

Seminar CS715 / SM444

Solving Complex Tasks using Large Language Models



- **Prof. Dr. Christian Bizer**
- Professor for Information Systems V
- Research Interests:
 - Web-based Systems
 - Web Data Integration
 - Deployment of Data Web Technologies
- Room: B6 - B1.15
- eMail: christian.bizer@uni-mannheim.de



Hello

- **Dr. Ralph Peeters**
- Postdoctoral Researcher
- Research Interests
 - Sustainable LLM-Agents
 - Entity Matching using Deep Learning
 - Data Integration
- Office: B6, 26 - C 1.04
- eMail: ralph.peeters@uni-mannheim.de



Hello

- **M.Sc. Aaron Steiner**
- Graduate Research Associate
- Research Interests
 - Entity Matching using LLMs
 - LLM agents using RAG
 - Data Integration
- Office: B6, 26 - C 1.04
- eMail: aaron.steiner@uni-mannheim.de



You and Your Experience

- A Short Round of Introductions
 - What are you studying?
 - Which DWS courses did you attend?
 - What kind of experience do you have with
 - Large Language Models (LLMs) and
 - LLM-based agents or workflows?
- Participants
 - Chen, Yen-An
 - Gänz, Philipp
 - Guo, Can
 - Liu, Yu-Wei
 - Schallwig, Emil
 - Croissant, David
 - Lyu, Shuaiqi
 - Scherr, Jacqueline
 - Tagne, Robin

Agenda of Today's Kickoff Meeting

1. Seminar organization
2. Introduction to LLM-based Agents
3. Topic Assignment
4. How to structure your paper / presentation?
5. Your Questions

1. Seminar Organization

Learning Goals

- Writing a seminar thesis as an exercise for your master thesis
- Searching and citing scientific papers / journal articles
- Understand and present state-of-the-art scientific literature
- Design experiments and present experimental results
- How to structure your thesis and presentation?
- How to write a scientific paper using LaTeX?

Schedule

Date	Session
Thursday, 18.09.2025 (10:15-11:45)	Kick-off meeting and topic/mentor assignment
	Read papers about your topic Search additional literature Design experimental setup Select methods/design experiments, prepare presentation outline
Until 9.10.2025	Meet with your mentor to discuss outline and/or experimental setup
	Prepare draft of your presentation
Until 26.10.2025	Send draft presentation to your mentor
	Finalize your presentation
Monday, 10.11.2025 (10:00-12:00) (14:00-16:00)	Presentation and discussion of your topic (30 % of your final grade)
	Write seminar thesis
Friday, 23.01.2026	Submission of your seminar thesis (70 % of your final grade)

Formal Requirements

- Presentation
 - 12 minutes + 8 minutes discussion
 - should be 100% understandable for all participants
- Written report (paper)
 - 12-15 pages single column
 - including abstract and appendixes
 - not including the bibliography
 - not including the page about LLM usage
 - every additional page reduces your grade by 0.3
 - written in English
 - use seminar report template from DWS templates
- Final grade
 - 70% written report
 - 30% presentation

Which template to use?

DWS Seminar Report Template

Seminar Report

Title of Your Report

Max Muster

July 1, 2024

Your report must contain an abstract. A good reference for report writing is Zobel (2014); we highly recommend that you study this or a similar book during your studies. He writes the following about the abstract:

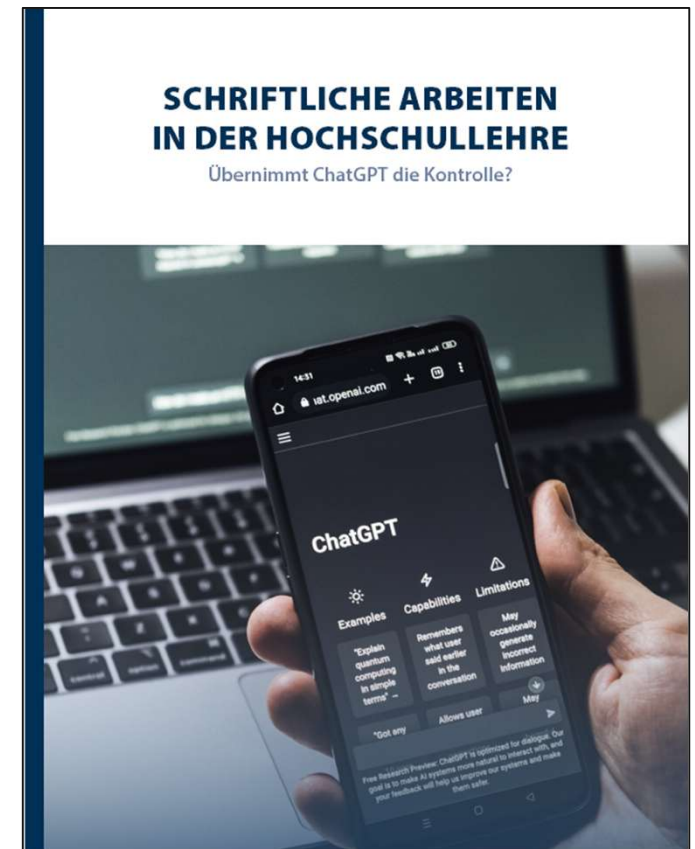
An abstract is typically a single paragraph of about 50 to 200 words. The function of an abstract is to allow readers to judge whether or not the paper is of relevance to them. It should therefore be a concise summary of the paper's aims, scope, and conclusions. There is no space for unnecessary text; an abstract should be kept to as few words as possible while remaining clear and informative. Irrelevancies, such as minor details or a description of the structure of the paper, are inappropriate, as are acronyms, abbreviations, and mathematics. Sentences such as "We review relevant literature" should be omitted.

