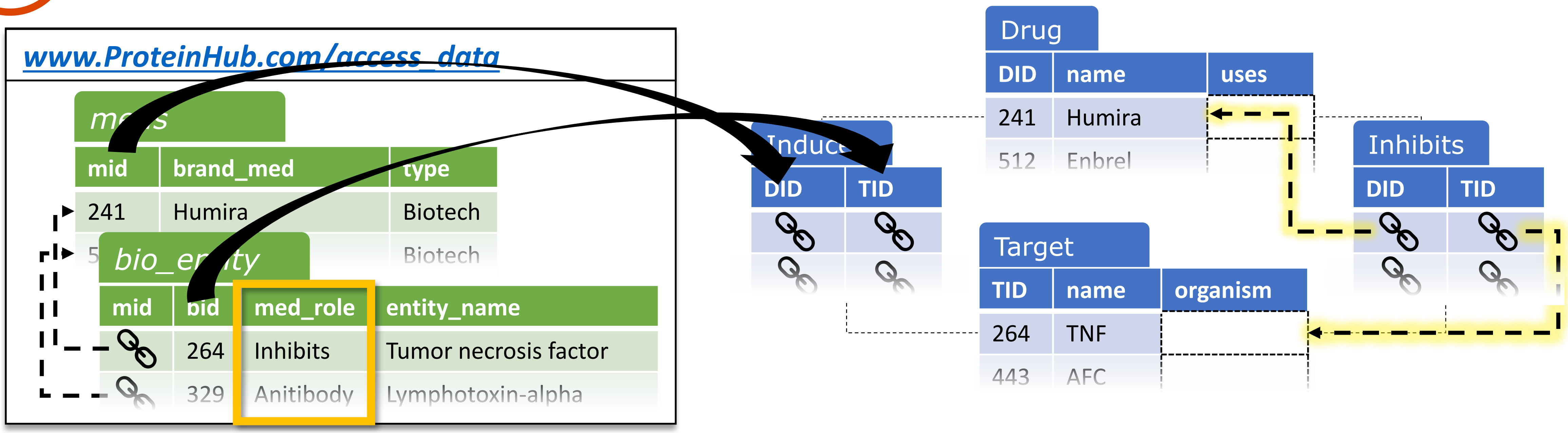
Add Drug-Induces-Target Information



Mapping:

```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).

Inhibits(mid, bid) :- bio_entity(mid, bid, “Inhibits”, _).
```

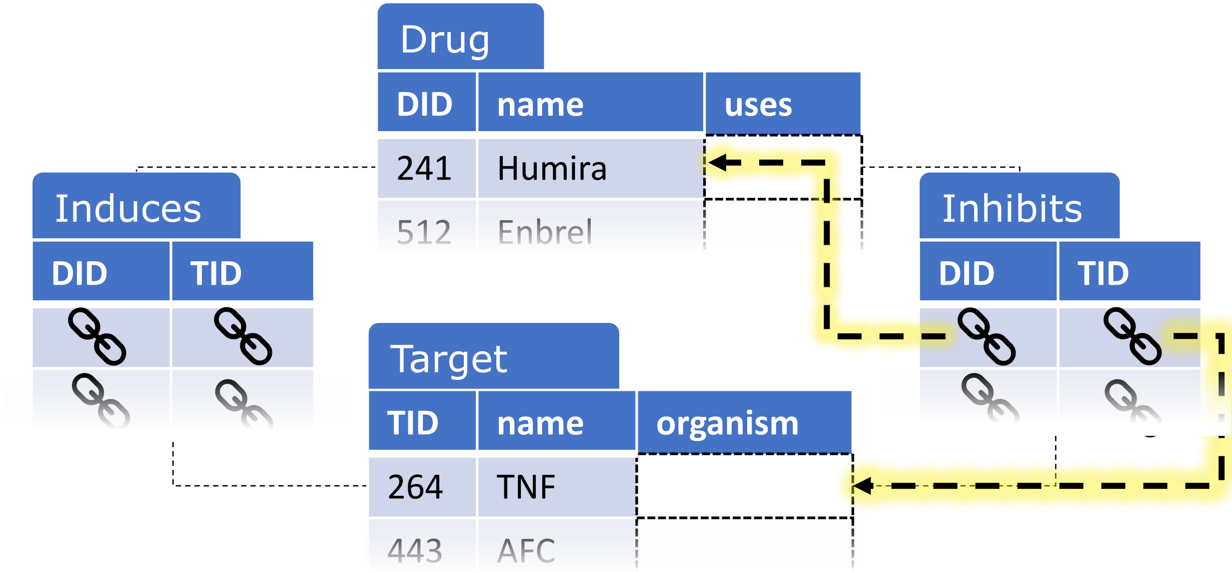
Add Drug-Induces-Target Information



Mapping:

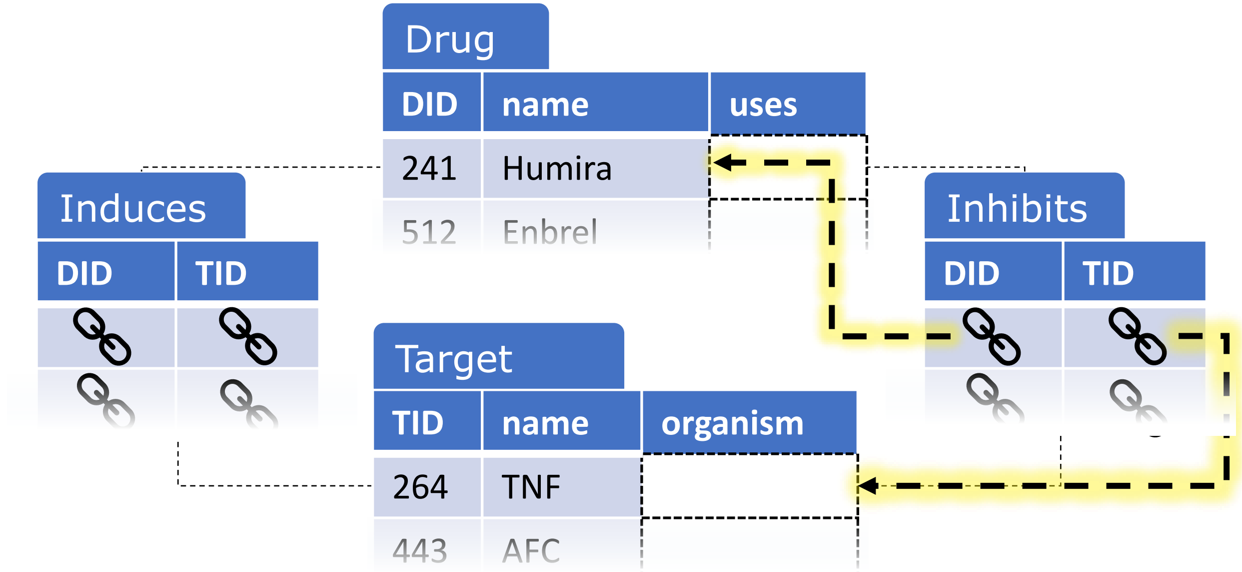
```
Drug(mid, brand_med, _) :- meds(mid, brand_med, _).

Target(bid, entity_name, _) :- bio_entity(_, bid, _, entity_name).

Inhibits(mid, bid) :- bio_entity(mid, bid, “Inhibits”, _).
Induces(mid, bid) :- bio_entity(mid, bid, “Induces”, _).
```

A (Populated) Database for Drug-Protein Interaction



~~A (Populated) Database for Drug-Protein Interaction~~

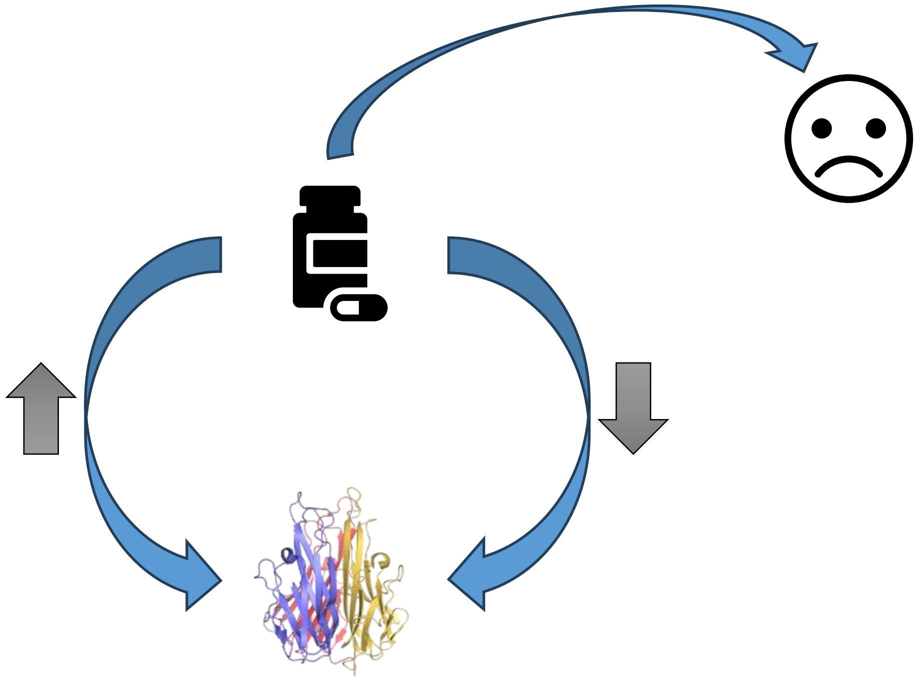
... Is not enough for drug repurposing!



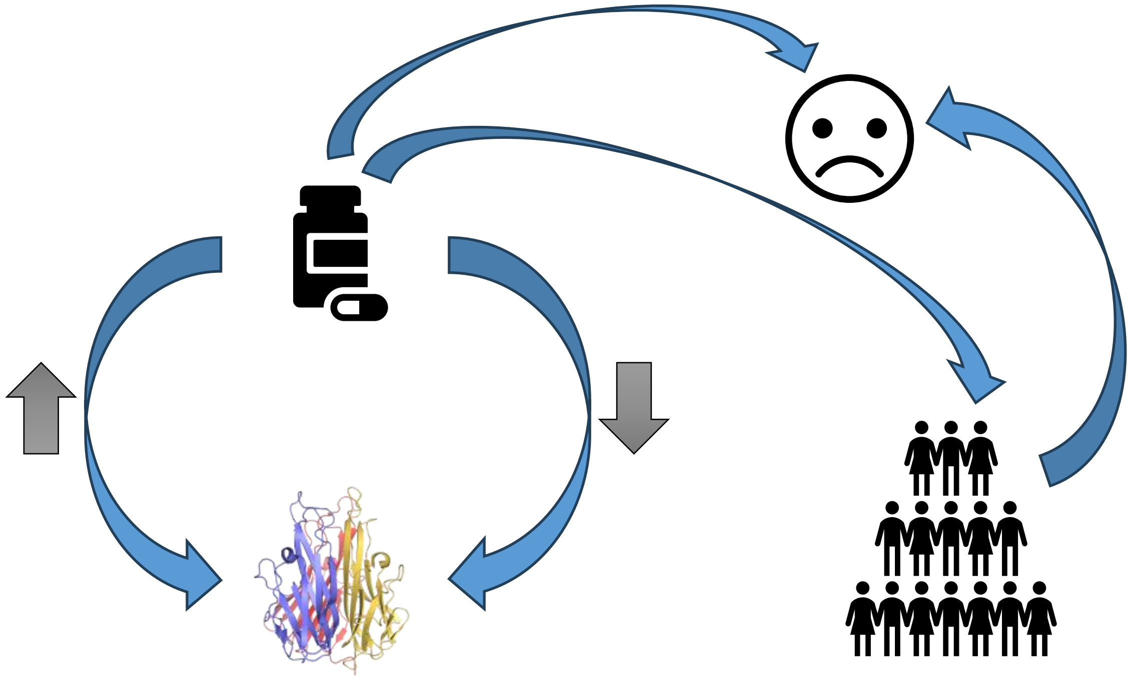
Drugs are complicated… Drug Repurposing is complicated…

