

Practical **Things** about Productizing **LLM Agent** System



Lele Cao

Thanks to



About Lele

Senior Principal AI/ML Researcher
Research Lead
@ King/Microsoft



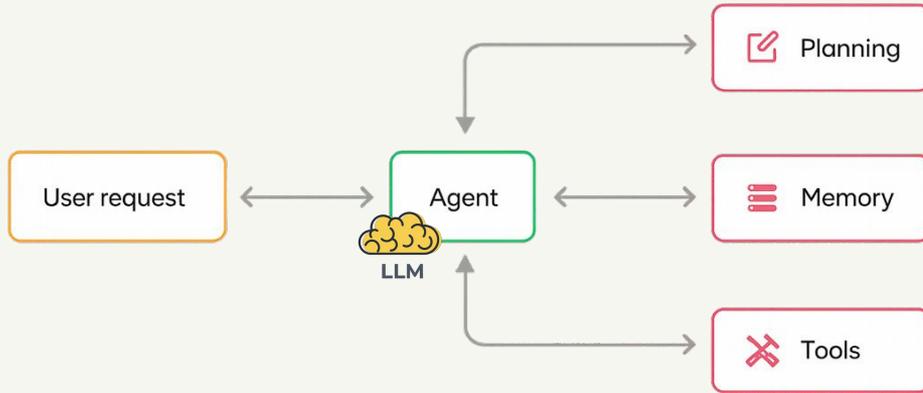
Industrial Part

- 16-year in King/Microsoft, EQT Motherbrain, Alibaba, Elisa and Ericsson.
- AI/ML Engineer, Data Scientist, Research Engineer, Software Engineer, Product Designer, Engineering Manager, General Manager.

Academic Part

- 4-year BS in Software Engineering
- 2-year MS in Interactive Systems Design
- 4-year PhD in AI and Robotics
- 2016: Almost started a Professorship career.
- >50 publications on deep learning, graph learning, timeseries, language model, robotics, reinforcement learning, computer vision, explainability ...

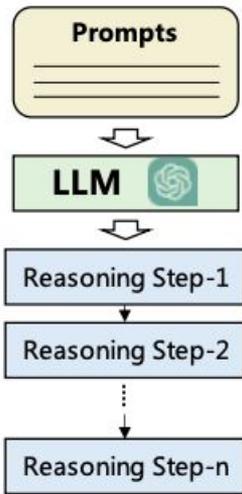
LLM Agent



A system powered by a Large Language Model (LLM) that goes beyond simple text generation to perform **goal-oriented** and **multi-step** tasks by planning, reasoning & interacting with **tools** and/or **data**.

