

# Towards **Agentic** Data Processing with DocETL

[docetl.org](https://docetl.org)

Shreya Shankar\*, Aditya G. Parameswaran, Eugene Wu  
UC Berkeley, EECS  
October 2024

EPIC  
DATA lab  
UC Berkeley

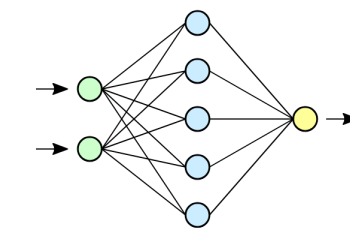
 COLUMBIA UNIVERSITY  
IN THE CITY OF NEW YORK

# Analyzing Unstructured Data is Hard

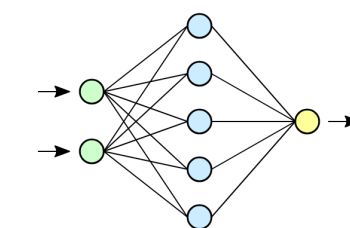
- Organizations have lots of data and *intelligent data analysis* needs
  - These require teams of human annotators 🧑
- Consider a big hospital 🏥
- Lots of data, e.g.,
  - Patient records
  - Documentation (medicines, illnesses, etc.)
- ML is promising but hard



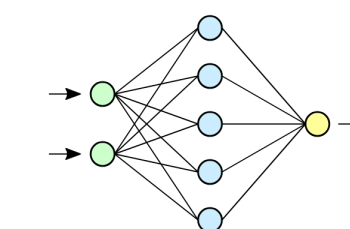