Need to know more

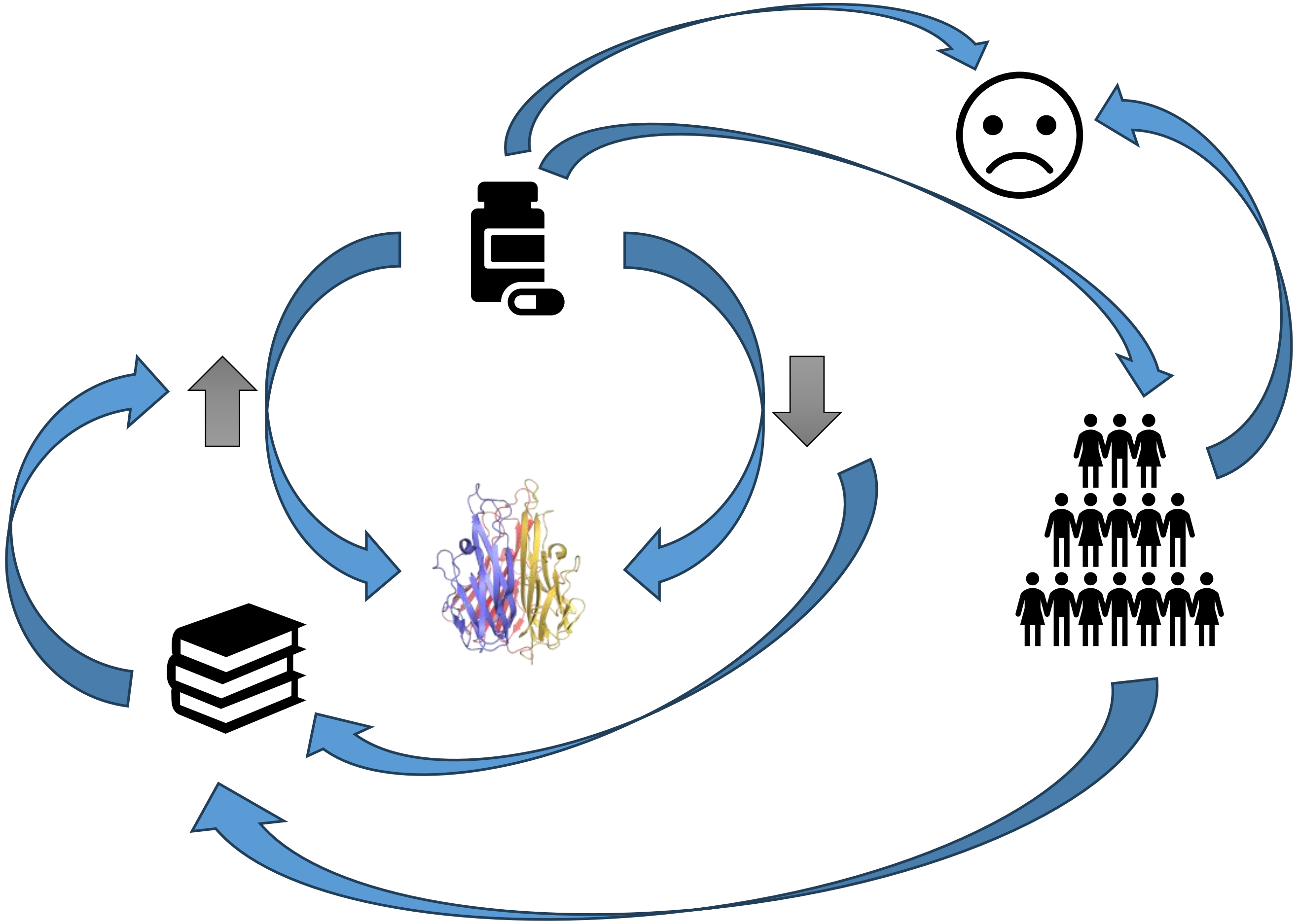
Like Reported Adverse Effects of Drugs



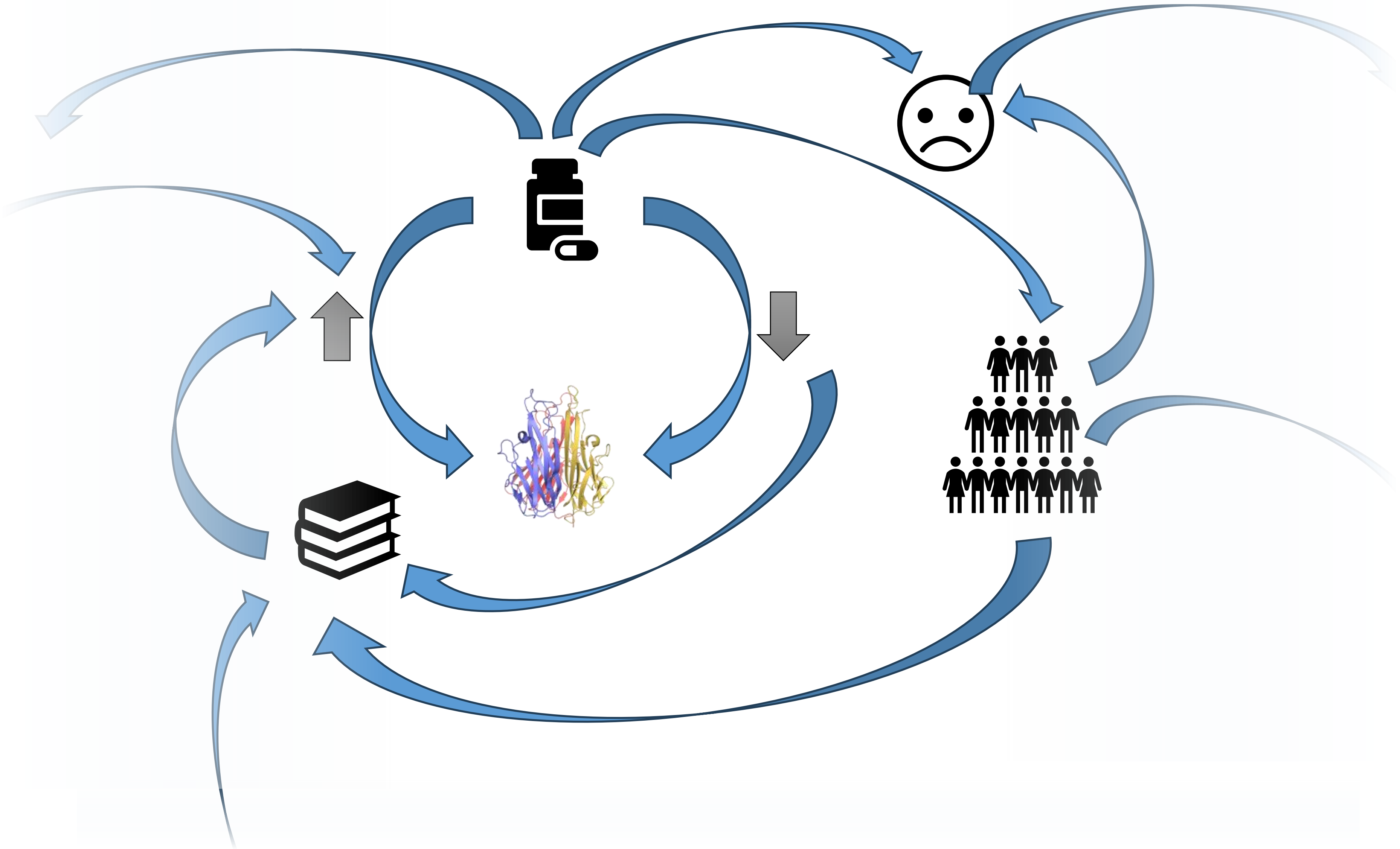
And the Newest Clinical Trial Data



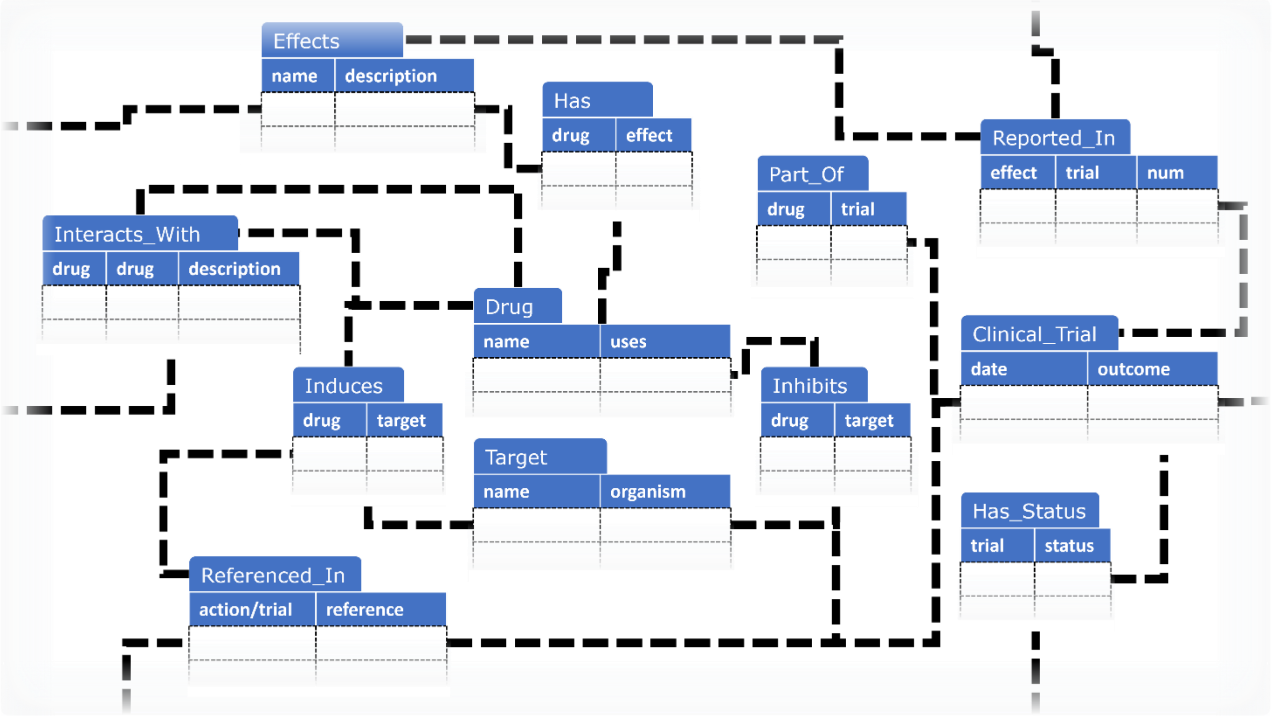
And The Research Behind all these Facts



Keep Going and Eventually, We Have ...