https://www.uni-mannheim.de/media/Einrichtungen/dws/Files_Teaching/Theses/dws-templates.zip

Statement About the Tools that You Used

Your report **must include an extra page** about

1. which generative AI tools you used
 - ChatGPT, OpenAI (API) (Researcher), Claude (Code), Gemini (cli), DeepL
2. for which purposes
 - structuring your paper
 - summarizing related work
 - writing text for specific chapters
 - improving English grammar and formulations
 - designing experimental setup
 - writing code
 - writing prompts
 - generating training data
 - summarize log files
 - perform error analysis
 -
3. How useful was each tool for this?



<https://www.uni-mannheim.de/infos-fuer/forschende-und-lehrende/lehren/ihre-lehre-im-fokus/>

Example of AI Tools Declaration (Part of DWS Template)

Ehrenwörtliche Erklärung

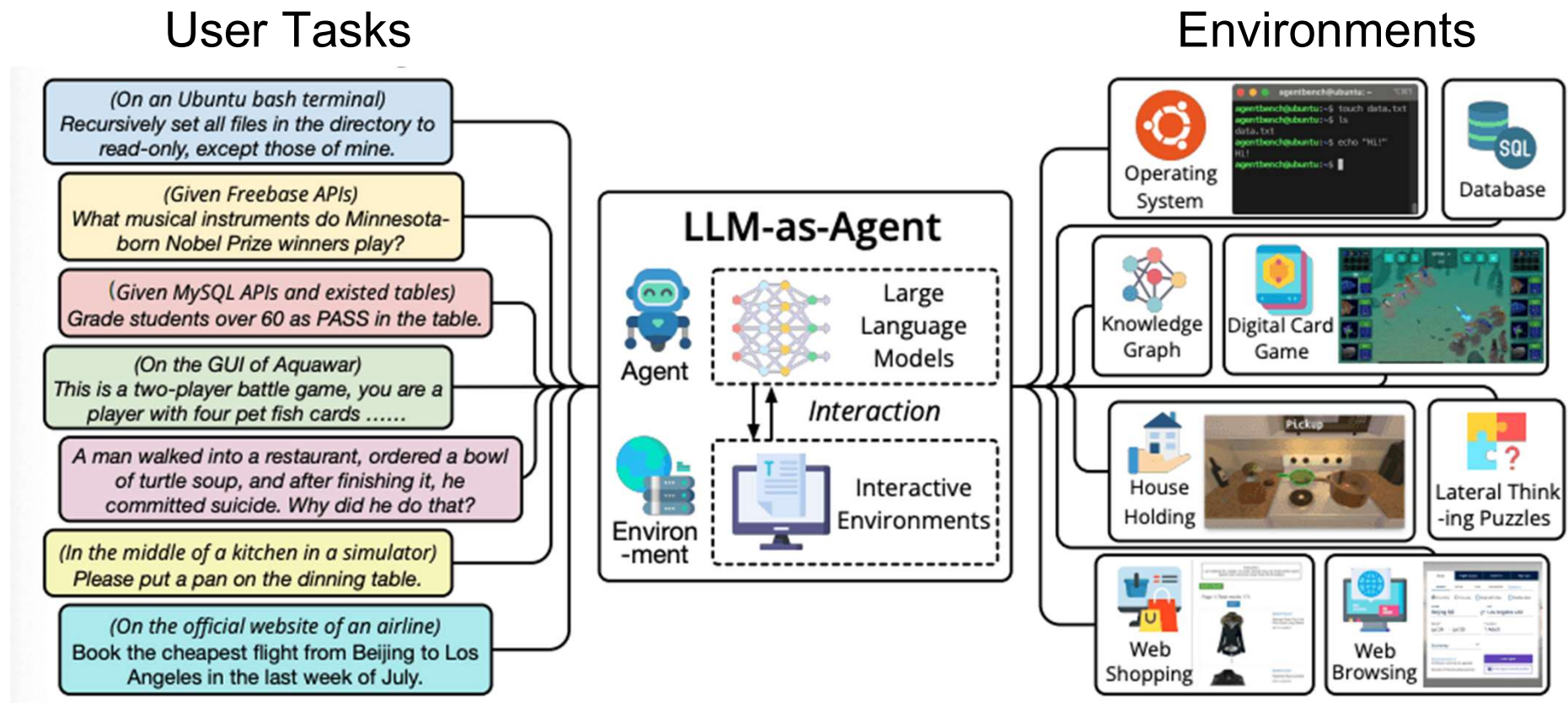
Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepL	Translation	Throughout	+
ResearchGPT	Summarization of related work	Sec. 2	-
Dall-E	Image generation	Figs. 2, 3	++
GPT-4	Code generation	functions.py	+
ChatGPT	Related work hallucination	Most of bibliography	++