## MLOps challenges



Extract medication side effects

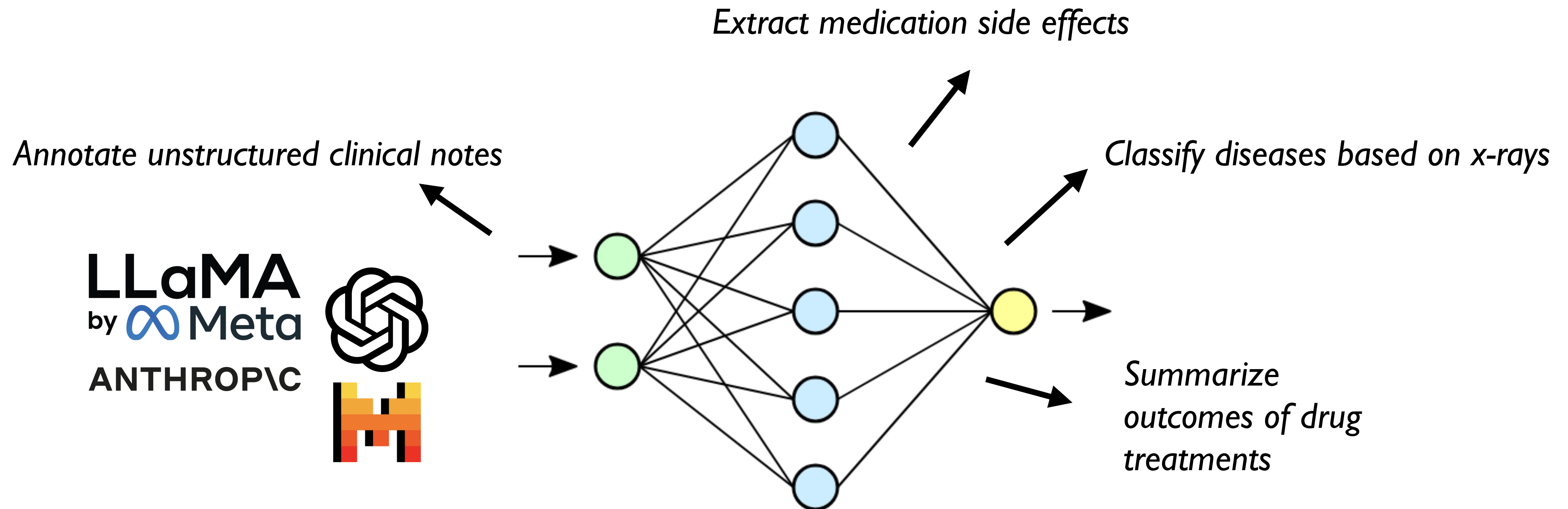


Classify diseases based on x-rays



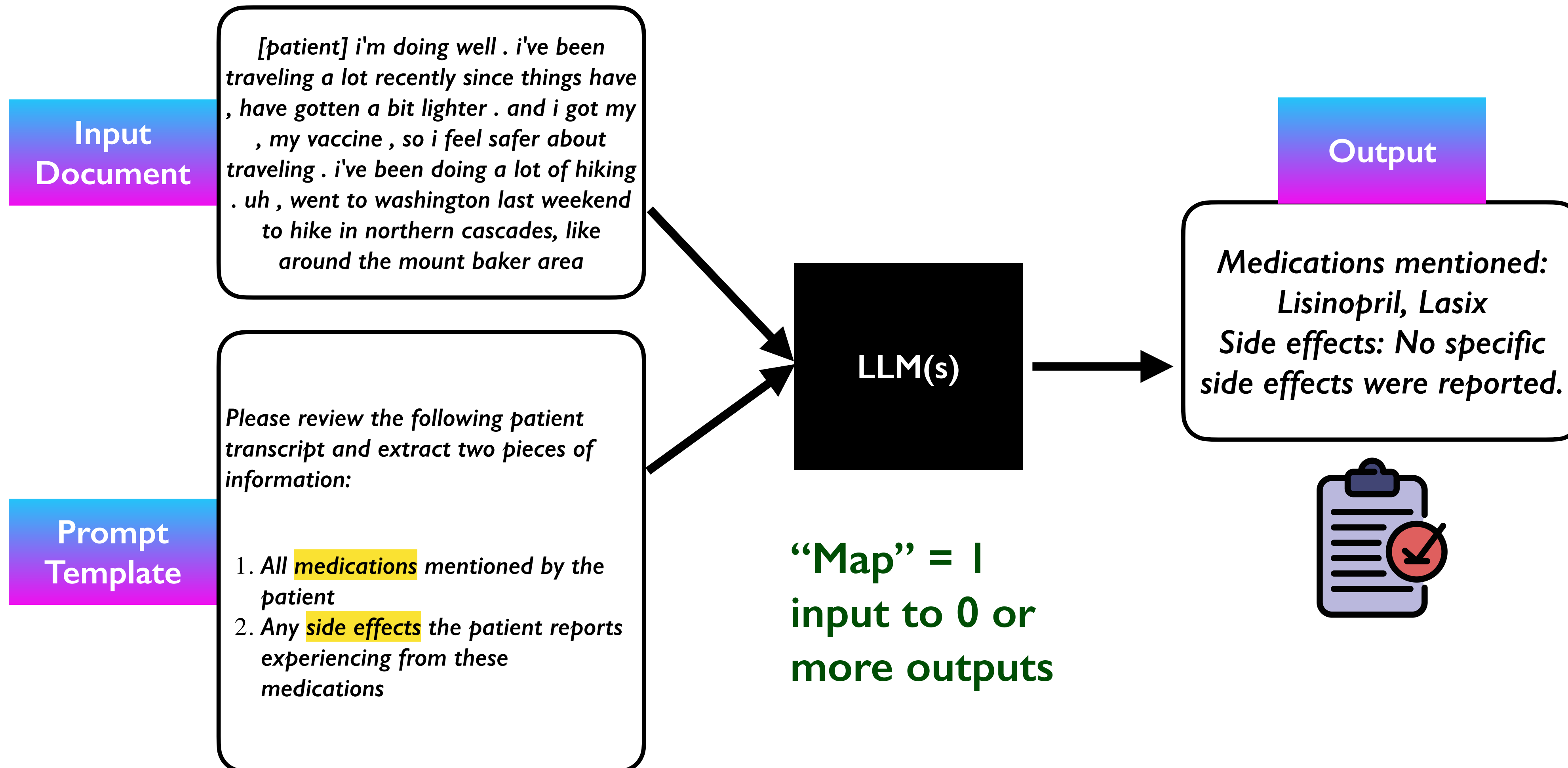
Annotate unstructured clinical notes

# LLMs Enable Unstructured Data Analysis



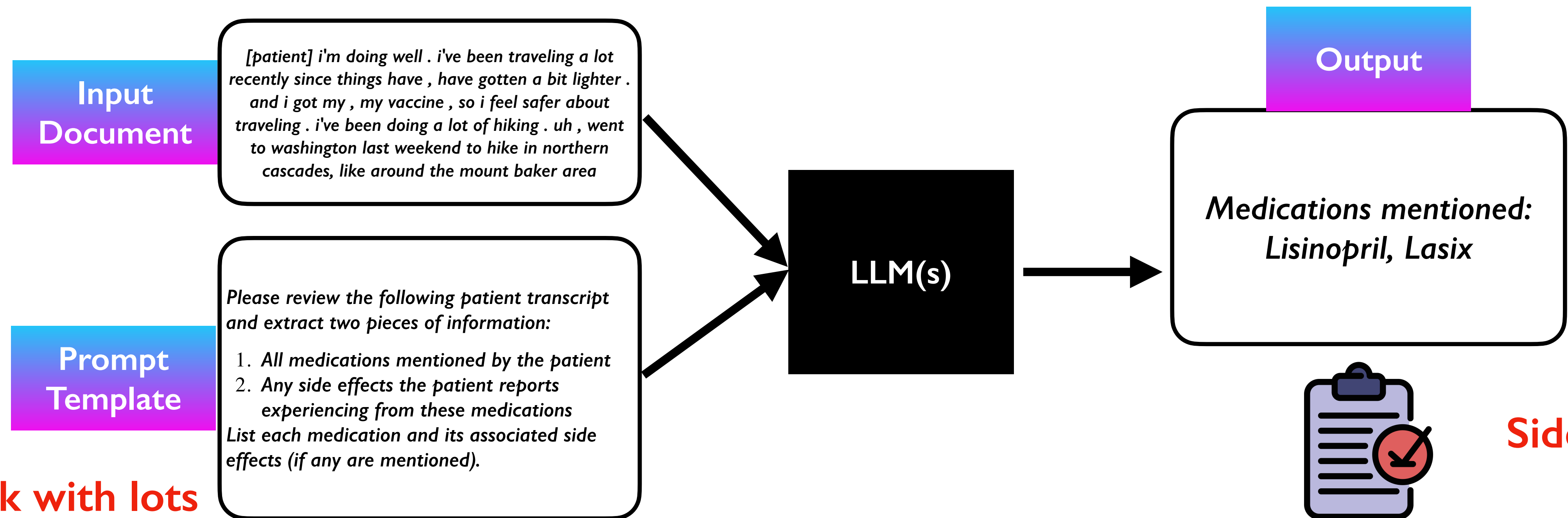
- LLMs enable intelligent data processing pipelines *without training models*
- Significantly simplifies the ML lifecycle

# Example: *Map* Operation on Doctor-Patient Visit Transcripts



# LLMs Make Unpredictable Mistakes

- Hallucinations, bad formatting, ignoring instructions, & more. Tasks can be hard.



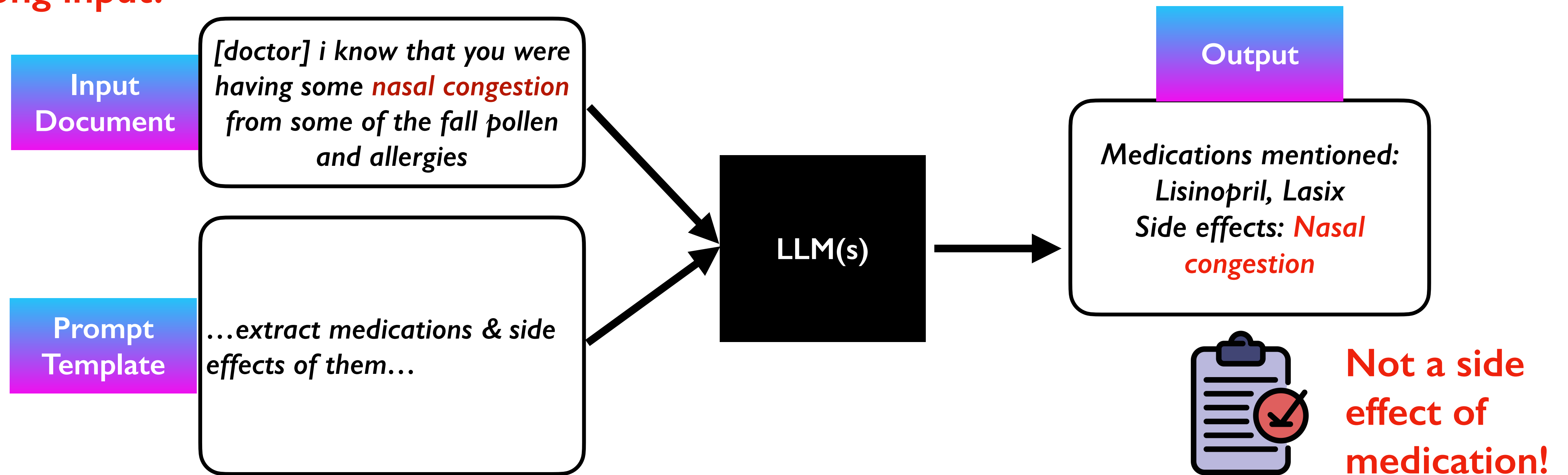
Hard task with lots of instructions?