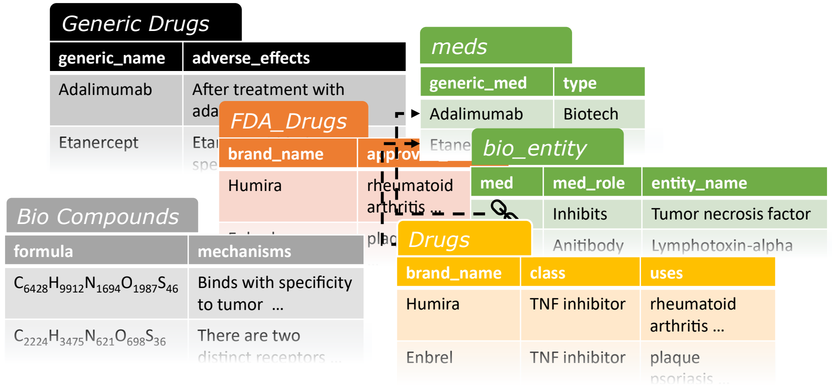


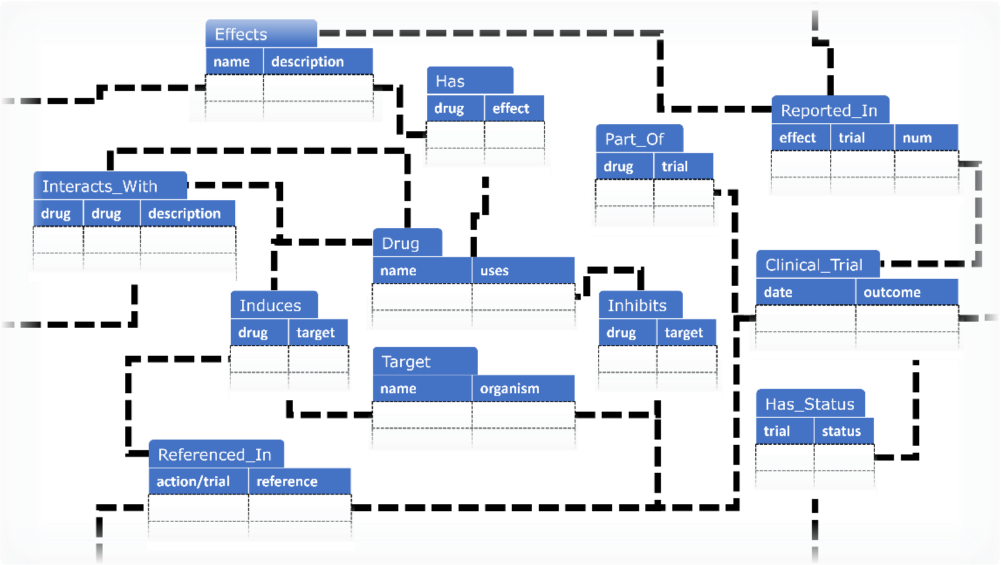
A Database for **Drug Repositioning**



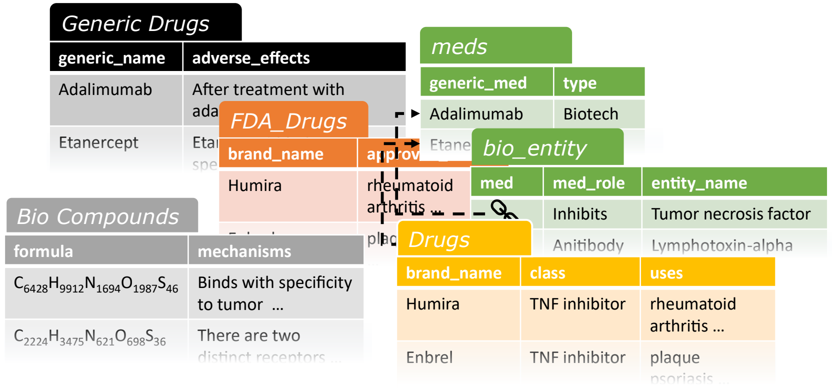
Populate THIS **database** with information

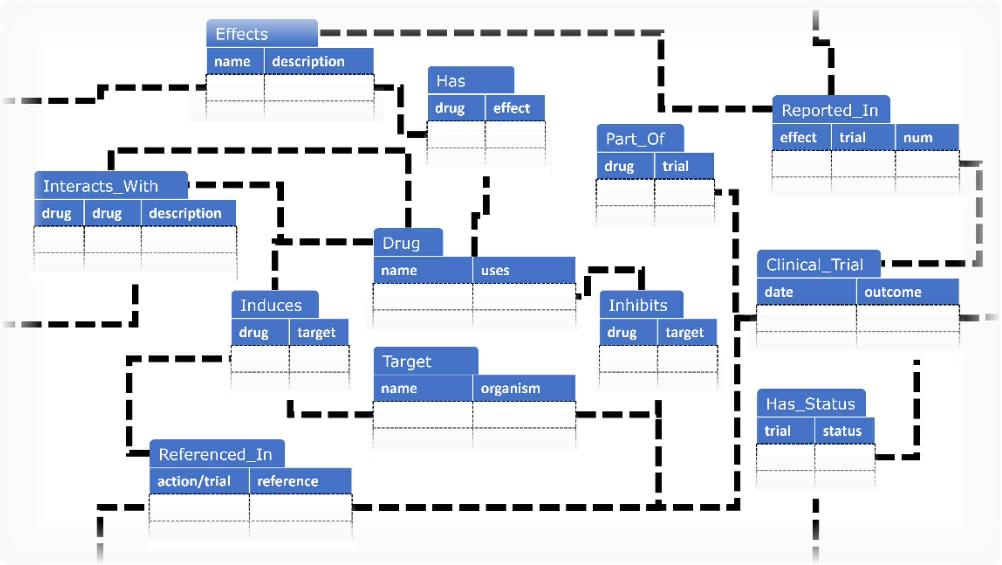
More Difficult: Requires Many More Sources…



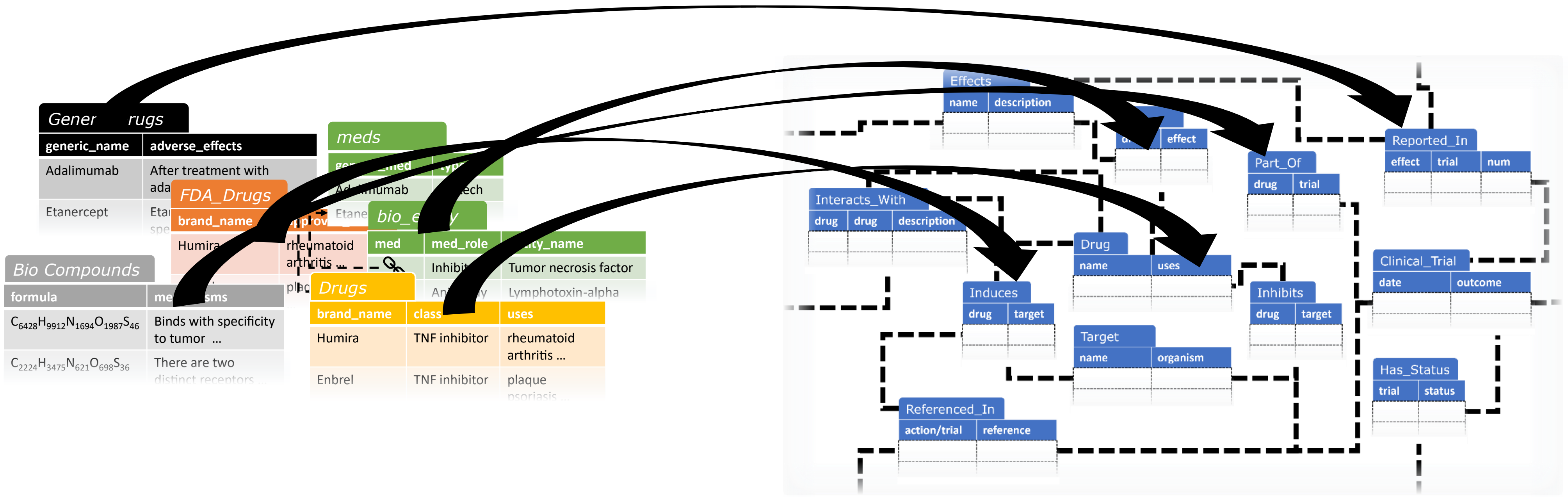


…and Many More Mappings

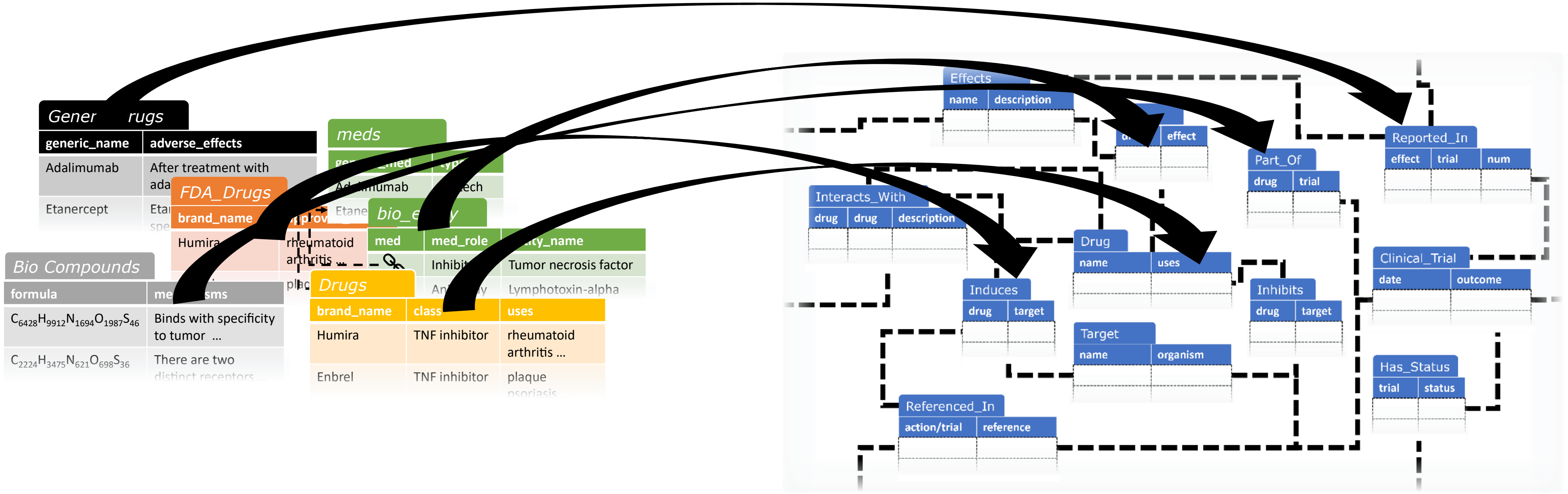




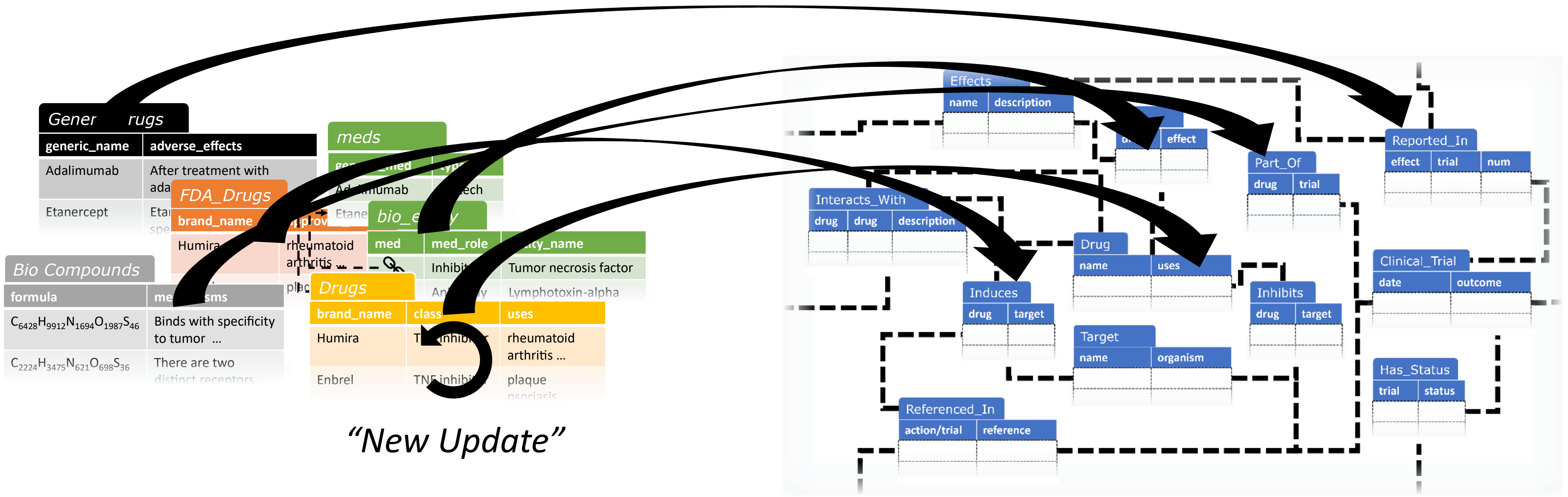
We Write the Mappings (Time-Consuming)



"Are we Finally Done?"