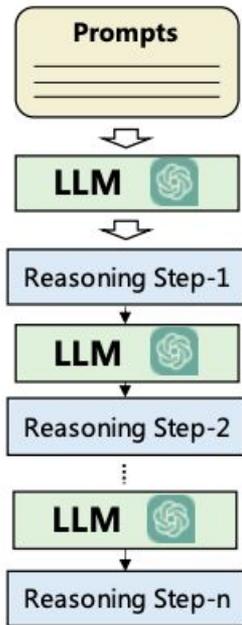
Planning & Reasoning

CoT , Zero-shot Cot

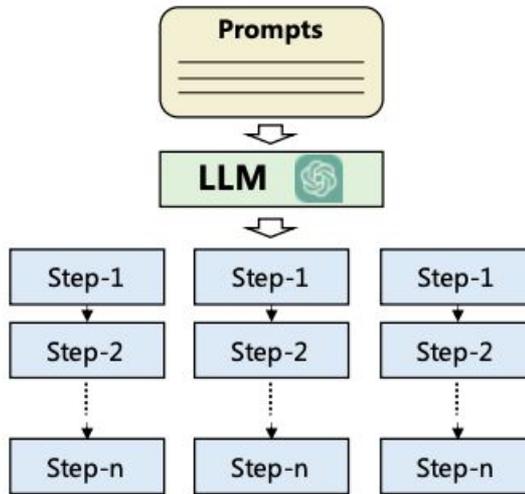


Single-Path Reasoning

ReWOO , HuggingGPT

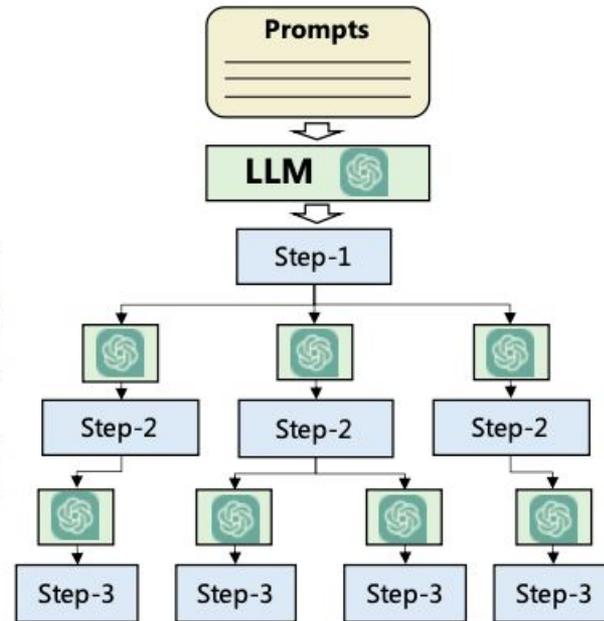


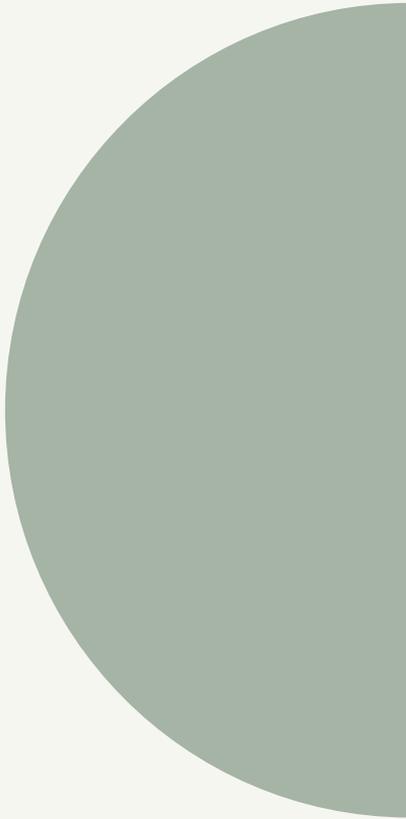
CoT-SC



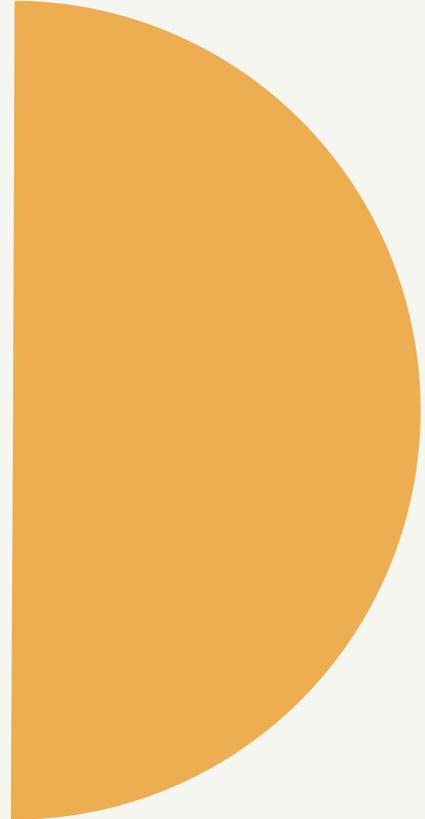
Multi-Path Reasoning

ToT , LMZSP , RAP





"When the steps
are predefined
(*not determined by
LLMs on the fly*),
it's an **Agent
Workflow**."



Lele Cao

Where my learnings come from

<https://cspaper.org>

AI agent system to help researchers to obtain fast and conference-specific review for their research papers.