Side effects?

# LLMs Make Unpredictable Mistakes

- Hallucinations, bad formatting, ignoring instructions, & more. Data can be hard.

Long input?



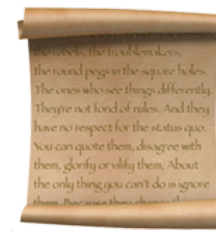


# Challenges in AI-Powered Document Processing

- **Tasks** can be hard. **Data** can be hard. Sometimes both!
- Accuracy also depends on **models**.
- How do we take a complex task on complex documents & break it down?
  - Today: we propose *agentic* rewriting—have an LLM rewrite the pipeline for better accuracy!

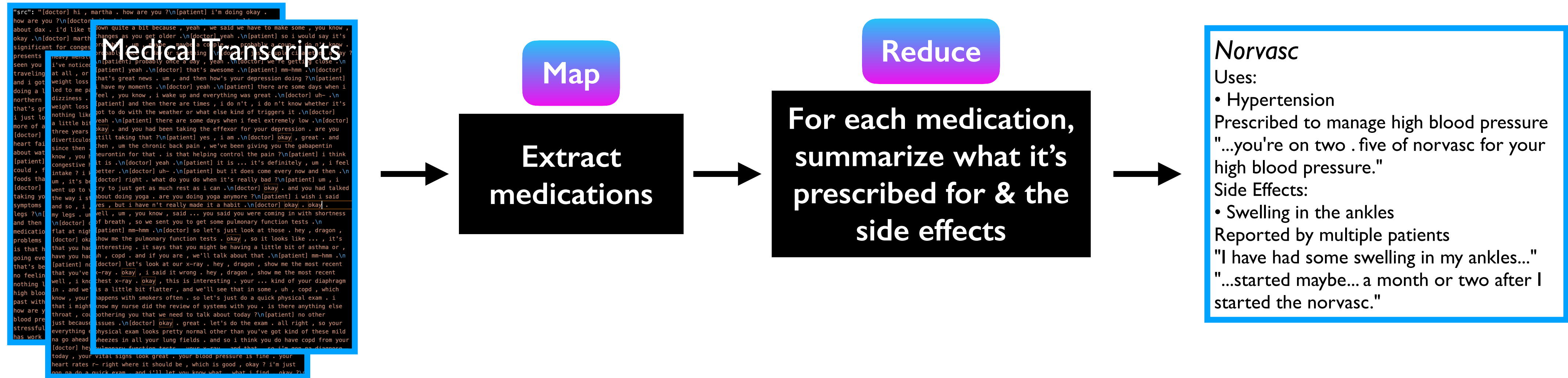






# DocETL

- DocETL ([docetl.org](http://docetl.org)) is a declarative system for optimizing & executing complex document processing pipelines



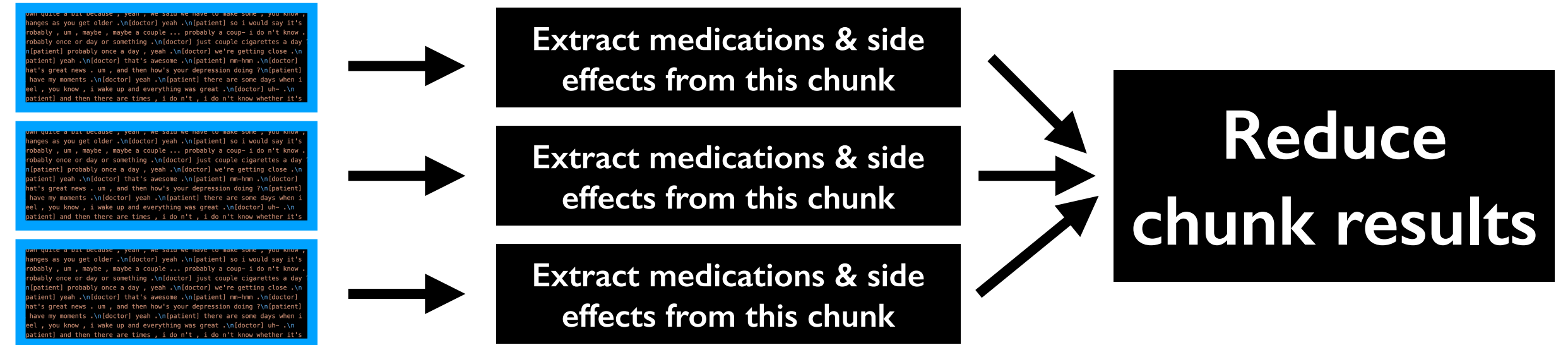
- Declarative = users specify only the ops & DocETL optimizes it to work *correctly & efficiently*
- LLM operators (map, reduce, resolve, filter, equijoin) & non-LLM operators (split, gather, unnest)