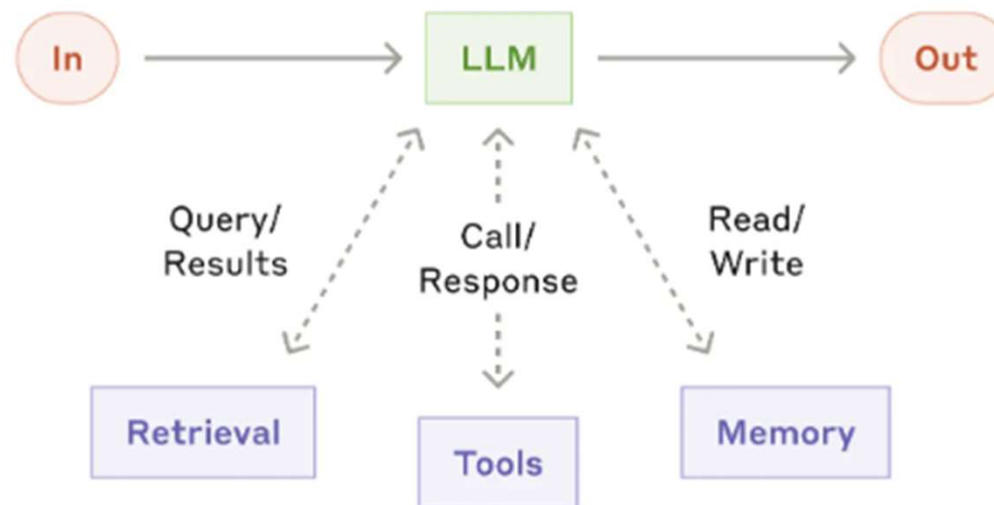
2. Introduction to LLM-based Agents

LLM-based agents autonomously interact with an environment to solve user tasks.



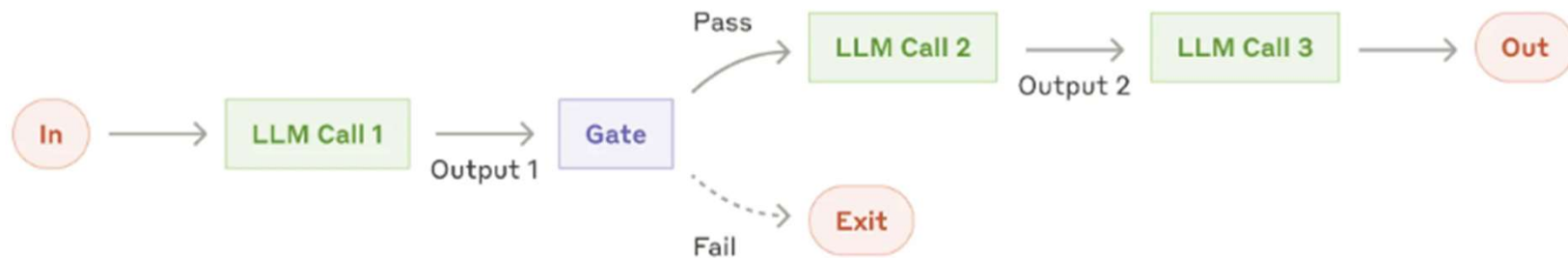
Tool Usage

- LLM agents use **tools**
 - search the Web or document repositories
 - query databases and ERP systems
 - running code in execution environment
 - move the mouse or type using the keyboards
 - memorizing results of previous interactions



LLM Workflows

- LLM-based applications increasingly **perform workflows**
- prompt chaining workflow:



- example where prompt chaining is useful:
 - generating marketing text, then translating it into a different language
- the workflows are hard-coded
 - using frameworks like LangGraph