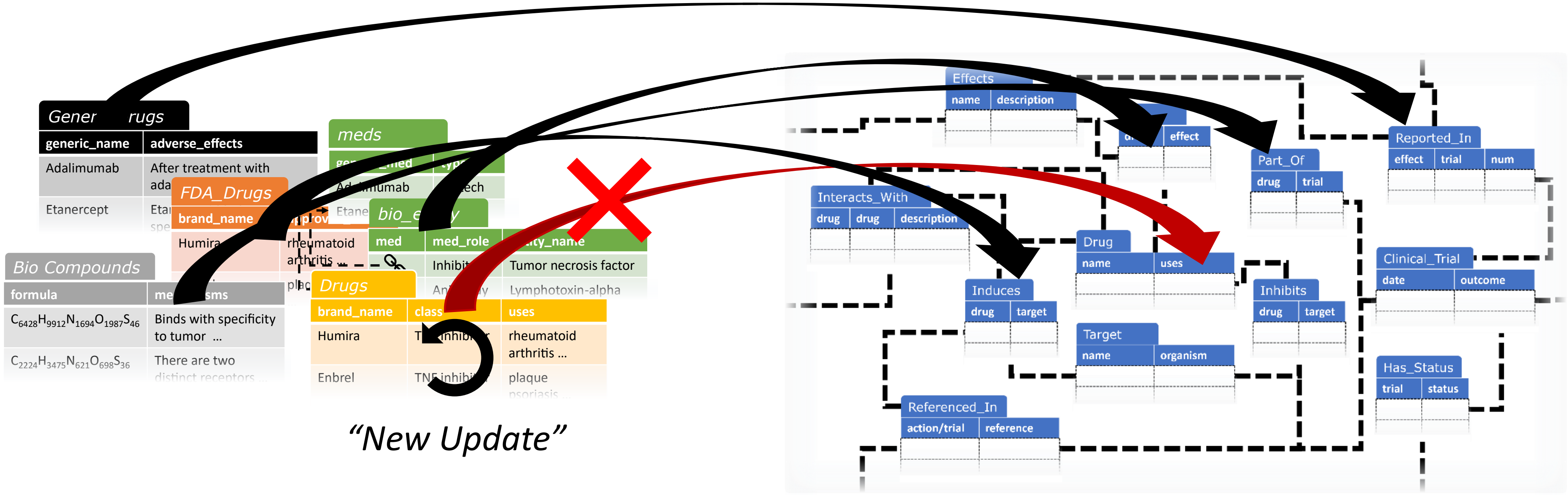


"Are we Finally Done?" No! Schema Evolution



Sources change over time

"Are we Finally Done?" No! Schema Evolution

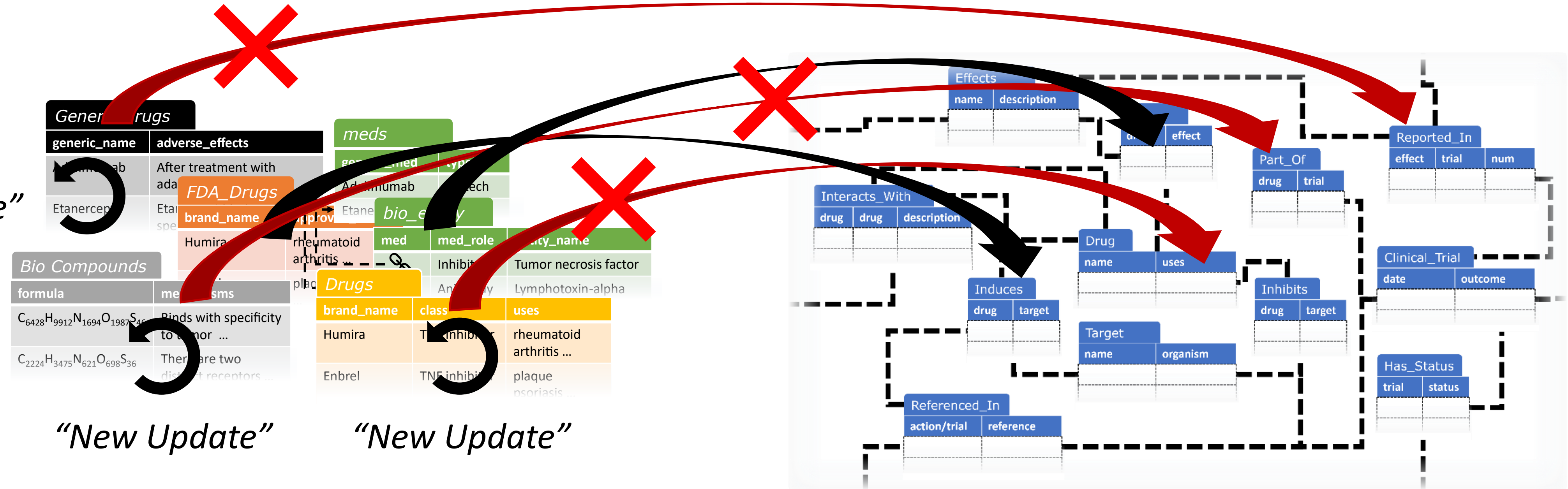


"New Update"

Sources change over time

- Must repair mapping

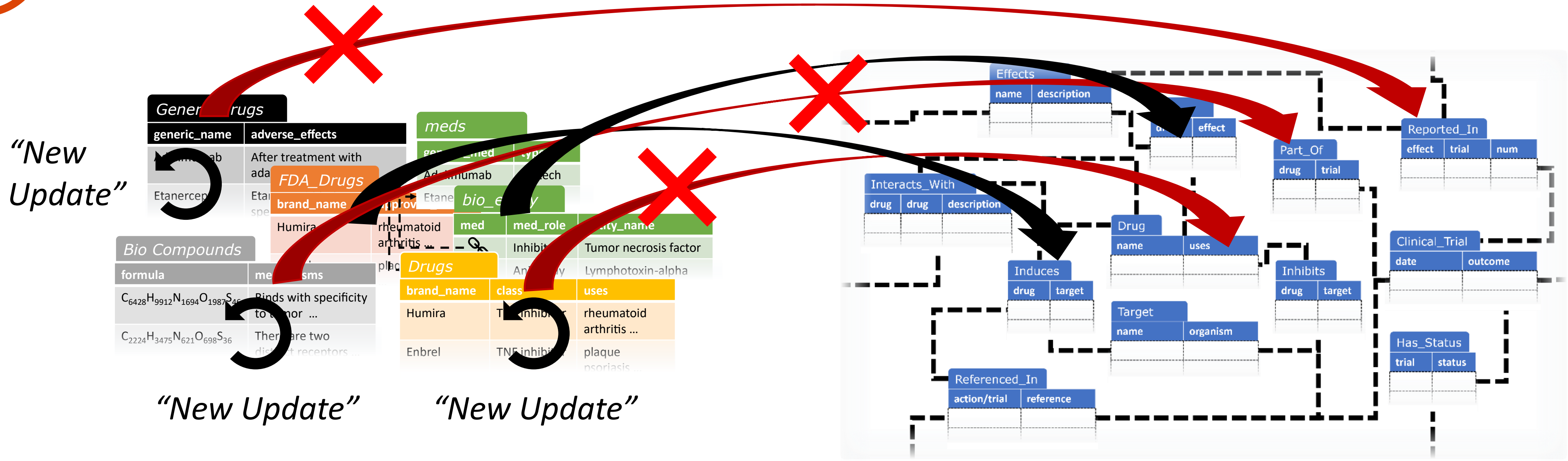
“Are we Finally Done?” No! Schema Evolution



Sources change over time

- Must repair mapping
- More sources = more repairs

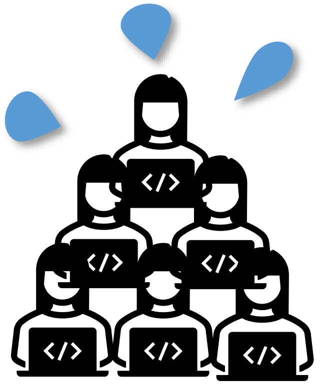
Writing + Maintenance = Effort + Delays!



We have first-hand experience…

Real Story: NIH Translator Consortium

- **Far-reaching:** ~30 teams each managing own domain-specific data integration project (database)
- **Our first-hand experience**: we've worked on one of these projects: drug repurposing for rare diseases
  - Uses ~73 sources
    - Need to integrate more, but hard to keep up with current sources!



Programmers

High maintenance cost:

Full consortium = **US$13.5 million per year!***

Time-consuming:

Long-running: Ongoing project (10+ years and going)

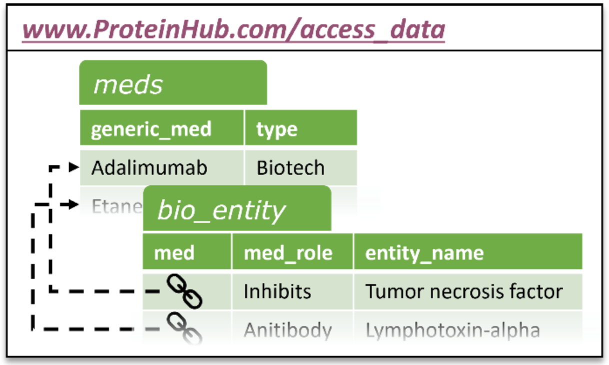
Not scalable! … Now more than ever …

Reduce Effort!
Build Mappings Faster!