# Unclear How To Make Pipelines More Accurate

- For our map operation (extract meds & side effects), we could *rewrite* it many ways. E.g.,

```
son't quite a bit because , yeah , we said we have to make some , you know ,
hanges as you get older .\n[doctor] yeah .\n[patient] so i would say it's
robably , um , maybe , maybe a couple ... probably a coup- i do n't know .
robably once or day or something .\n[doctor] just couple cigarettes a day .
[patient] probably once a day , yeah .\n[doctor] we're getting close .\n
[patient] yeah .\n[doctor] that's awesome .\n[patient] mm-hmm .\n[doctor]
that's great news - um , and then how's your depression doing ?\n[patient]
have my moments .\n[doctor] yeah .\n[patient] there are some days when i
eel , you know , i wake up and everything was great .\n[doctor] uh- \n
[patient] and then there are times , i do n't , i do n't know whether it's
ot to do with the weather or what else kind of triggers it .\n[doctor]
eah .\n[patient] there are some days when i feel extremely low .\n[doctor]
ay . and you had been taking the effort for your depression . are you
till taking that ?\n[patient] yes , i am .\n[doctor] okay , great . and
hen , um the chronic back pain , we've been giving you the gabapentin
eurontin for that . is that helping control the pain ?\n[patient] i think
t is .\n[doctor] yeah .\n[patient] it is ... it's definitely , um , i feel
etter .\n[doctor] uh- \n[patient] but it does come every now and then .\n
[doctor] right . what do you do when it's really bad ?\n[patient] um , i
ry to just get as much rest as i can .\n[doctor] okay . and you had talked
bout doing yoga . are you doing yoga anymore ?\n[patient] i wish i said
es - but i have n't really made it a habit .\n[doctor] okay - okay -
ell , um , you know , said ... you said you were coming in with shortness
f breath , so we sent you to get some pulmonary function tests .\n
[patient] mm-hmm .\n[doctor] so let's just look at those . hey , dragon ,
how me the pulmonary function tests . okay , so it looks like ... , it's
interesting - it says that you might be having a little bit of asthma or
h , copd , and if you are , we'll talk about that .\n[patient] mm-hmm .\n
[doctor] let's look at our x-ray . hey , dragon , show me the most recent
-ray . okay , i said it wrong . hey , dragon , show me the most recent
hest x-ray . okay , this is interesting . your ... kind of your diaphragm
s a little bit flatter , and we'll see that in some , uh , copd , which
appens with smokers often . so let's just do a quick physical exam . i
now my nurse did the review of systems with you . is there anything else
othering you that we need to talk about today ?\n[patient] no other
issues .\n[doctor] okay , great . let's do the exam - all right . so your
physical exam looks pretty normal other than you've got kind of these mild
heezes in all your lung fields . and so i think you do have copd from your
```



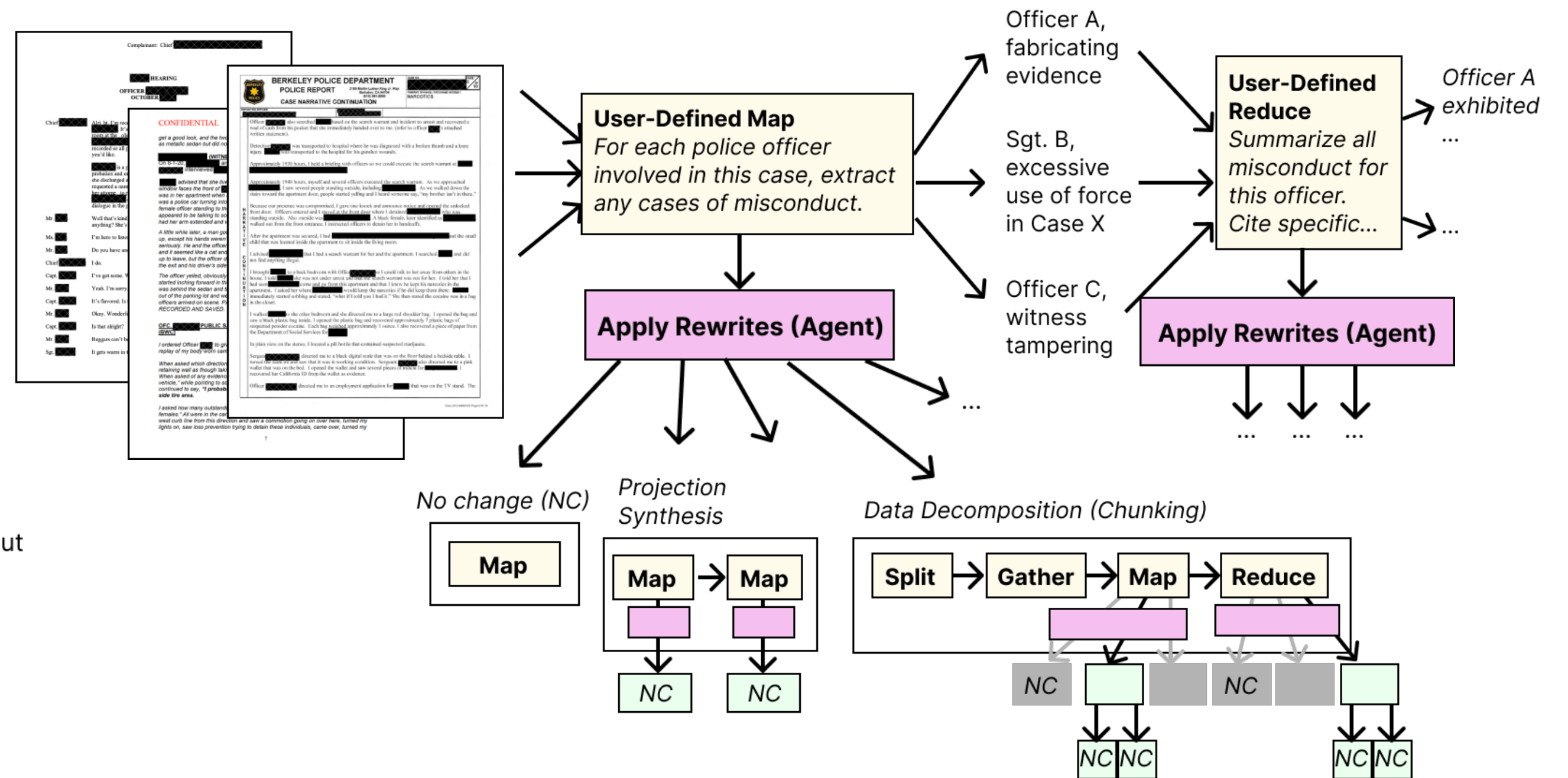
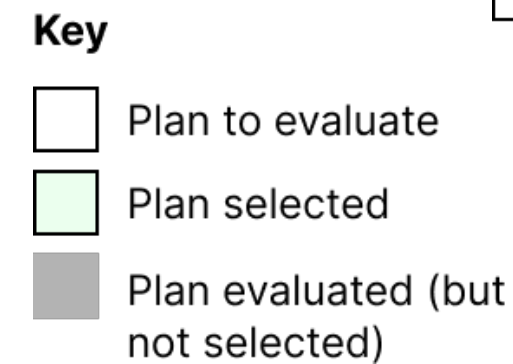
- Chunk up data → apply map to each chunk  
Map task one → map task two  
→ reduce the results

- How do we decompose the task? How do we validate accuracy? Automating this is hard!

