

LLM agents that **interview the researcher first** write a legal extraction schema that covers more of an expert's questions than the other experts do.



Schematize: An Agentic System for Generating and Refining Information-Extraction Schemas for Legal Research

Albert Sawczyn[†] · Jakub Binkowski · Kamil Tagowski · Łukasz Augustyniak · Berenika Kaczmarek-Templin · Tomasz Kajdanowicz
Wrocław University of Science and Technology [†] albert.sawczyn@pwr.edu.pl

■ A real run: from a two-sentence idea to a tested schema

1 The idea

HUMAN IN THE LOOP

👤 **Legal researcher:** "I would like to study **personal-rights violation** cases and assess **how severely** the perpetrator violated them."

Two sentences. That is all the system gets. Scope, law, variables and the real hypothesis are still in the researcher's head.

2 The interview

INTERVIEW AGENT

🔍 **How do courts weigh severity?**

👤 Not really. Are cases getting more trivial year by year?

🔍 **Severity criteria: reach, intent...?**

👤 Above all what the perpetrator did or said. Rate severity 0-5.

🔍 **Separate online and workplace cases?**

👤 By place, with social media as its own field.

3 First draft

CBR LOOP 5 FIELDS

```
violation_type: enum  
[reputation, privacy, image]  
severity: int 0--5  
court_found_violation: bool  
compensation_awarded: bool  
compensation_amount: float
```

Drafted from the problem definition, then criticised: clear? complete? extractable? No agent sees documents yet.

4 Final schema

DGR LOOP +5 FIELDS

```
violation_type: enum  
severity: int 0--5  
court_found_violation: bool  
compensation_awarded: bool  
compensation_amount: float  
+ perpetrator_act: text  
+ violation_place: enum  
[public, private, workplace]  
+ social_media: bool  
+ legal_basis: enum  
[art. 23, art. 24 KC, other]  
+ judgment_year: int
```

Green: added after real rulings. Illustrative until the real run.

■ How it works

1 Interview **HUMAN**

A helper agent asks; a definer agent formalises.

2 Query generation

Queries fetch the rulings to apply the schema to.

3 Draft and critique **CBR**

Generator, assessment agent, refiner; document-blind.

4 Reality check **DGR**

Real rulings expose what the schema cannot capture.

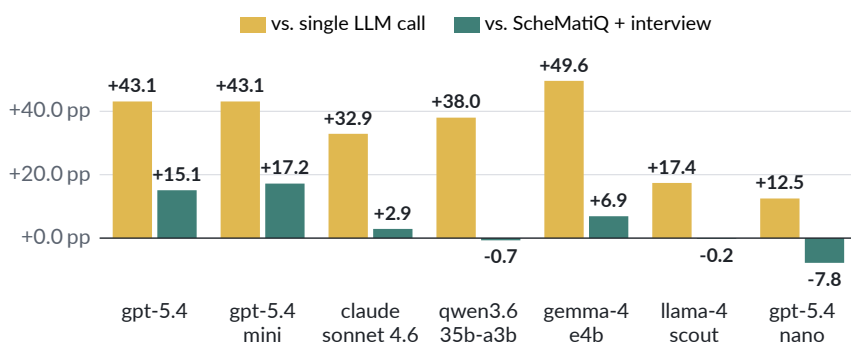
5 Summary and chat **HUMAN**

Every change explained; edits in plain language.

■ How we checked

Experts never see a schema. 7 legal advisers wrote the questions the data must answer (3 Polish problems, about 29 each). An LLM judge checks which questions a schema covers. Inter-expert coverage: **53.8 %**.

■ Does it work? Coverage gain of Schematize, in pp



■ Takeaways

4 of 7

backbones: beats the other experts' coverage.

6 of 7

backbones: best or second-best.

-14.3 pp

without the interview stage.

\$0.02

per problem with a small open model.

The library

pip install schematize

Any LLM. Any LangChain chat model or OpenAI-compatible endpoint, e.g. via LiteLLM.

Your data. HuggingFace and Weaviate retrievers built in, or one async method.

Typed output. Pydantic schemas with field names, types, descriptions and enums.

Evaluate and tune. Built-in coverage evaluator; refinement depth is configurable.



Wrocław University
of Science and Technology

Code & paper →

github.com/pwr-ai/schematize · pwr-ai.github.io/schematize · albert.sawczyn@pwr.edu.pl