A Short Paper:

[CSPaper Review: Fast, Rubric-Faithful Conference Feedback](#)

CSPaper.org Beta

My Reviews

YOUR REVIEW

Desk Rejection Assessment:

Paper Length
Topic Compatibility
Minimum Quality

Expected Review Outcome:

Summary
Strengths and Weaknesses
Quality
Clarity
Significance
Originality
Questions
Ethical concerns
Limitations
Overall
Confidence

How a Critical Reviewer Might Rate

Quality
Clarity
Significance
Originality
Overall
Confidence

Sign out

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones Polosukhin

Reviewed on June 29, 2025 [View Paper](#)

Reviewed as if submitted to NeurIPS 2025 - ma

Desk Rejection Assessment:

Disclaimer: Anonymity and formatting are not checked. This assessment focuses on correctness, novelty, clarity, and scientific quality per NeurIPS desk rejection criteria.

Paper Length

Pass

Topic Compatibility

Pass

The paper presents a new sequence transduction model for applications in natural language processing (machine translation, parsing). These subjects fall squarely within the scope of NeurIPS, including machine learning, deep learning architectures, NLP, and evaluation.

Minimum Quality

Pass

All required major sections are present: Abstract, Introduction, Model Architecture (including Experiments/Tables of quantitative results), Discussion/Conclusion. The paper provides empirical and comparative evidence, describes a complete and original method, and shows improvements over prior work. No major methodological or clarity flaws are present.

Expected Review Outcome:

Disclaimer: Only the main paper is subject to review, in accordance with the NeurIPS Track.

Summary

Practical Things



Start with stateless agent workflow

Tools unlock true LLM-agnostic system

Code first. LLM when necessary

Optimize task density per token

Affordance: not every app should be a chat

How does a **stateful multi-round** chat remember things?

A

The LLM itself is updated with all historical information gradually.

B

Things are distilled into compact knowledge-base and becomes a part of LLM.

C

Things are stored in DB as is, and LLM knows how to pick relevant info.

D

All historical things always are sent as they are cumulatively to new chat rounds.



How does a **stateful multi-round** chat remember things?

A

The LLM itself is updated with all historical information gradually.

B

Things are distilled into compact knowledge-base and becomes a part of LLM.

C

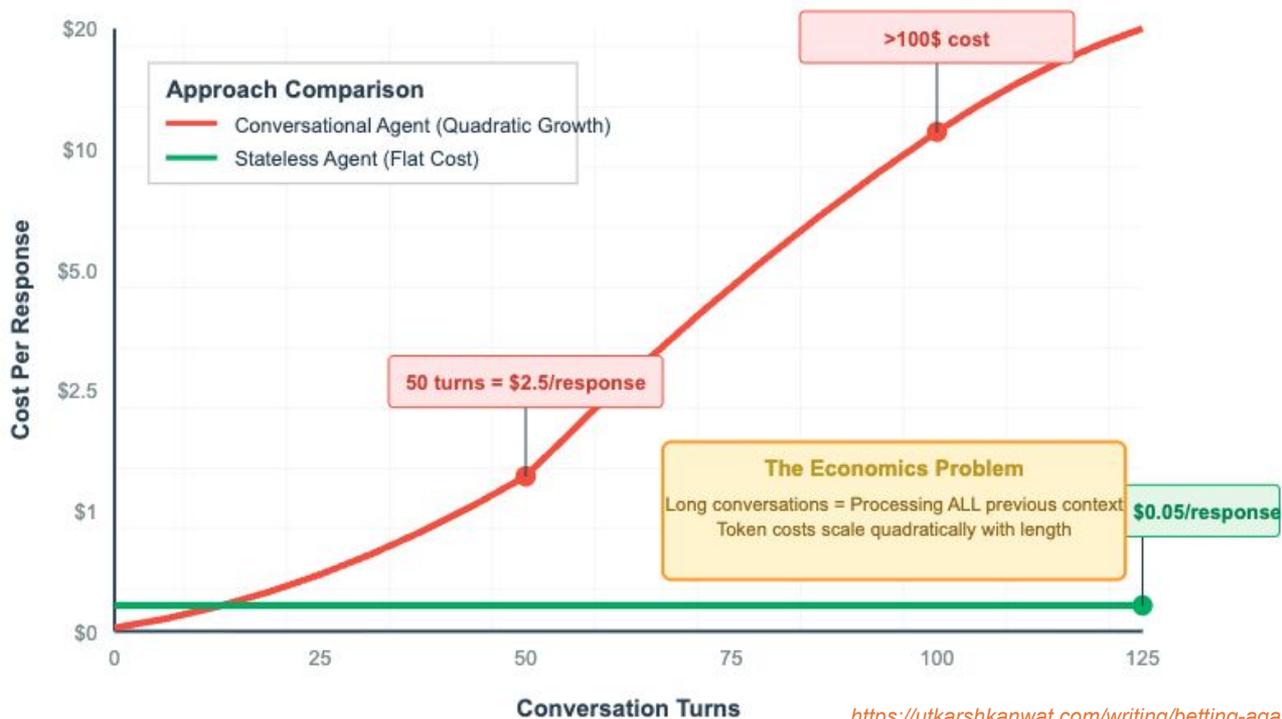
Things are stored in DB as is, and LLM knows how to pick relevant info.