# Agentic Optimizer for Intelligent Data Processing

- 3 (new!) ingredients to the optimizer
  - Rewrite directives (implemented by LLM agents)
  - Validation agents to come up with task-specific evaluation criteria
  - Comparing & evaluating candidate plans

*If "no change" is good enough (as determined by an LLM agent), we stop applying rewrites for the relevant operation.*

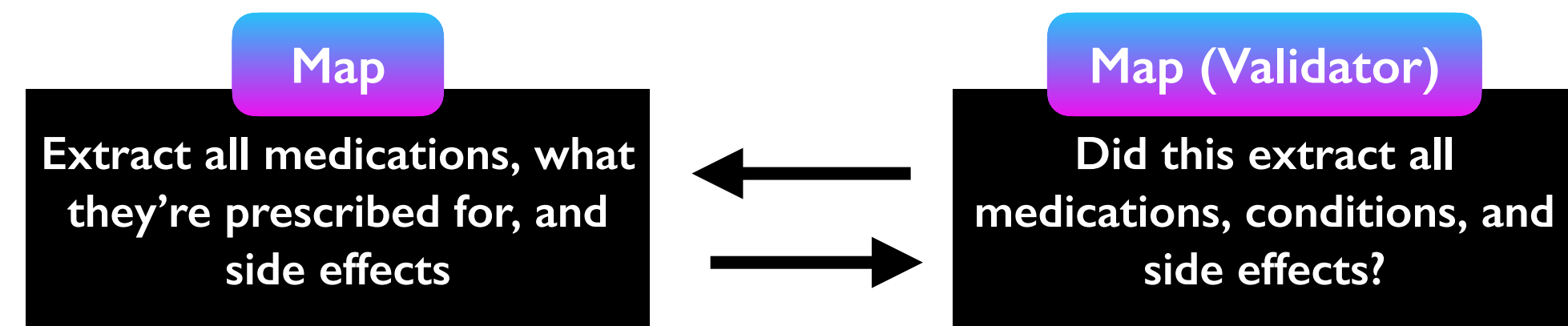
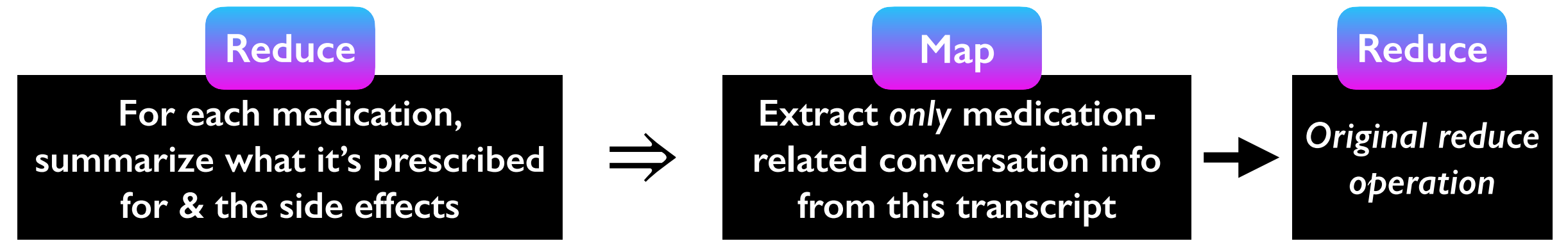
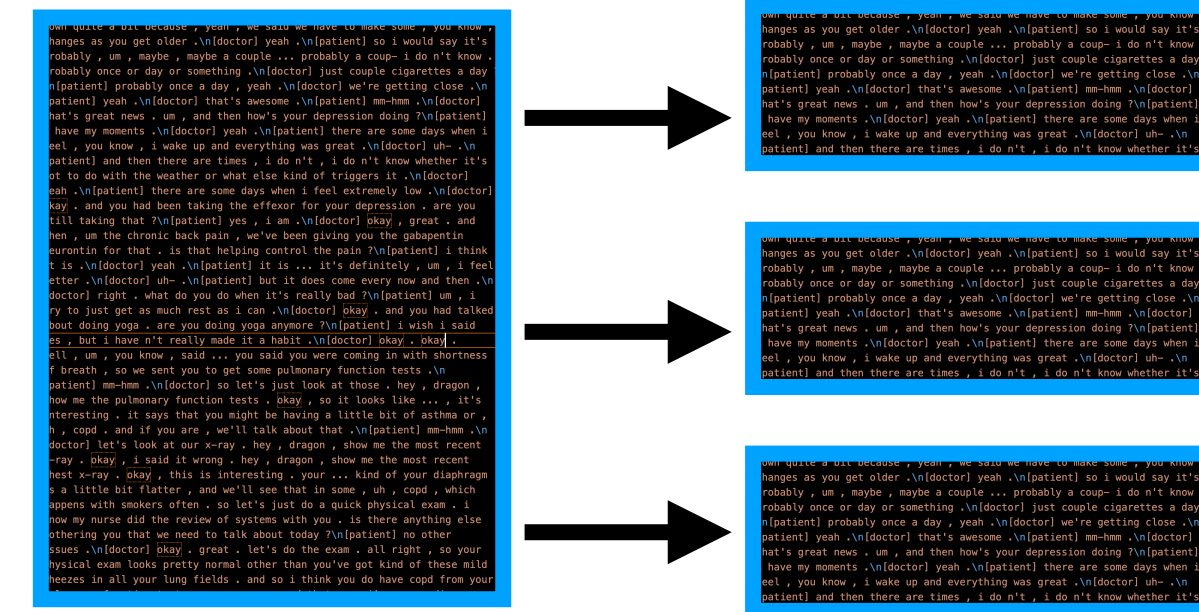


*Optimizing a pipeline for identifying officer misconduct in police records obtained via FOIA requests initiated by journalists at UC Berkeley.*

# Agentic Optimizer

## I. Generating Plans with 13 Rewrite Directives

- **Data decomposition:** *breaking down the unit of data fed into the LLM call*
  - Document chunking (map)
  - Multi-level aggregation (reduce)
- **Projection synthesis:** *breaking down the task (i.e., prompt) into helpful intermediate step(s)*
- **LLM-centric improvements:** iterative refinement (i.e., looping until the output is good), duplicate key/entity resolution



How to come up with these validators?  
More generally: how do you pick the right plan?