<https://www.anthropic.com/research/building-effective-agents>

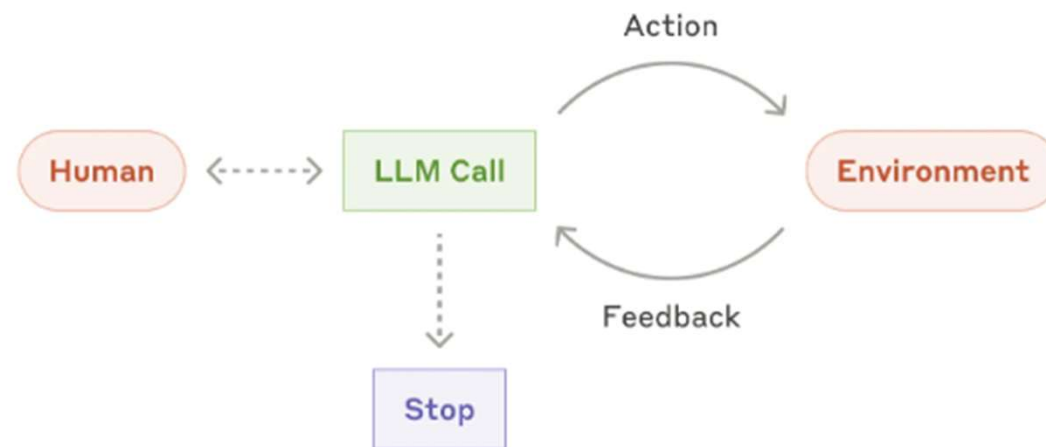
Workflow: Orchestrator-Workers



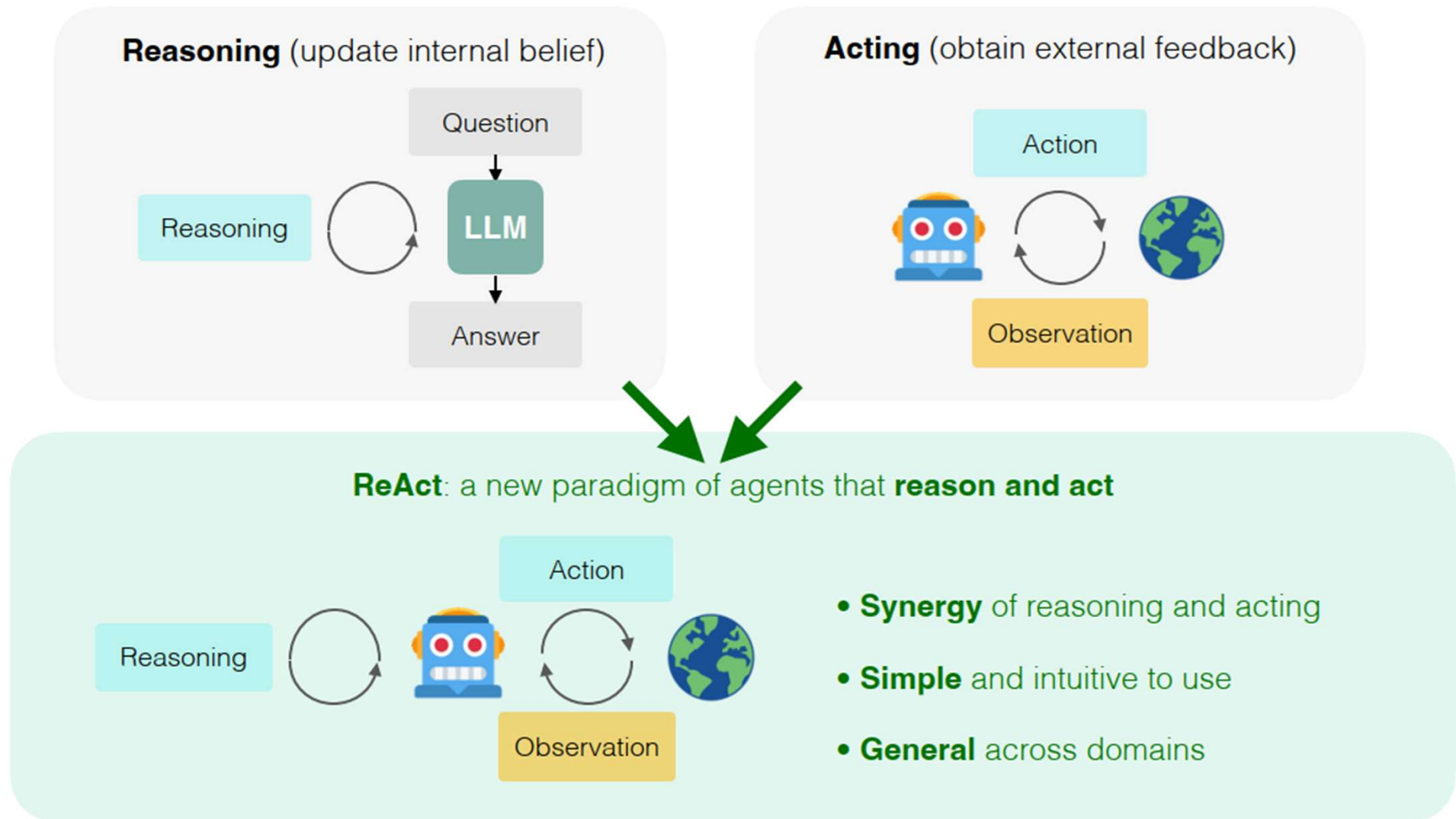
- examples where orchestrator-workers is useful:
 - search tasks that involve gathering and analyzing information from multiple sources for possible relevant information, e.g. GPT Researcher
 - coding tasks that make complex changes to multiple files
- still hard-coded workflow

LLM-based Agents

- given a task, agents plan and operate autonomously
 - no fixed workflow, but flexible series of actions planned by agent
- agent consists of a **loop** iterating between LLM and tool calls (actions)
- in each iteration
 1. the agent observes the environment
 2. reasons about the task given the environment and its previous actions
 3. choose an action to perform
 4. the agent's runtime environment executes the action



The ReAct Paradigm



Yao, S., et al., 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

Example ReAct Prompt

System
prompt
defines
agent

You are an agent that answers questions by using two actions:

- search[query]: Google search the query. You can also use it to calculate math.
- finish[answer]: return the answer

Your generation should have the following format:

Thought: to analyze the observation and inform the action.

Action: your action

User
task

Question: If I have seven trillion dollars today, can I buy Apple, Nvidia, and Microsoft? If not, how much more money do I need?