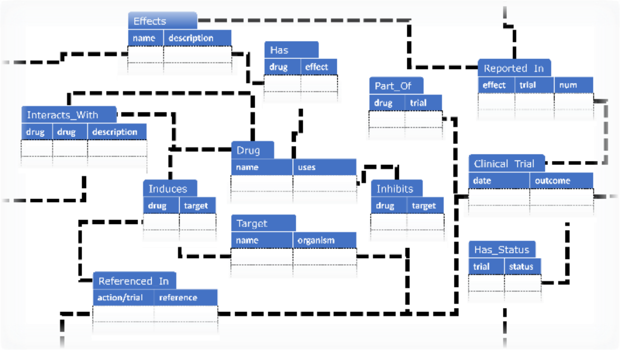
*https://ncats.nih.gov/research/research-activities/translator/about

Idea: Given a Source and Our Database…

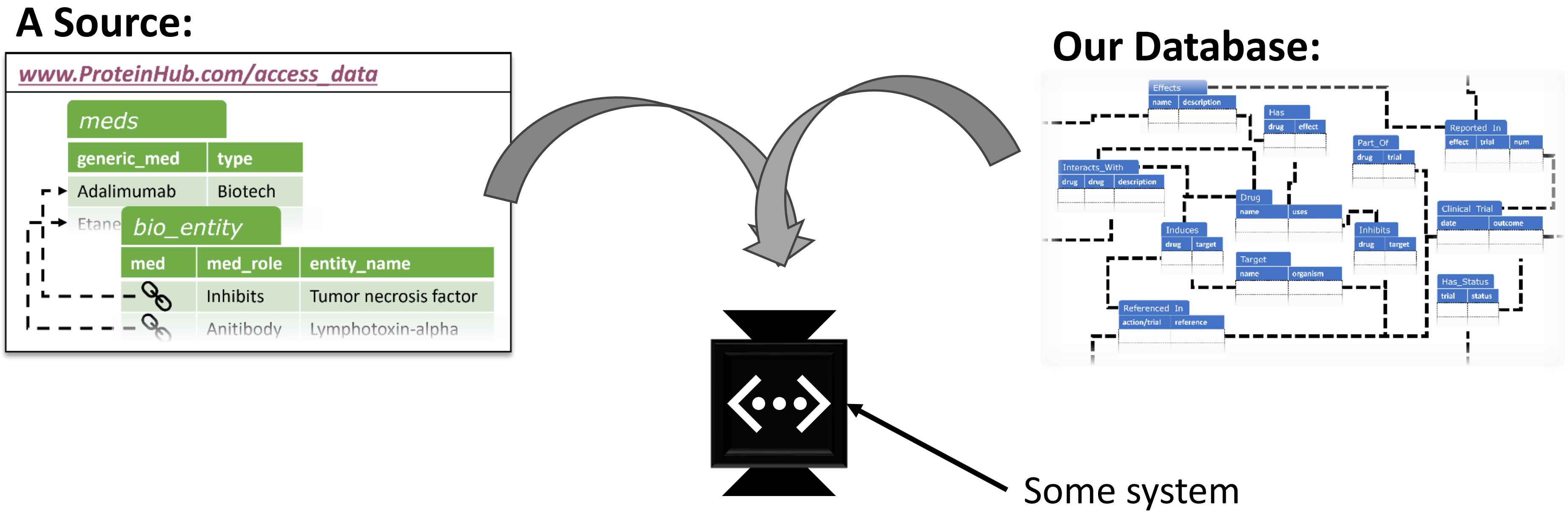
A Source:



Our Database:

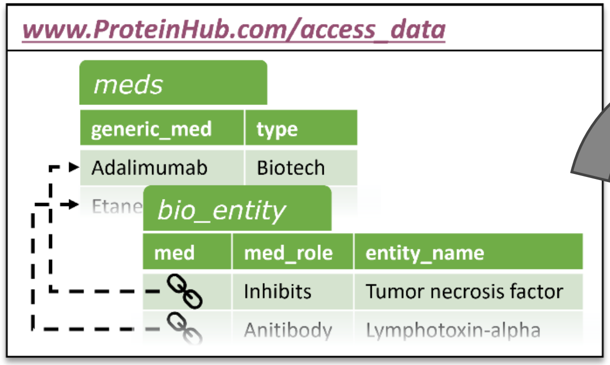


Build a System that Takes Both…

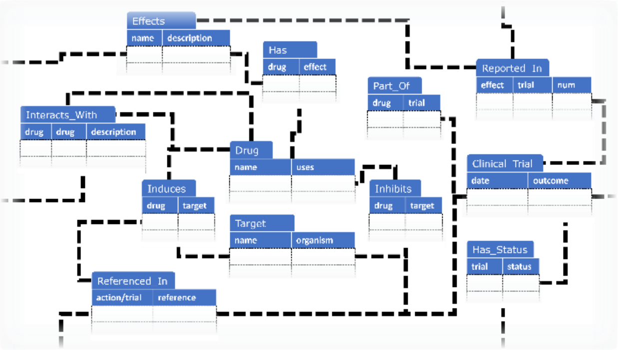


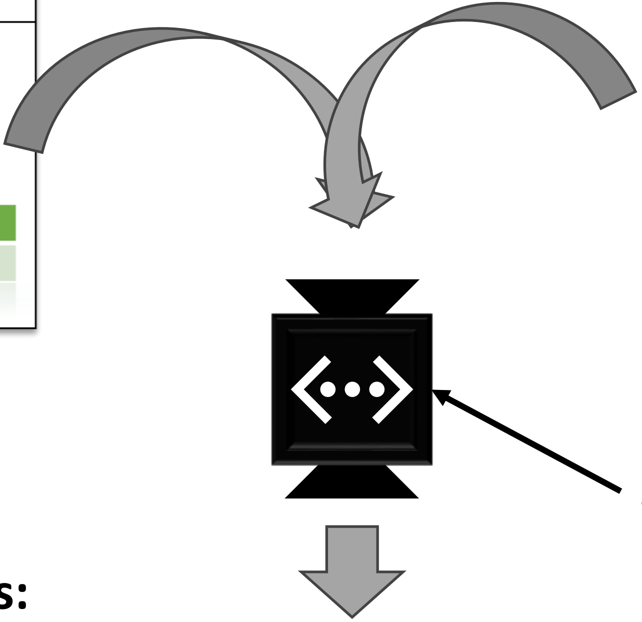
…and Produces Most Promising Mappings…

A Source:



Our Database:





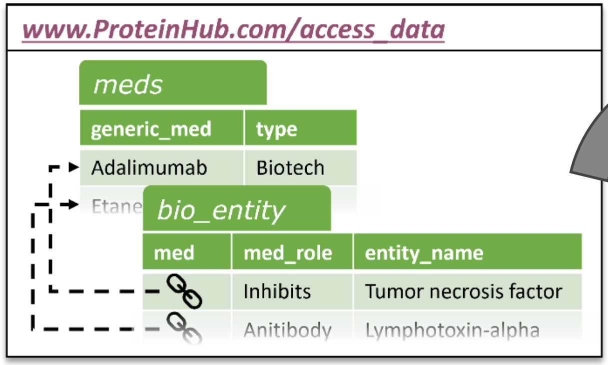
Some system

Promising Mappings:

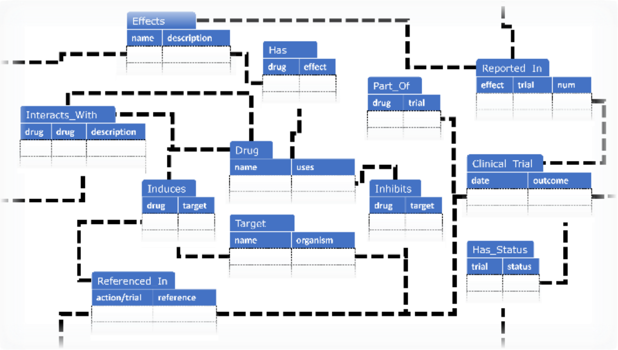
```
Drug(did, generic_med, _) :- meds(did, generic_med, _).
Target(bid, entity_name) :- bio_entity(bid, _, entity_name, _).
Inhibits(did, bid) :- bio_entity(bid, did, _, “Inhibits”).

Target(did, entity_name) :- meds(did, _, generic_med, _).
                          …
```

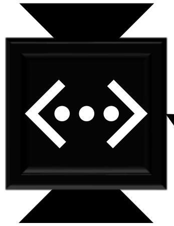
…Which Someone Can Verify and Use

A Source:



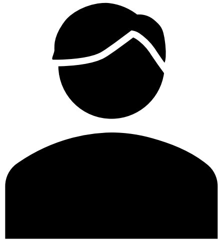
Our Database:





Some system

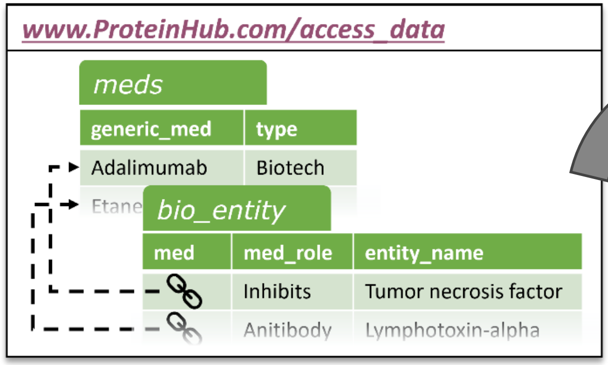
Promising Mappings:



```
Drug(did, generic_med, _) :- meds(did, generic_med, _).
Target(bid, entity_name) :- bio_entity(bid, _, entity_name, _).
Inhibits(did, bid) :- bio_entity(bid, did, _, “Inhibits”).

Target(did, entity_name) :- meds(did, _, generic_med, _).
…
```

How Can we Build this System?

A Source:



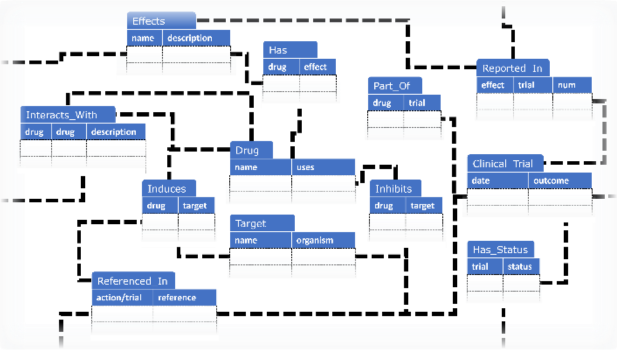
meds

generic_med	type
Adalimumab	Biotech
Etane	

bio_entity

med	med_role	entity_name
	Inhibits	Tumor necrosis factor
	Anitibody	Lymphotoxin-alpha

Our Database:



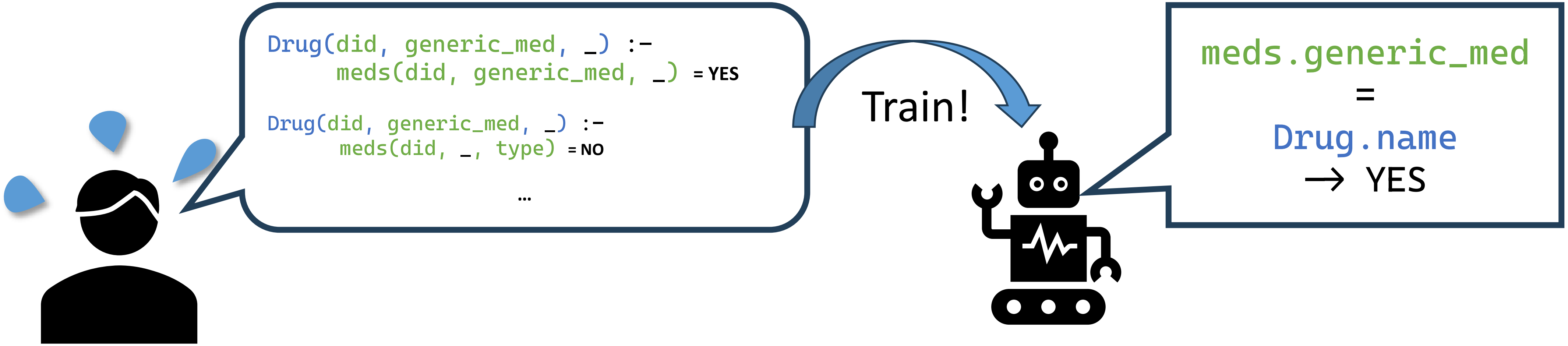
?