# Agentic Optimizer

## 2. Validation Agents

- In addition to generation, agents also **validate** plans
- To validate an operator, we use LLM agents to:
  - Synthesize validation criteria
  - Evaluate a candidate plan's sample outputs given this criteria

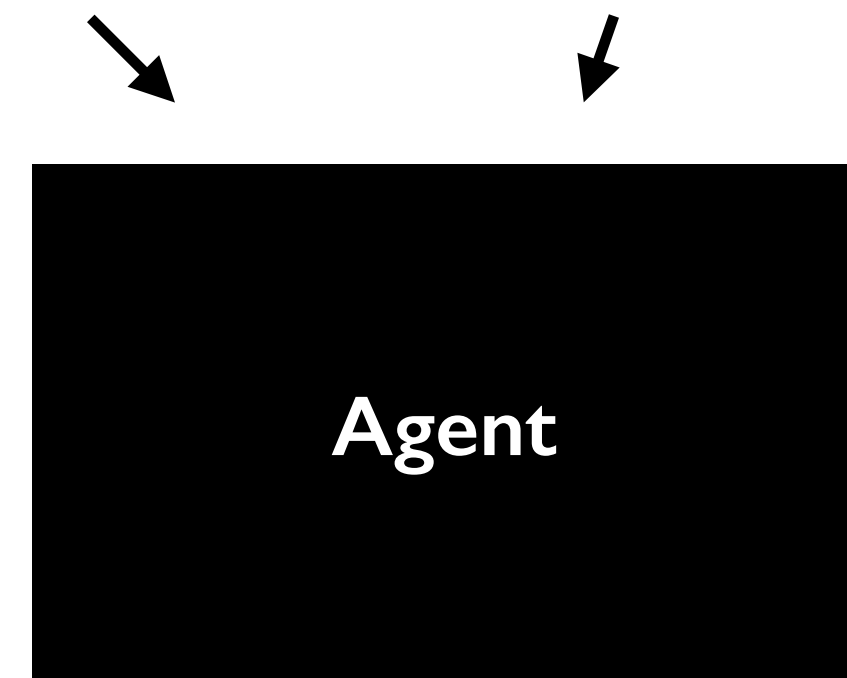
Example rubric:

- Are *all* medications in the input reflected in the output?
- For each medication, are its side effects reported?
- Are there medications or side effects in the output not in the input document?

Sample inputs & outputs of unoptimized operation



Operation prompt  
(list all medications  
and their reported  
side effects)



↓  
Rubric

# Agentic Optimizer

## 3. Evaluating (Ranking) Candidate Plans

- Picking the best plan involves:
  - Rating each plan's output on 1 (bad) to 5 (good) scale
  - Pairwise comparisons of the top k plans
- Top-performing plan is listed 11th!
- Coarse-grained ratings are not perfect, but they are scalable in  $O(n)$
- Pairwise comparisons are too expensive in  $O(n^2)$
- Best of both worlds:  $O(n) + O(k^2)$

```
Comparison Details:
Input 1: "plan_1" because Plan 1 output provides a clear and concise extraction of the medications, their purposes, and side effects mentioned in the transcript, without unnecessary duplication. In contrast, Plan 2 repeats entries for each medication multiple times with slight variations, leading to redundancy and potential confusion. Plan 1's approach is more organized and aligns better with the task requirement to list medications and their details accurately and succinctly.
Input 2: "plan_2" because Plan 2 provides a more comprehensive output by including both the ibuprofen and wrist splint in the list of interventions for carpal tunnel syndrome. The task prompt asks for a list of medications and their uses, and while the wrist splint is not a medication, it is a significant part of the treatment plan discussed in the conversation. Therefore, Plan 2 captures a fuller picture of the treatment strategy outlined in the transcript.
Input 3: "plan_2" because Plan 2 performed better because it identified additional medications mentioned in the transcript, specifically the MMR vaccine and the Shingles vaccine, which were both overlooked by Plan 1. Plan 2 also provided a more detailed explanation of the vitamin D3 dosage. Both plans maintained a clear format, but Plan 2 had a more comprehensive extraction of medications discussed in the conversation.
Input 4: "plan_2" because Plan 2 is better because it captures both medications mentioned in the conversation: Tylenol and the potential steroid injection. It also provides more detailed descriptions of their uses and effects. Plan 1 only includes Tylenol and misses the mention of steroid injection, which is a significant consideration for the patient's treatment plan.
Input 5: "plan_2" because Plan 2 provides a more detailed extraction of the medication use and patient adherence to each, which aligns well with the requirement to list medications along with their uses. It also captures the dosage of Prozac and the patient's stability on it, offering a more comprehensive view of the medication management. Plan 1 is accurate but less detailed compared
```

*“Plan 2 provides a more comprehensive output by including both the ibuprofen and wrist splint...”*

```
optimizer.py:359
optimizer.py:376
```