 D

All historical things always are sent as they are cumulatively to new chat rounds.



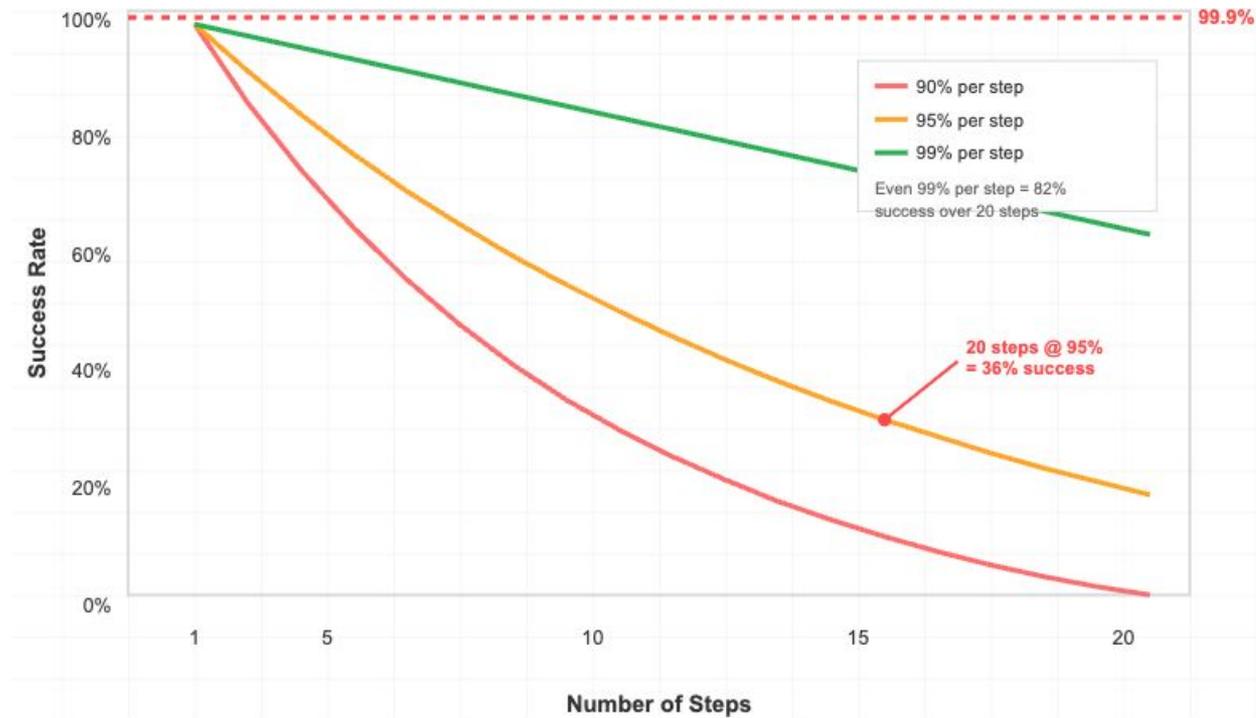
Non-sustainable token economics



Start with stateless agent workflow

Errors compound exponentially

If you must use multi-step agent, make sure you **capture** (be it automatically or manually) and **quarantine** errors in each step!