Promising Mappings:

```
Drug(did, generic_med, _) :- meds(did, generic_med, _).
Target(bid, entity_name) :- bio_entity(bid, _, entity_name, _).
Inhibits(did, bid) :- bio_entity(bid, did, _, "Inhibits").

Target(did, entity_name) :- meds(did, _, generic_med, _).
...
```

Supervised Learning

1. Label training data
2. Feed to a model
3. Generate mappings



Labeling data takes a lot of time and manual effort…

…which needs to be repeated as sources evolve

Opportunity:
LLMs for Schema Mapping

<u>Some examples:</u>
Zhang et al. "SMAT: An attention-based deep learning solution to the automation of schema matching." ADBIS. (2021)
Mudgal et al. "Deep learning for entity matching: A design space exploration." SIGMOD. (2018).

Current State: Using LLMs Column Alignment

Input:

Drug

name	uses
Humira	Rheumatoid ...

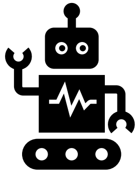
bio_entity

med	med_role	entity_name
	Inhibits	Tumor necrosis factor
	Anitibody	Lymphotoxin-alpha

Including supporting info…

- Column/Table descriptions
- Schematic (types, etc,.)
- Sample data values …

Prompt



Response (column pairs):

```
(meds.generic_med, Drug.name)
…
```

Data from __ can be mapped to __

Some Examples:

Huang et al. "Transform Table to Database Using Large Language Models." TaDa @ VLDB. (2024)
Sheetrit et al. "ReMatch: Retrieval Enhanced Schema Matching with LLMs." arXiv (2024)

Goal: Maximize Response Quality w/o Training

Input:

Drug

name	uses
Humira	Rheumatoid …

bio_entity

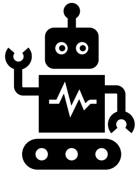
med	med_role	entity_name
	Inhibits	Tumor necrosis factor
	Anitibody	Lymphotoxin-alpha

Including supporting info…

- Column/Table descriptions
- Schematic (types, etc,.)
- Sample data values …

Prompt

Response (column pairs):



```
(meds.generic_med, Drug.name)
…
```

LLMs are **sensitive to task phrasing**!
… mitigate this sensitivity.

Research suggests* that effective techniques for…

- sampling candidate responses, and
- combining those responses

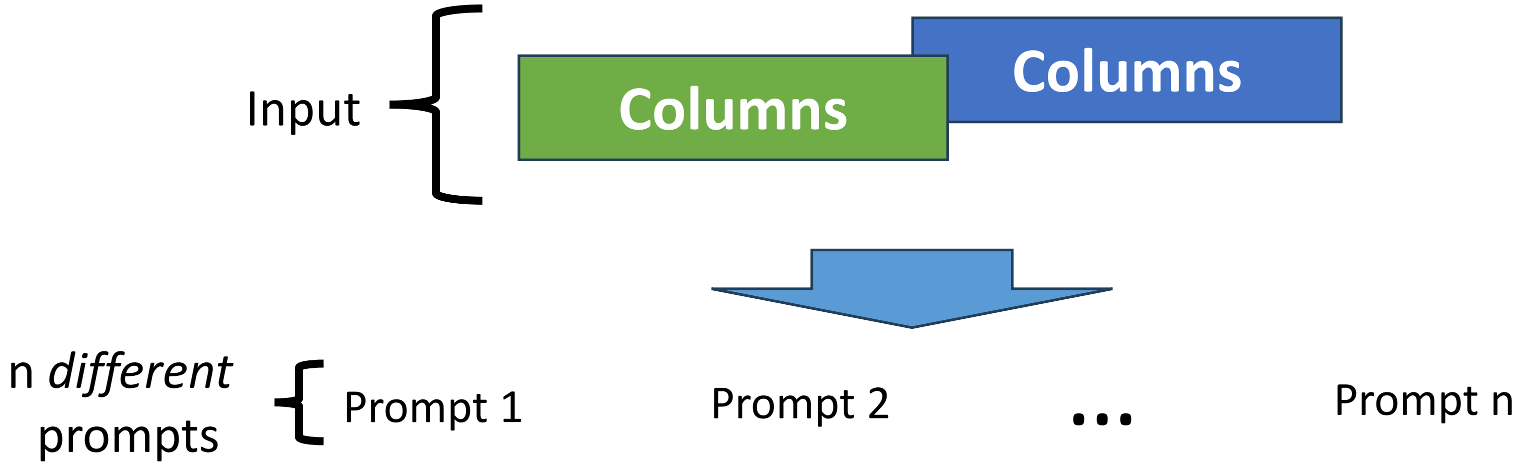
Can rival fine-tuned performance**

Us: develop sampling and combining techniques for column alignment

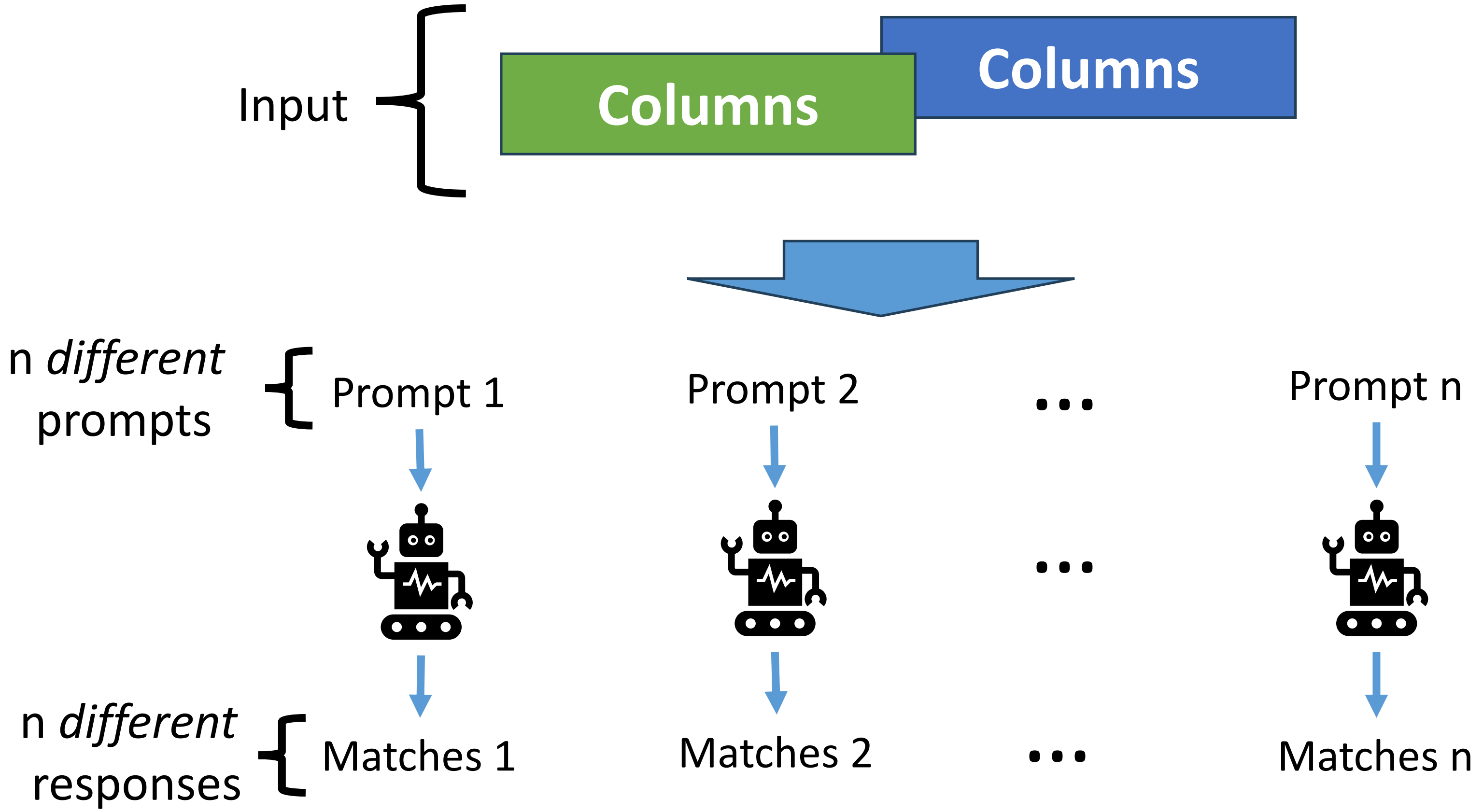
X. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models." arXiv (2023)
*** authors observe this trend over general reasoning benchmarks*