Plan	Score	Runtime	Pairwise Wins
no_change	3.80	8.45s	5
gleaning_1_rounds	3.40	4.04s	6
chunk_size_190_peripheral_previous_tail_2_full_previous...	3.00	9.67s	4
chunk_size_549_peripheral_previous_tail_1_full	3.00	4.75s	8
chunk_size_909_peripheral_previous_tail_1_full	3.00	3.92s	3
chunk_size_190_peripheral_previous_tail_1_full_previous...	2.80	8.54s	4
chunk_size_190_peripheral_previous_tail_1_full_previous...	2.80	7.60s	3
chunk_size_190_peripheral_previous_tail_1_full	2.80	5.66s	2
chunk_size_190_peripheral_previous_tail_1_full_next_he...	2.80	5.18s	1
chunk_size_909_peripheral_previous_tail_1_full_next_he...	2.80	3.48s	2
chunk_size_1269_peripheral_previous_tail_1_full	2.80	2.48s	9
chunk_size_190_peripheral_previous_tail_2_full_next_he...	2.60	8.99s	0
chunk_size_190_peripheral_previous_tail_2_full_previous...	2.60	6.73s	0
chunk_size_549_peripheral_previous_tail_1_full_next_he...	2.60	6.56s	0
chunk_size_190_peripheral_previous_tail_2_full	2.60	6.46s	0

```
optimizer.py:377
Choosing chunk_size_1269_peripheral_previous_tail_1_full for operation extract_medications (Score: 2.80, Runtime: 2.48s)
optimizer.py:380
```

# Towards **Agentic** Data Processing

This is more than just optimizing accuracy...

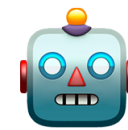
- Agent = LLM executing on behalf of human
- We are in a new world of...
  - Agents looking directly at the data
  - Agents synthesizing information across document boundaries
- Catch-22: Users need to see the outputs of an operation before they write the operation

For each side effect reported, summarize all the medications and illnesses associated



**Dizziness: ...**

**Stomach upset...**

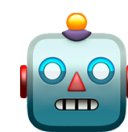


Actually, first, categorize the side effects by *physiological system and severity*. Then...