ReAct Trajectory

LLM

(GPT-4)

Thought: I need to find the current market capitalization of Apple, Nvidia, and Microsoft to understand how much it would cost to buy all of them.

Action: search["current market capitalization of Apple, Nvidia, and Microsoft"]

Observation: "Nvidia's market cap stood at \$1.832 trillion as of about 10 a.m. ET Wednesday, while Alphabet's was at \$1.814 trillion, bringing Nvidia behind just Microsoft (MSFT) at \$3.015 trillion and Apple (AAPL) at \$2.836 trillion."



(Mar 2024)

Thought: Now that I have the market cap for all companies, let's add them together and see if \$7 trillion is enough.

Action: search["1.832 trillion + 3.015 trillion + 2.836 trillion"]

Observation: 7683000000000

Thought: The amount of money needed in addition to the \$7 trillion is \$683 billion.

Action: finish["You need an additional \$683 billion to be able to buy Apple, Nvidia, and Microsoft."] ✓

Observation Space and Action Space

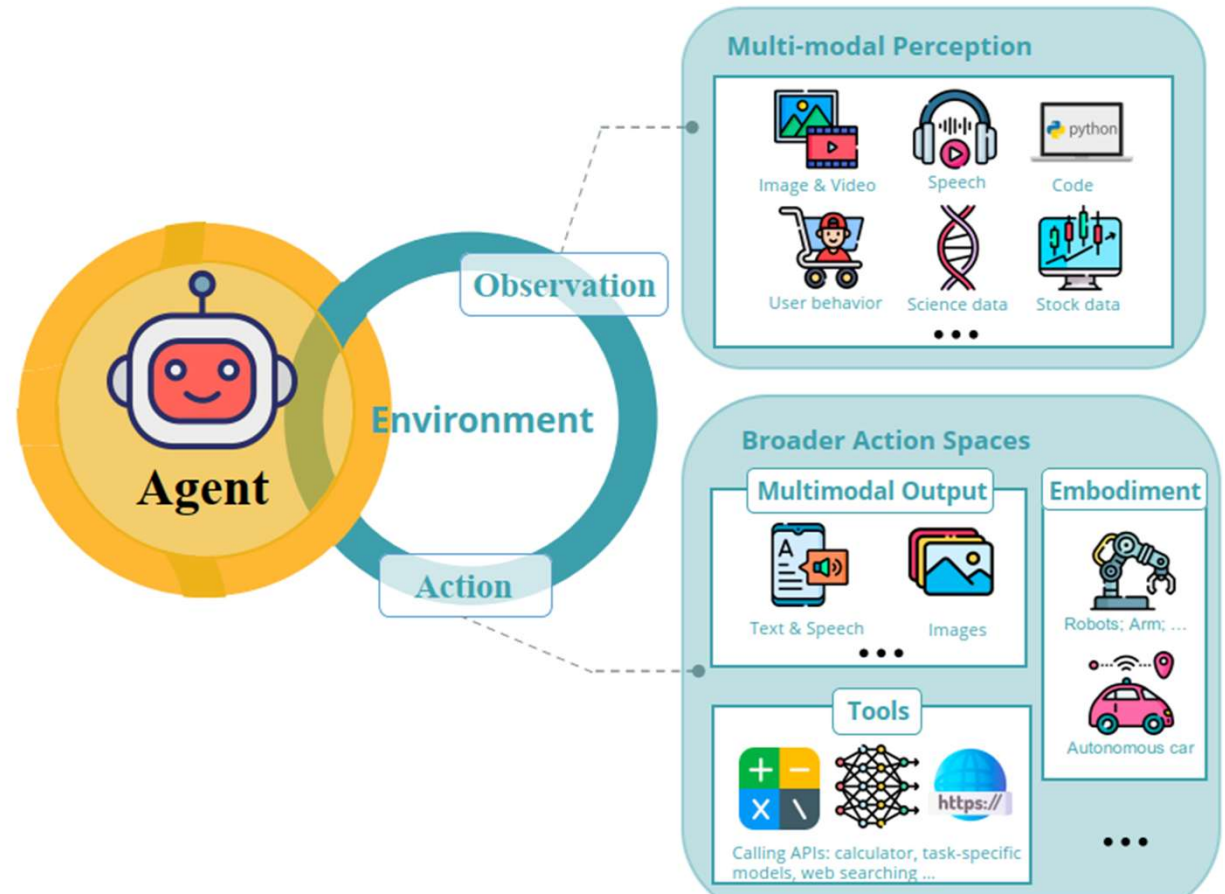


Action

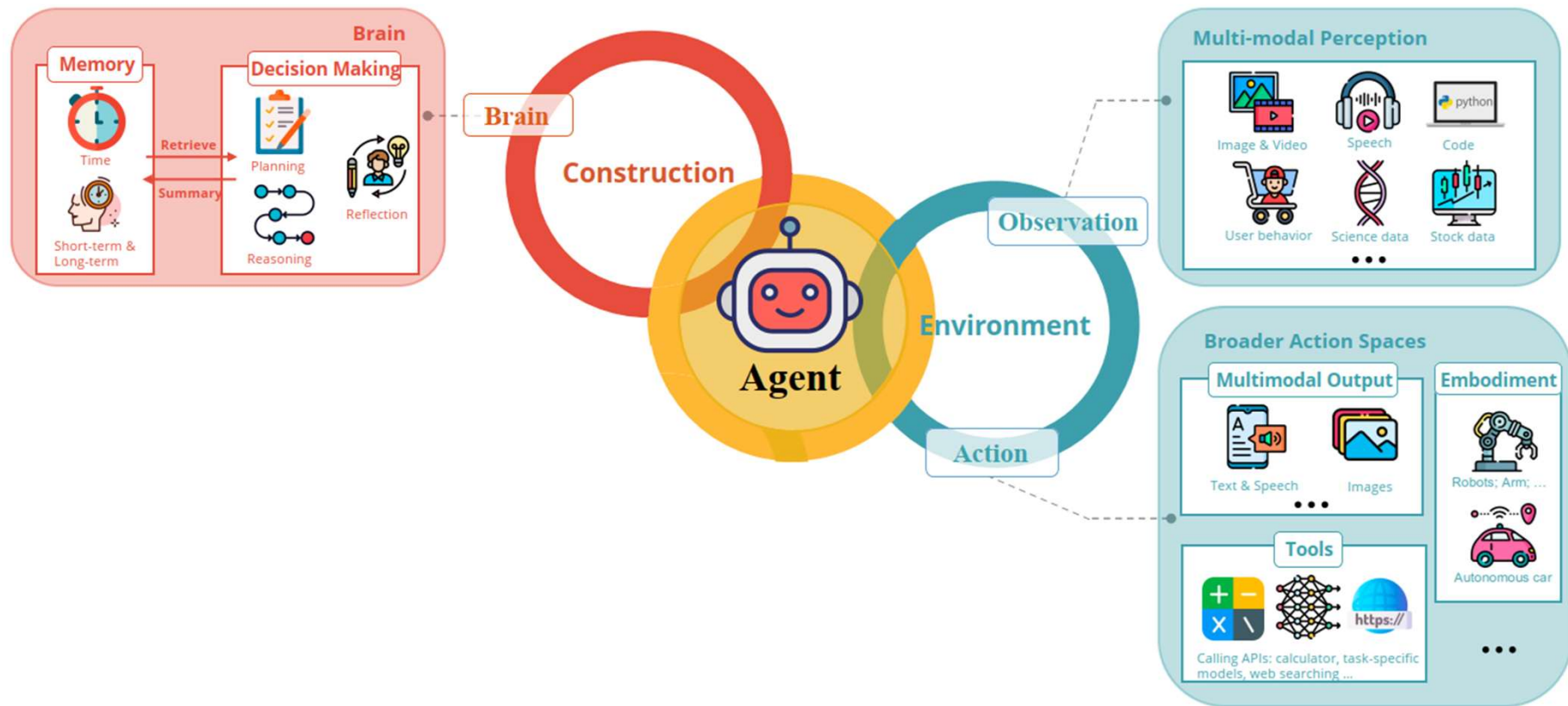
- call external **APIs** for extra information that is missing from the model weights (often hard to change after pre-training):
Generating multimodal outputs;
Embodied Action; Learning tools;
Using tools; Making tools;



Figure 2: **VOYAGER** consists of three key components: an automatic curriculum for open-ended exploration, a skill library for increasingly complex behaviors, and an iterative prompting mechanism that uses code as action space.