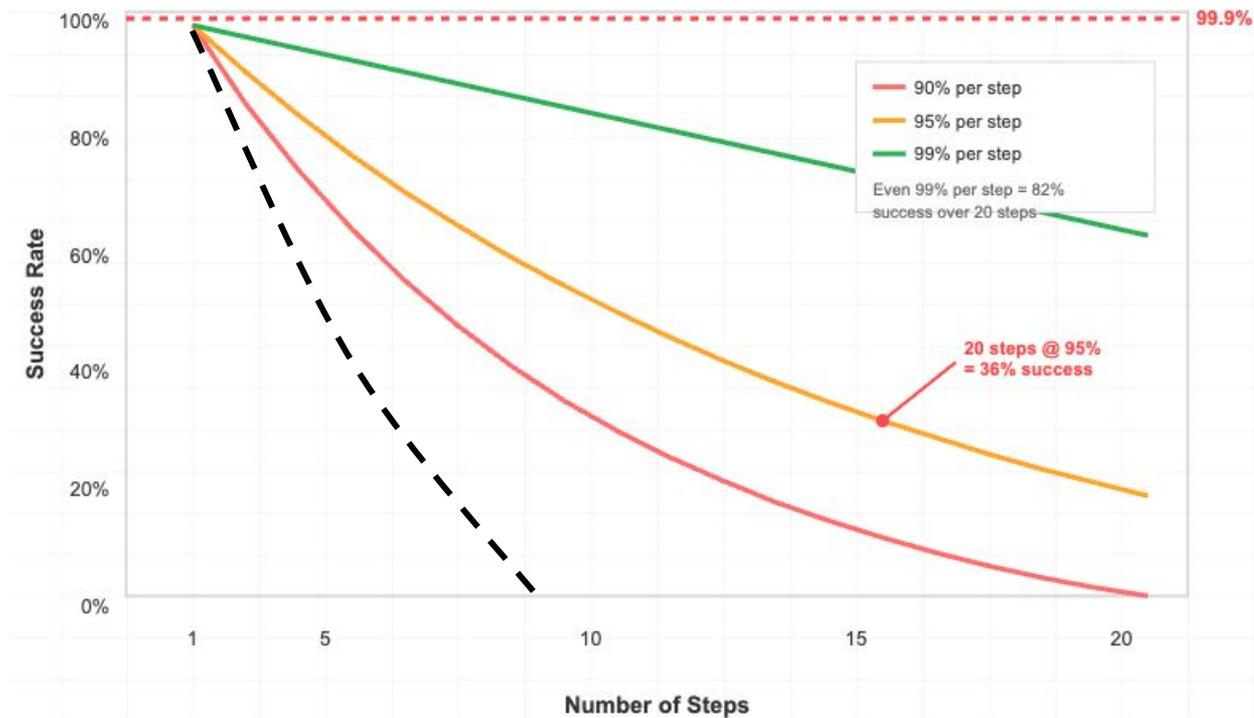


Long Context \neq Good Quality

Poisoning

Distraction

Confusion



Start with stateless agent workflow

Implications

- Minimize the times/steps you need LLM, preferably only once.
- Use agent workflow, especially when you know the problem domain inside-out.
- Make your agent/steps stateless as much as possible.
- Only provide your agent directly relevant and verified information.
- Ofc, you can mix-and-match!