High-Level: Given a Column Alignment Task

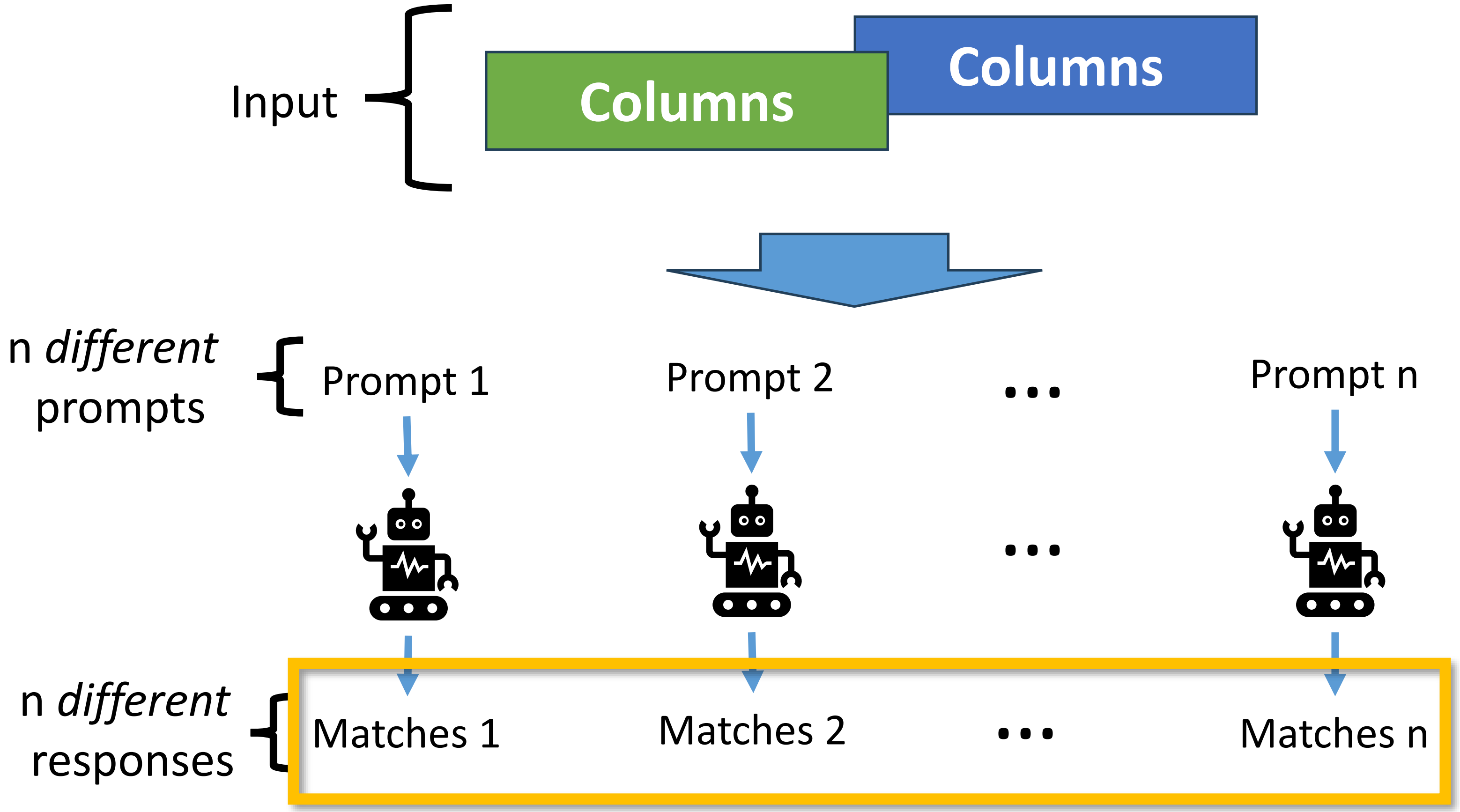
Generate n Prompts



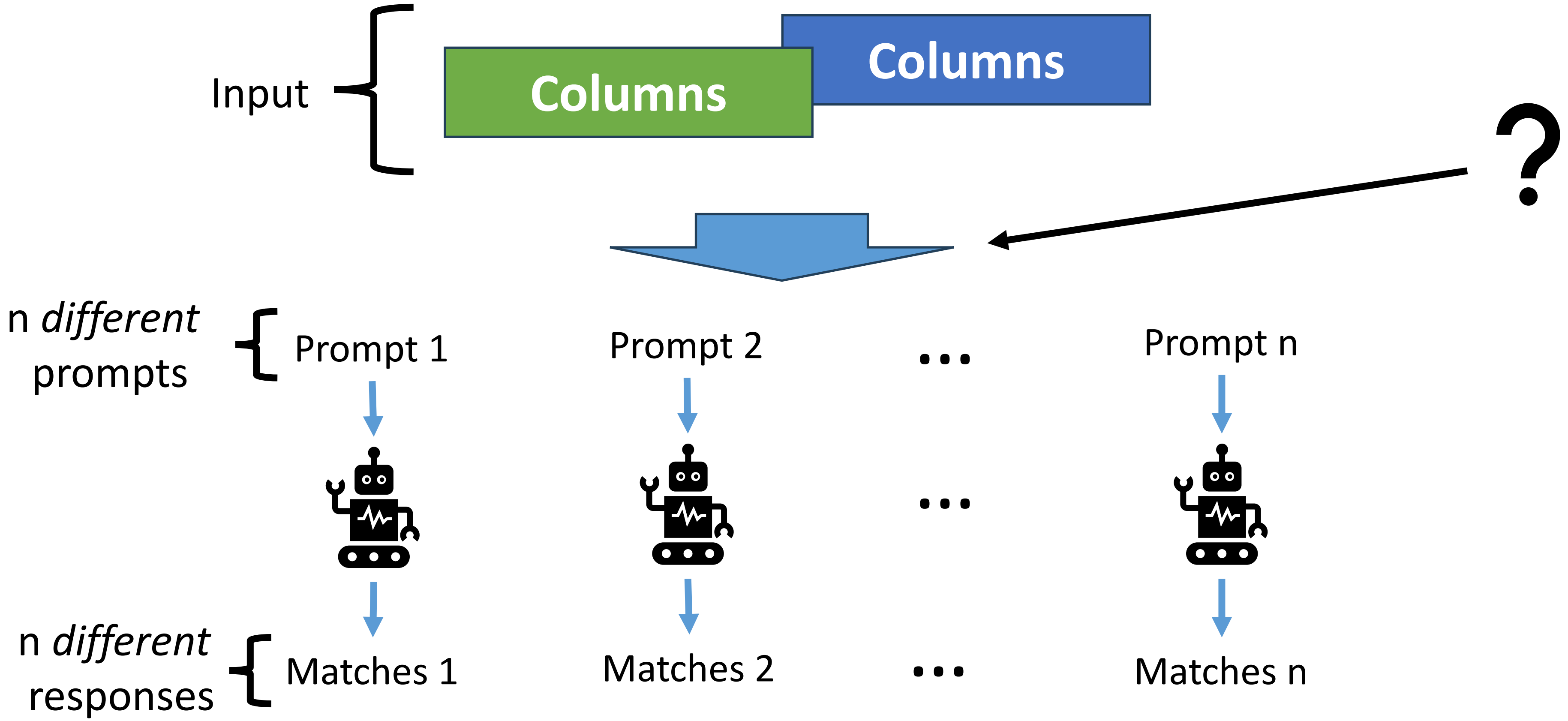
Giving n Different Responses



Derive Most-Consistent Alignment Pairs



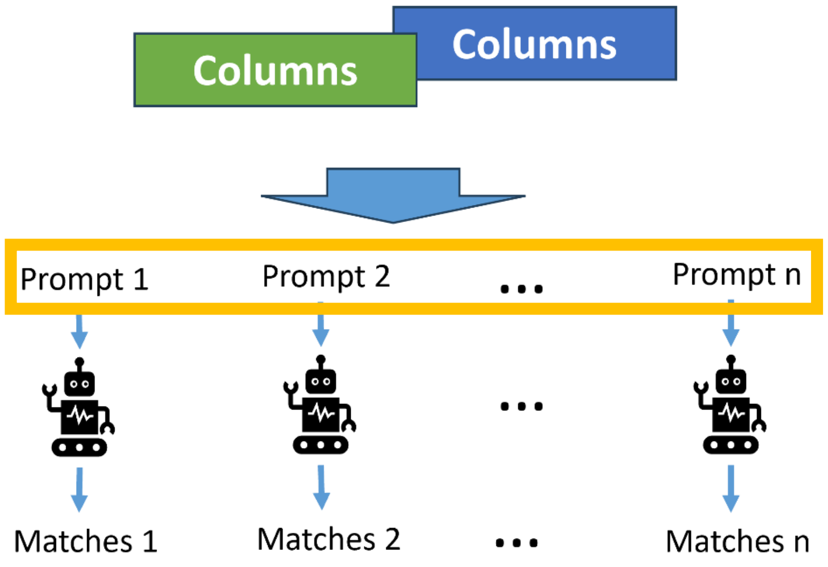
Generate Prompts to Offset Phrasing Noise



Techniques for Generating Prompt Variations

Want: all prompts reflect same task

w/ variations in phrasing



Combine

Resample data values for each column

Drug.name: Samples: **Humira, Enbrel …** …	…	Drug.name: Samples: **Vyvanse, Advil …** …

Take Advantage of Problem Symmetries:

- Randomly reorder columns

Drug.name: … … Target.name: …	…	Target.name: … … Drug.uses: …

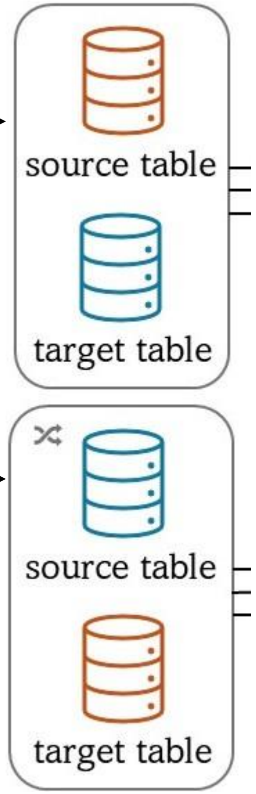
- Swap source table and our table

Source:	Ours:
meds.type: …	Drug.name: …
…	…

Source:	Ours:
Drug.name: …	meds.type: …
…	…

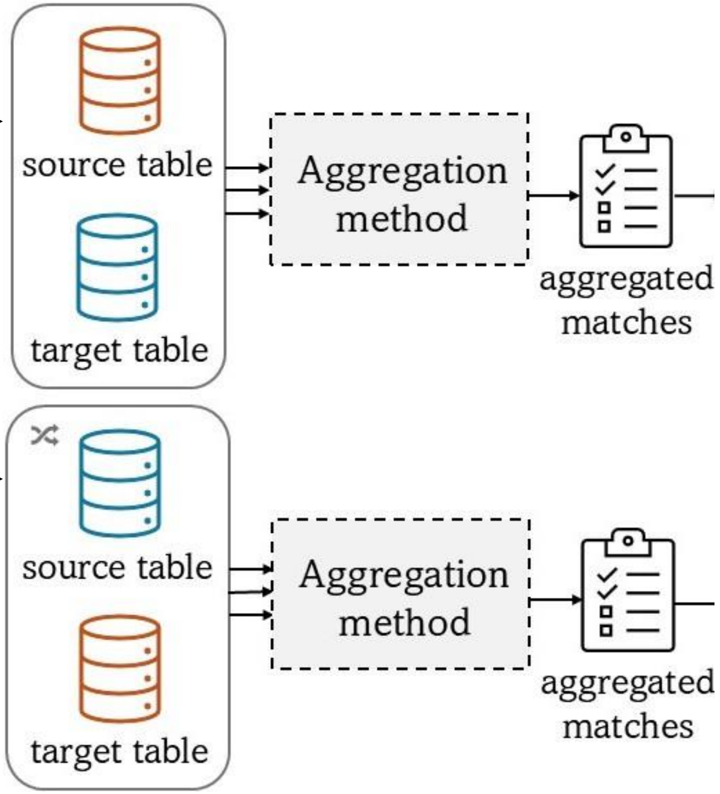
Response Combination (Bidirectional Matching)

1. Prompt 6 times
   - 3x Unswapped
   - 3x Swapped



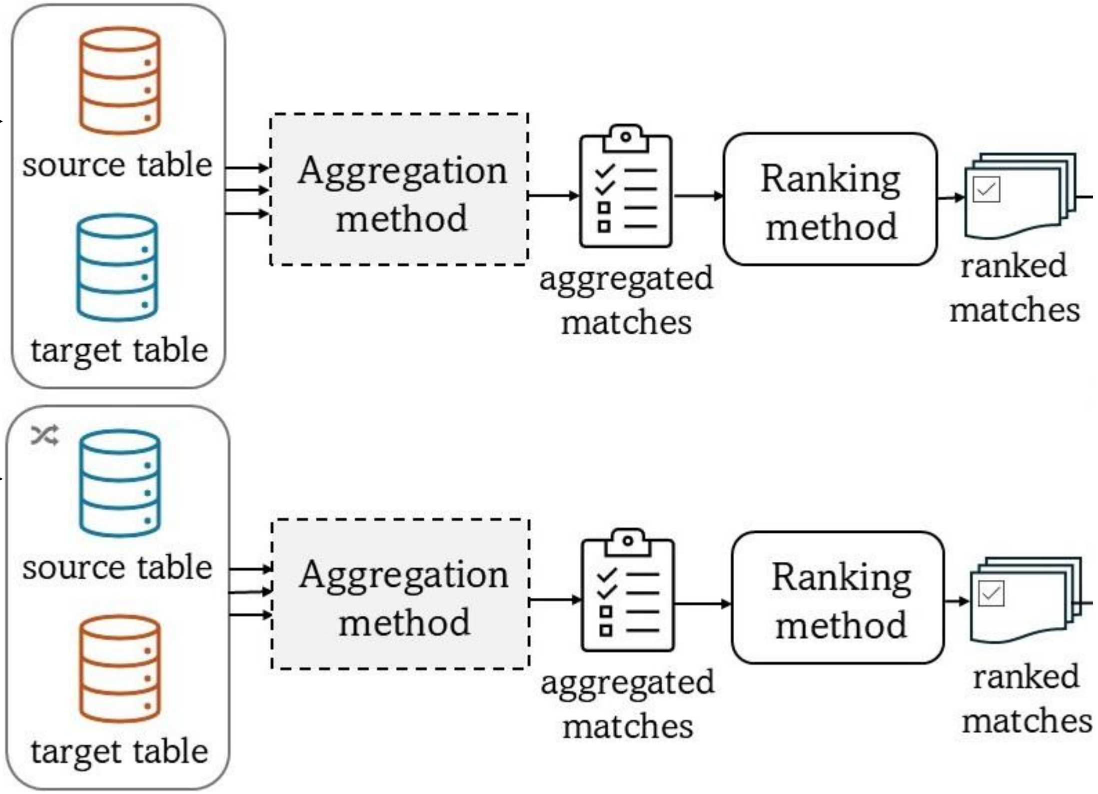
Response Combination (Bidirectional Matching)