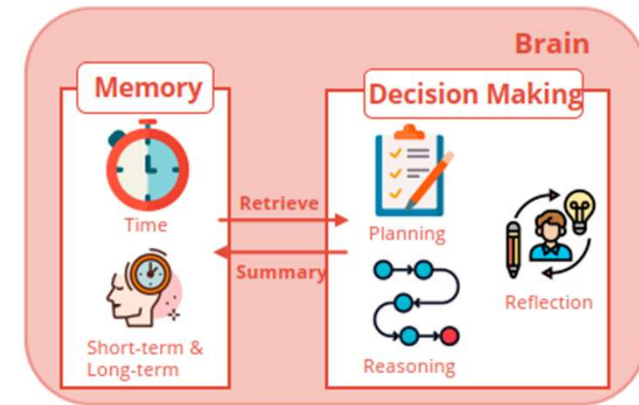


The “Brain”: Decision Making and Memory



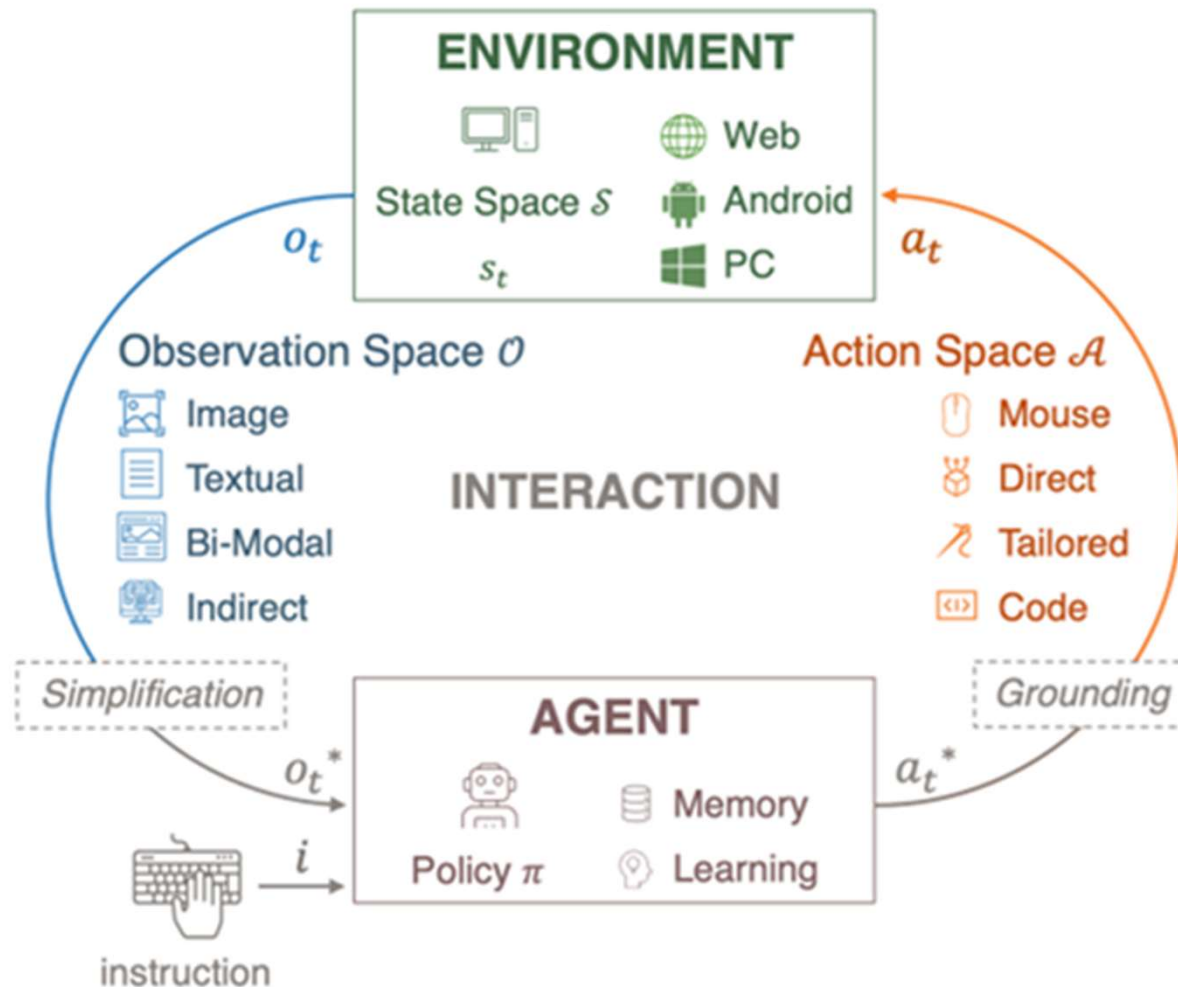
The “Brain”

- **Memory:** stores sequences of agent’s past observations, thoughts and actions
 - simple case: agent’s trajectory
 - past observations and actions
 - more sophisticated memory
 - long-term and short-term memory
 - long-term memory is abstract



- **Decision Making Process:**
 - **Planning:** Decomposition into subgoals – Break down large tasks into smaller, manageable subgoals, enabling efficient handling of complex tasks
 - **Reasoning:** Self-criticism and self-reflection over past actions, learn from mistakes and refine for future steps

Focus of the Seminar: Web Agents



Sager et al.: AI Agents for Computer Use: A Review of Instruction-based Computer Control, GUI Automation, and Operator Assistants. arXiv:2501.16150, 2025

Example: Prompt of Web Browsing Agents (abbreviated)

SYSTEM: You are an agent trying to solve a web task based on the content of the page and user instructions. You can interact with the page and send messages to the user. Each time you submit an action it will be sent to the browser and you will receive a new page.

SYSTEM:

Goal: Find the cheapest offer for an Iphone 13.

Observation of current step:

Currently open tabs:

Tab 0 (active tab):

Title: WebMall

URL: http://localhost:8085/

AXTree: ...

Action space:

15 different types of actions are available.

noop(wait_ms: float = 1000)

scroll(delta_x: float, delta_y: float)

keyboard_press(key: str)

click(bid: str, button: Literal['left', 'middle', 'right'] = 'left')

fill(bid: str, value: str)

hover(bid: str)

tab_focus(index: int)

new_tab() ...

Concrete Example

Here is a concrete example of how to format your answer.

<think>

From previous action I tried to set the value of year to "2022", using select_option, but it doesn't appear to be in the form. It may be a dynamic dropdown, I will try using click with the bid "a324" and look at the response from the page.

</think>

<action> click('a324') </action>

Agent:

<think>

To find the cheapest offer for an Iphone 13, I need to explore different webshops. I will start by visiting Shop 1.