Practical Things

Start with stateless agent workflow



Tools unlock true LLM-agnostic system

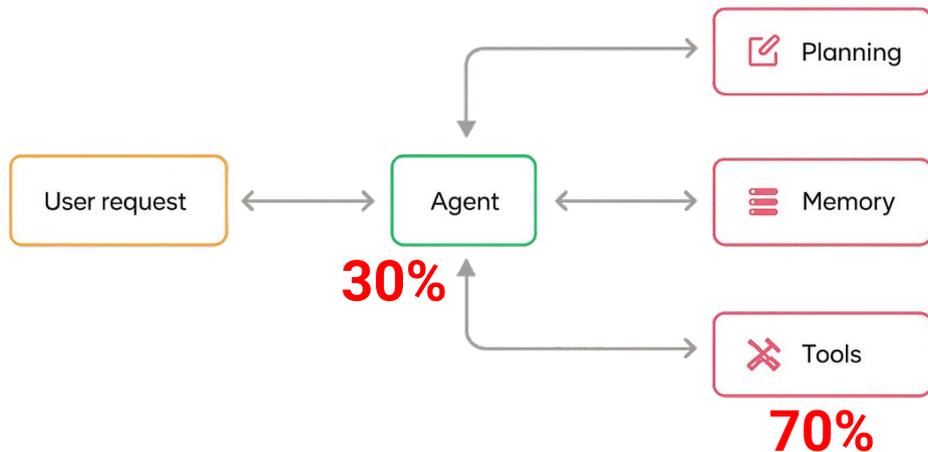
Code first. LLM when necessary

Optimize task density per token

Affordance: not every app should be a chat

The success mostly on tools

- With input from great tools, you can choose pretty much any established LLMs without significant performance decay.



But, tool overload is also real

- More tools in the prompt → worse tool selection. Small models call irrelevant tools more often.
- Dynamic tool loading: rank/select tools per turn (e.g., semantic lookup over tool descriptions)