**Digestive system:**

**Light nausea: ...**

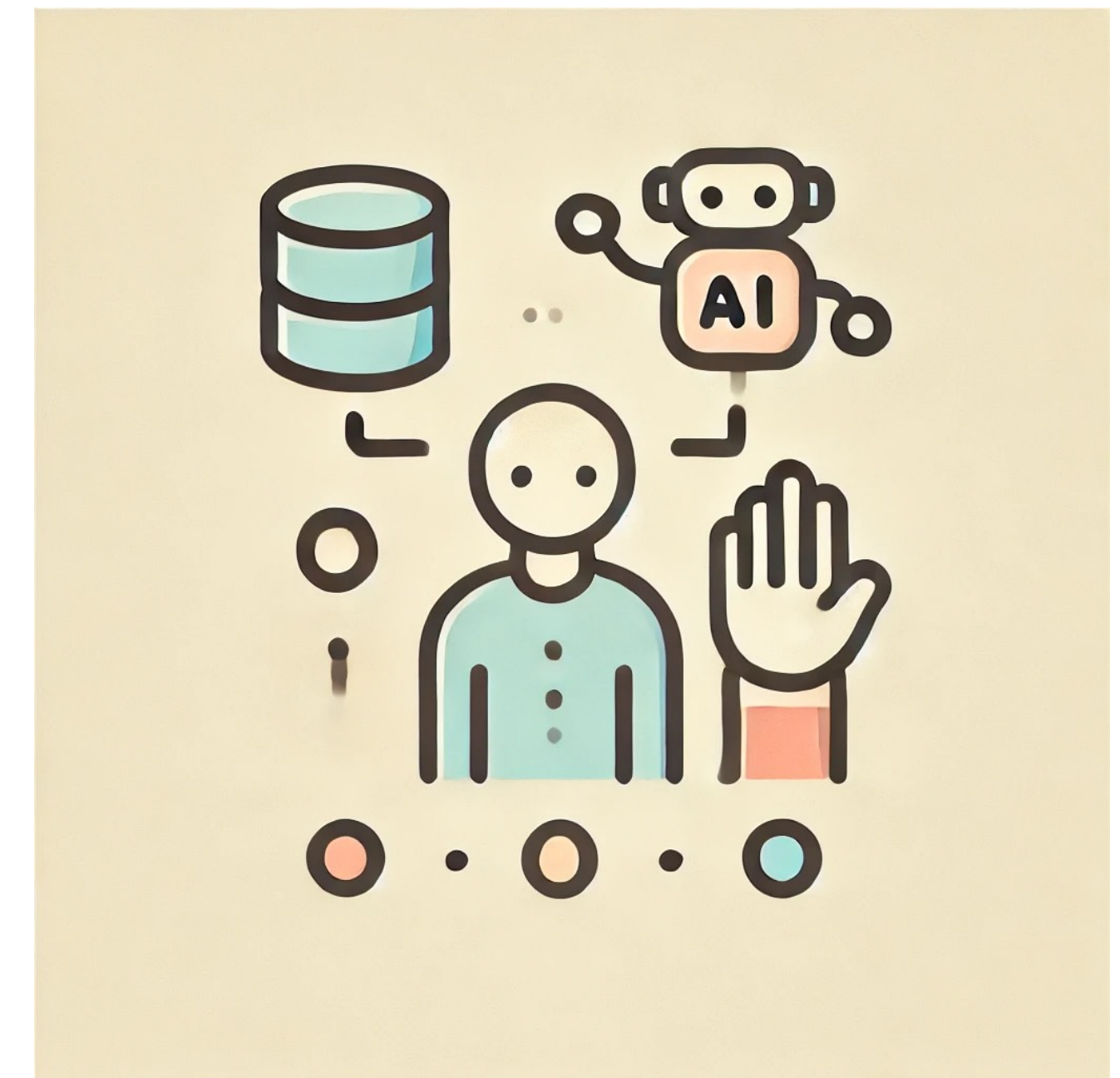




# Towards **Agentic** Data Processing

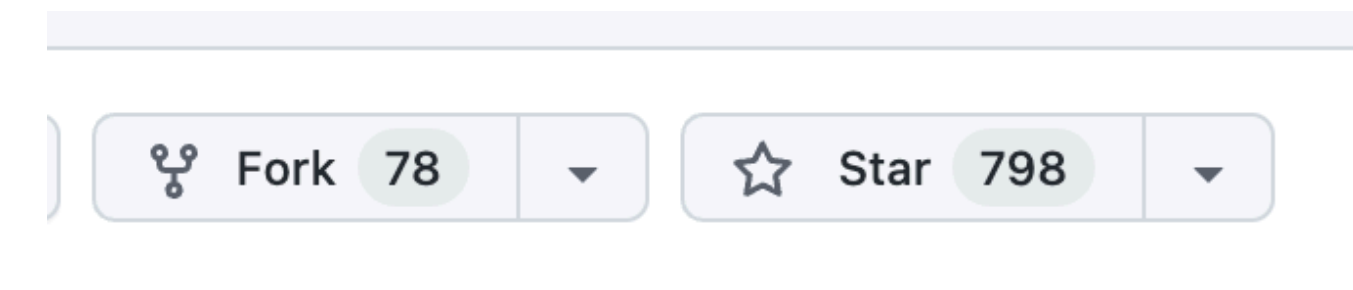
## Roles of Humans; Role of Agents

- What should the human do? *Steer the artifact towards their goal*
- What should the optimizer do? *Solve the steered (sub)-task as accurately and efficiently as possible*
- DocETL pipelines are low-code (written in YAML) and optimized interactively in the terminal 🖥️
- Towards no-code
  - Natural language specification of pipelines 🧑
  - Interactive optimization in a web app 🌐



# Growing Traction

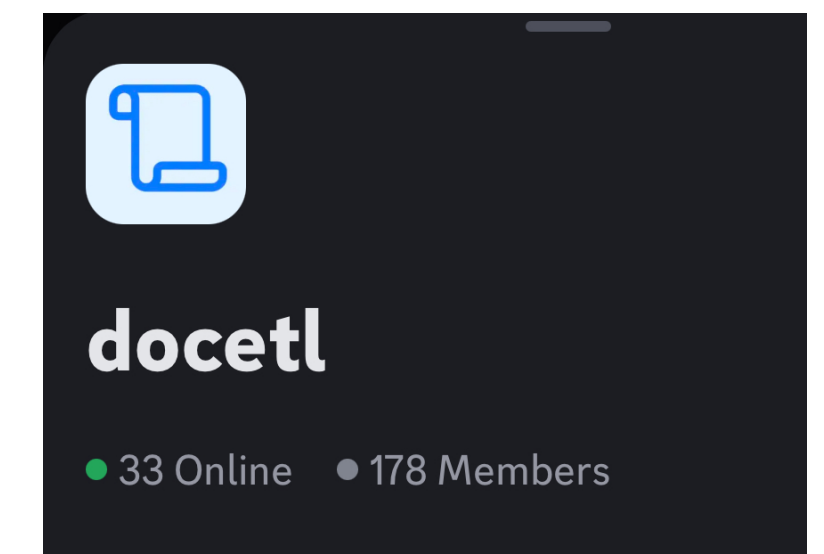
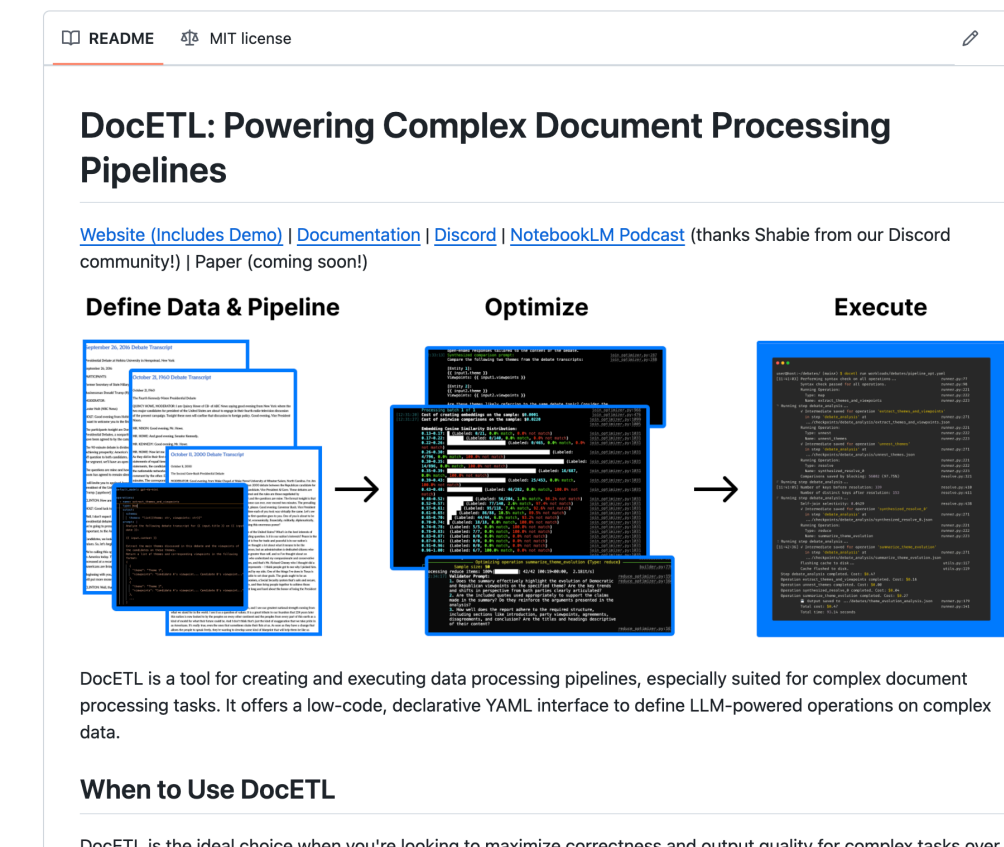
- Launched 14 days ago!
- Some use cases so far
  - Civic engagement: reports on why & how council members voted on issues
  - Email analysis
  - Forensic psychiatry
  - Mining law articles
  - Mining climate reports to make policy recommendations
- Please join our community! Always looking for contributors & users.
- Exciting research & engineering ahead!



## About



A system for agentic LLM-powered data processing and ETL



# Takeaways

- Building data processing pipelines around LLMs requires **experimentation** and **validation**
- Optimizers have to be human-centered and agentic!
- DocETL is a low-code, declarative LLM-powered data processing system ([docetl.org](https://docetl.org))

 [sh-reya.com](https://sh-reya.com)  [@sh\\_reya](https://twitter.com/sh_reya)  [shreyashankar@berkeley.edu](mailto:shreyashankar@berkeley.edu)



 COLUMBIA UNIVERSITY  
IN THE CITY OF NEW YORK

EPIC  
DATA lab  
UC Berkeley