1. Prompt 6 times
   - 3x Unswapped
   - 3x Swapped
2. Aggregate
   - Use <u>majority vote</u> over alignment pairs



Response Combination (Bidirectional Matching)

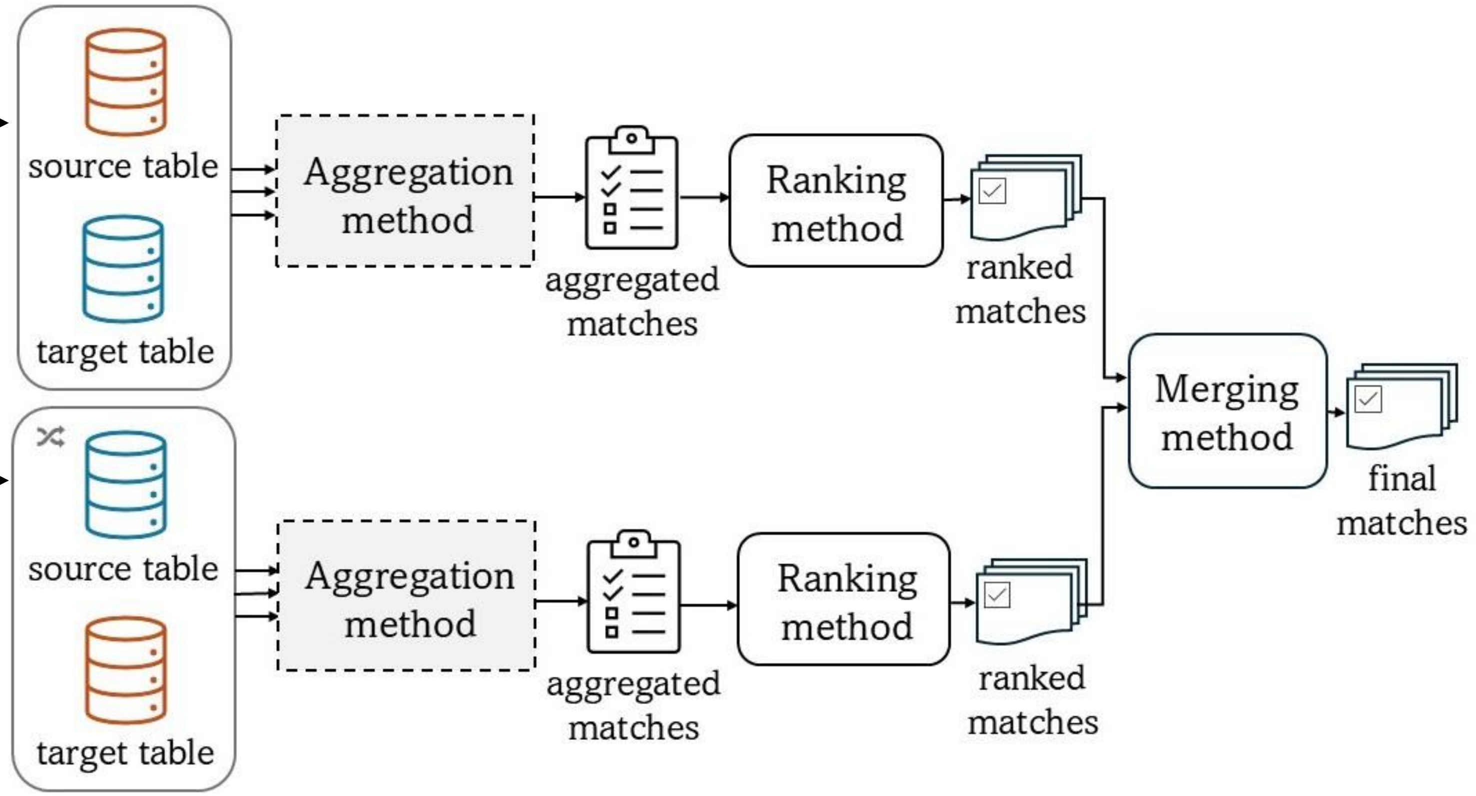
1. Prompt 6 times
   - 3x Unswapped
   - 3x Swapped
2. Aggregate
   - Use majority vote over alignment pairs
3. Rank Aggregated Pairs
   - Score pairs using probability from LLM (logits)



Response Combination (Bidirectional Matching)

1. Prompt 6 times
   - 3x Unswapped
   - 3x Swapped
2. Aggregate
   - Use <u>majority vote</u> over alignment pairs
3. Rank Aggregated Pairs
   - <u>Score pairs</u> using probability from LLM (logits)
4. Merge Ranked Lists
   - <u>Average</u> **OR** <u>Multiply</u> scores

     OR
   - Find <u>Stable Matching</u>
     - See paper for more details



Preliminary Experiments

Dataset: MIMIC and Synthea (clinical)

Metric: Accuracy@1

- Lower in rank = User less likely to see

LLM: we use Llama-3.1 70B Parameter (quantized INT4) *[open-source]*

Competitive with Methods that Use GPT-4

Dataset: MIMIC and Synthea (clinical)

Metric: Accuracy@1

- Lower in rank = User less likely to see

LLM: we use Llama-3.1 70B Parameter (quantized INT4) *[open-source]*

Dataset	Method	Accuracy@1
MIMIC	MatchMaker *	62.20 ± 2.40
	Bidirectional (Stable Matching)	0.78 ± 0.00
	Bidirectional (Average)	0.49 ± 0.01
	Bidirectional (Multiply)	0.77 ± 0.01
Synthea	MatchMaker *	70.20 ± 1.70
	Bidirectional (Stable Matching)	0.69 ± 0.01
	Bidirectional (Average)	0.64 ± 0.01
	Bidirectional (Multiply)	0.70 ± 0.01

Significantly better

Not significantly worse

**As reported in,*
Seedat and Schaar. Matchmaker: Self-Improving Compositional LLM Programs for Table Schema Matching. TRL @ NeurIPS. (2024)

Competitive with Methods that Use GPT-4

Dataset: MIMIC and Synthea (clinical)

Metric: Accuracy@1

- Lower in rank = User less likely to see

LLM: we use Llama-3.1 70B Parameter (quantized INT4) *[open-source]*

Dataset	Method	Accuracy@1
MIMIC	MatchMaker *	62.20 ± 2.40
	Bidirectional (Stable Matching)	0.78 ± 0.00
	Bidirectional (Average)	0.49 ± 0.01
	Bidirectional (Multiply)	0.77 ± 0.01
Synthea	MatchMaker *	70.20 ± 1.70
	Bidirectional (Stable Matching)	0.69 ± 0.01
	Bidirectional (Average)	[illegible]
	Bidirectional (Multiply)	[illegible]

Significantly better

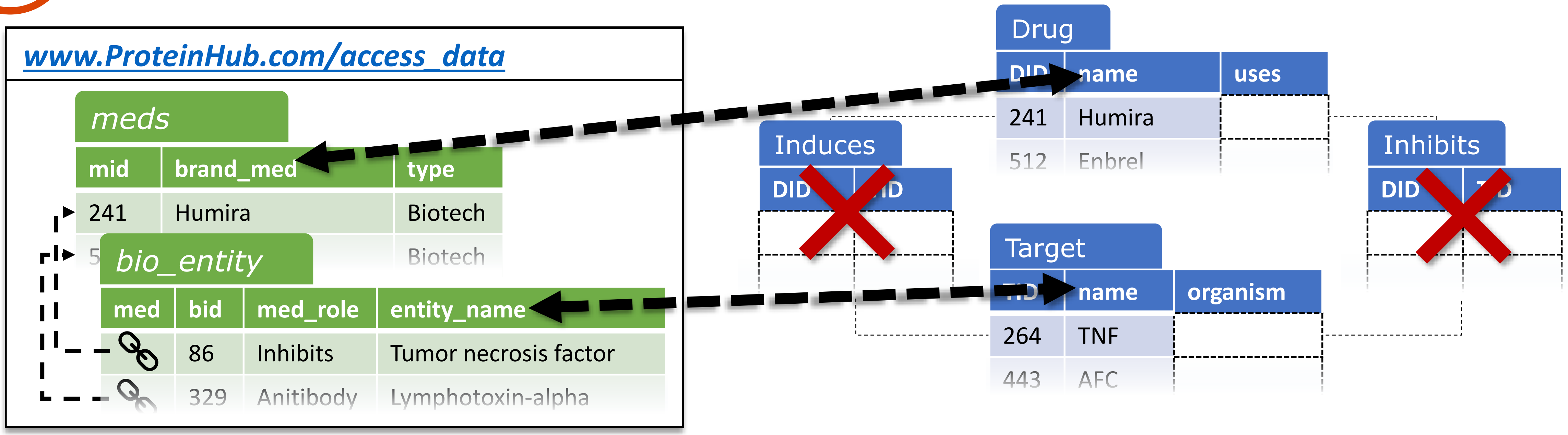
Not significantly worse

Great, but column alignments have limited usefulness

**As reported in,*

Seedat and Schaar. Matchmaker: Self-Improving Compositional LLM Programs for Table Schema Matching. TRL @ NeurIPS. (2024)

Column Alignments = Too Simple



Can tell us…

- "Move data from this column to that one…"

Cannot tell us…

- Which **Drugs** induce (inhibit) which **Targets**

Not suitable for many common mapping scenarios

How do we extend these techniques to more expressive mappings?

Moving Beyond Column Alignments (Complex!)

*See paper for more detailed discussion

Set of column pairs → **Set of multi-query programs**

How to Sample & Combine Responses?
- Swapping schemas = drastically change output
- Not clear how to combine outputs

How to Divide & Conquer? Give LLM…
- too many relations = poor performance
- too few relations = incorrect mapping

Future Work

What Output Language?
- LLMs can generate SQL query given *question* and *schema* **[Text-to-SQL]**
- What about **Schema Mapping**?
    - Multiple queries; rigid requirements on output structure

Preliminary Results

Experiment: Effectiveness

Dataset: Amalgam (bibliography):

- 8 independent mappings programs (prompt for each, individually)

Metric: Table-Overlap (Avg. 20 runs)

- Average of metrics over **gold** vs. **predicted** table rows

(a) Metrics		
Prec.	**Rec.**	**F1**
0.56 ± 0.03	0.85 ± 0.03	0.66 ± 0.03

*See paper for more experiments and results

Moves too much data

SQL seems OK.
Focus on techniques for improving output.

Thank you!

Please share your questions

Portland State University

Oregon State University

Shortcomings: Existing Approaches

Provide supplemental information

- Group columns into semantic categories prior to matching
- Identify helpful knowledge sources, build locally or connect to API for querying

Still requires (potentially significant) human effort

Train over Synthetic Data

- LLM generates training data (in-context learning)

LLMs are **sensitive to phrasing**, and same phrasing can still give **conflicting answers**!

Find most consistent response -> rivals fine-tuned performance

Some Examples:

Narayan et al. "Can Foundation Models Wrangle Your Data?." VLDB (2022)
Huang et al. "Transform Table to Database Using Large Language Models." TaDa @ VLDB. (2024)
Sheetrit et al. "ReMatch: Retrieval Enhanced Schema Matching with LLMs." arXiv (2024)