Practical Things

Start with stateless agent workflow

Tools unlock true LLM-agnostic system



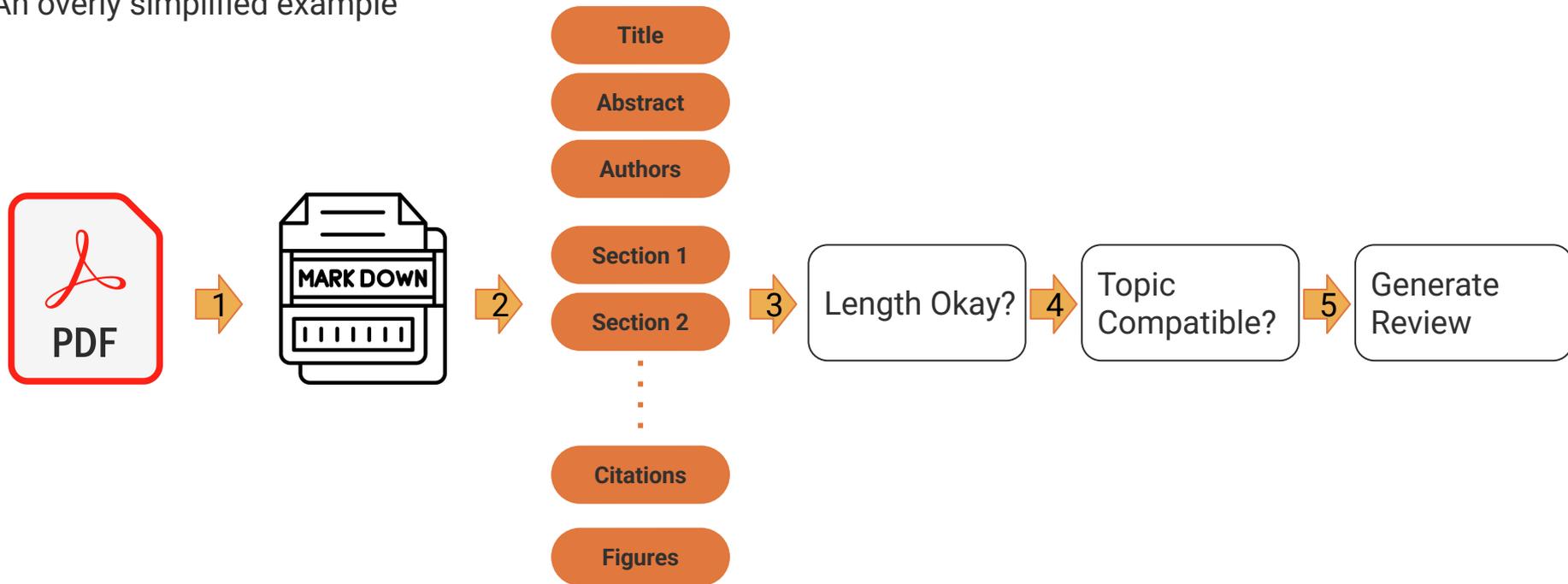
Code first. LLM when necessary

Optimize task density per token

Affordance: not every app should be a chat

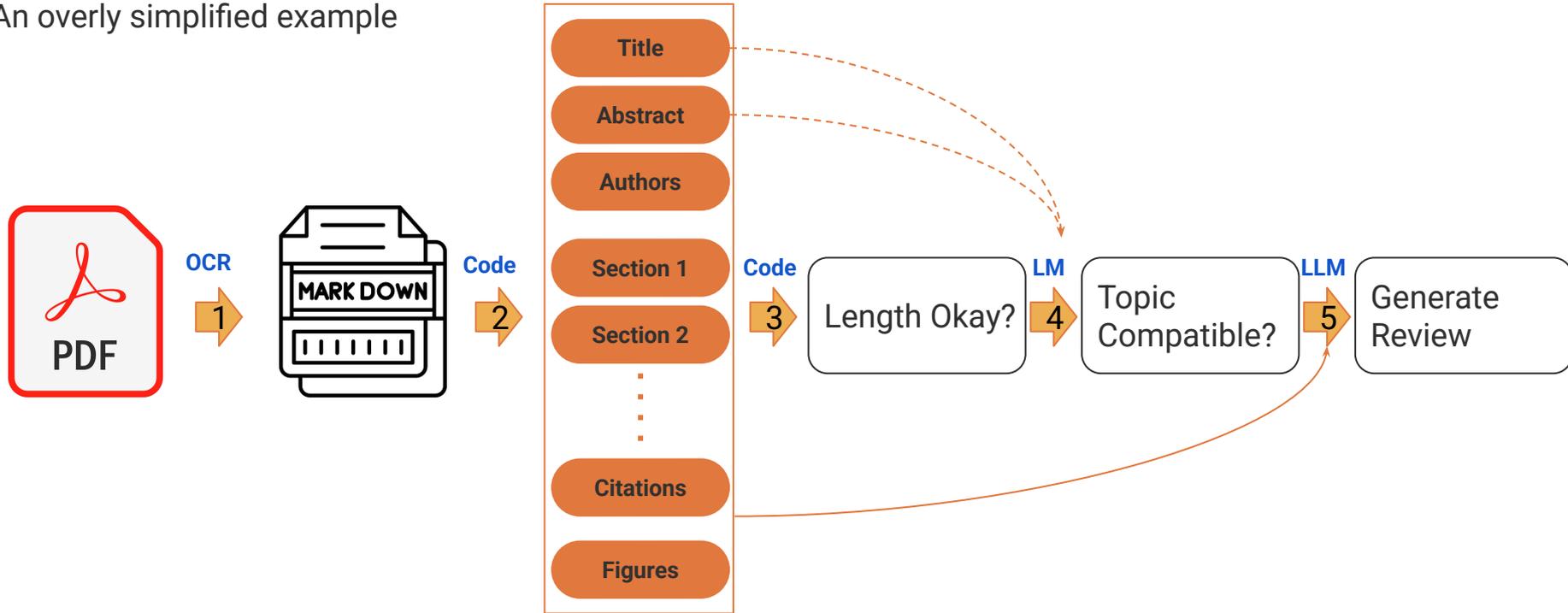
Which steps have to use LLM?

An overly simplified example



Which steps have to use LLM?

An overly simplified example



Code first. LLM when necessary

Practical Things

Start with stateless agent workflow

Tools unlock true LLM-agnostic system

Code first. LLM when necessary



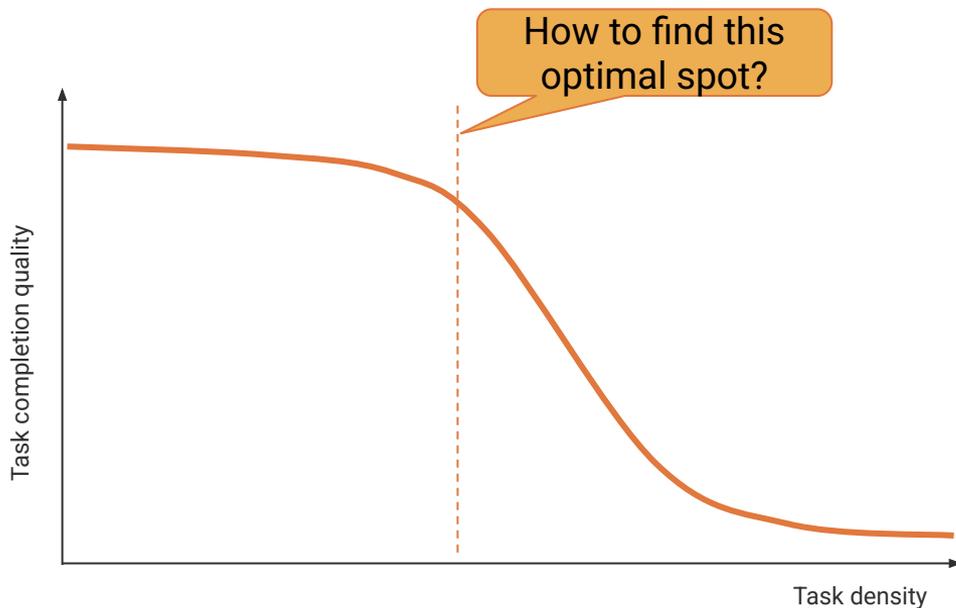
Optimize task density per token

Affordance: not every app should be a chat

Task density vs. completion quality

Task density measures how much **intended** reasoning, problem-solving, or challenge is packed into each unit of computational resource (e.g., tokens consumed).

$$\text{TaskDensity} = \frac{\text{Task Challenge or Cognitive Load}}{\text{Tokens Consumed}}$$



Optimize task density per token

Benchmarking – an example

Generate
Review

- A. Trigger one LLM call for each paper section
- B. Call LLM only once providing all sections

Task Density	Task completion quality (MAE)	# Tokens per paper
A (low)	0.12	~100k
B (high)	0.15	~10k



Optimize task density per token

Practical Things

Start with stateless agent workflow

Tools unlock true LLM-agnostic system

Code first. LLM when necessary

Optimize task density per token



Affordance: not every app should be a chat

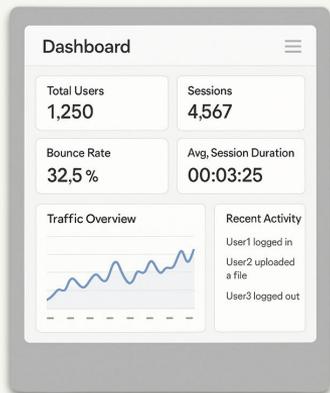
Don't craft a hammer without a nail

- Make your design naturally imply its intended usage – minimize the need of educating your user or forcing them read manual!
- Support safe exploration – free chat is not such an environment most of the time!
- Get a domain expert!
- Get an interaction designer!
- Deliver swiftly and minimally while iterating with your users!



Final

Thoughts



Survey

How satisfied are you with our product?

Very Unsatisfied Neutral Satisfied
 Very Satisfied

How often do you use our product?

Daily Weekly

What is your favorite feature?

Any additional comments?

Submit

Do not intend to get everything right the first a few iterations!

Prioritize simplicity, control, and robustness over flashiness!

Carefully track cost and usage — especially important without major investors funding you.



Thank You