</think>

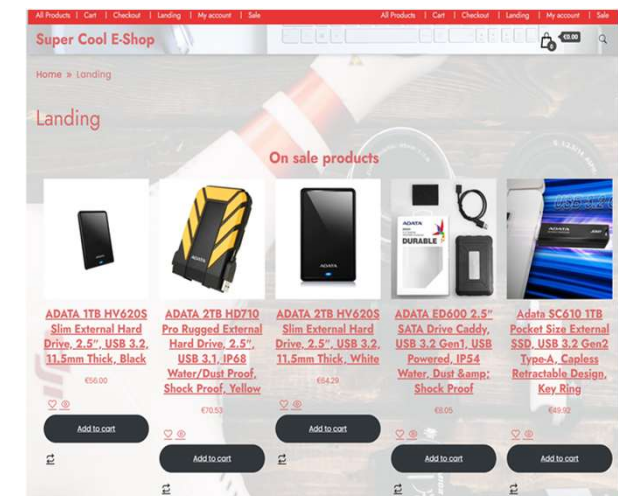
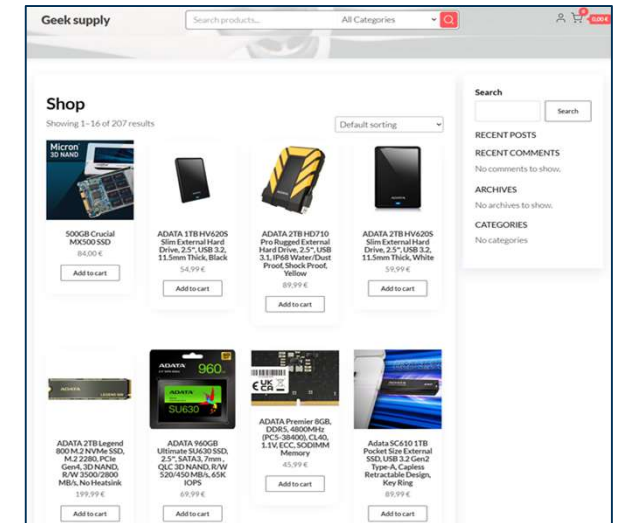
<action>

click('9')

</action>

WebMall - A Multi-Shop Benchmark for Evaluating Web Agents

- benchmark for evaluating the ability of agents to find and compare offers from multiple shops
- simulates an online shopping environment consisting of four heterogeneous online shops
- defines basic and advanced tasks
 - basic tasks: comparing offers with concrete requirements, adding offers to the shopping cart
 - advanced tasks: searches with vague requirements, searches for compatible and substitute products
- the shops offers different interfaces
 - HTML pages, MCP APIs, NLWeb interface



<https://wbsg-uni-mannheim.github.io/WebMall/>
<https://webmall-1.informatik.uni-mannheim.de/> (can be accessed with VPN turned on)

2. Seminar Topics and Topic Assignment

- The seminar features literature (1) as well as experimental topics (8).
- The goal of the **experimental topics** is to verify methods from literature by applying them to tasks beyond the tasks used in the respective papers.
- The goal of the **literature topics** is to describe and compare the state of the art methods/approaches concerning the respective topic.

1. Advanced Web Agents for Online Shopping

- Experimental topic
- Student: Robin Tagne
- Mentor: Ralph Peeters

Some papers as starting point

- Ning et al., A Survey of WebAgents: Towards Next-Generation AI Agents for Web Automation with Large Foundation Models, in Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2025, pp. 6140–6150.
- He et al., WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models, in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 2024, pp. 6864–6890.
- Peeters et al., WebMall – A Multi-Shop Benchmark for Evaluating Web Agents, arXiv:2508.13024, 2025.

2. Token-Efficient Web Agents for Online Shopping

- Experimental topic
- Student: Philipp Gänz
- Mentor: Ralph Peeters

Some papers as starting point

- Chezelles et al., The BrowserGym Ecosystem for Web Agent Research, Transactions on Machine Learning Research, 2024
- Zhou et al., A Survey on Efficient Inference for Large Language Models, arXiv:2404.14294, 2024.
- Peeters et al., WebMall – A Multi-Shop Benchmark for Evaluating Web Agents, arXiv:2508.13024, 2025.

3. Memory Design for Agents on Long-running Tasks

- Experimental topic
- Student: Shuaiqi Lyu
- Mentor: Ralph Peeters

Some papers as starting point

- Zhang et al., A Survey on the Memory Mechanism of Large Language Model based Agents, ACM Trans. Inf. Syst., 2025.
- Maharana et al., Evaluating Very Long-Term Conversational Memory of LLM Agents, in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 2024, pp. 13851–13870.
- Wang et al., Augmenting Language Models with Long-Term Memory, Advances in Neural Information Processing Systems, vol. 36, pp. 74530–74543, 2023.

4. Robustness of Web Agents under Adversarial Conditions

- Experimental topic
- Student: David Croissant
- Mentor: Ralph Peeters

Some papers as starting point

- Yang et al., GUI-Robust: A Comprehensive Dataset for Testing GUI Agent Robustness in Real-World Anomalies. [arXiv:2506.14477](#), 2025.
- Nitu et al., Machine-Readable Ads: Accessibility and Trust Patterns for AI Web Agents interacting with Online Advertisements. [arXiv:2507.12844](#), 2025.
- Abuelsaad et al., Agent-E: From Autonomous Web Navigation to Foundational Design Principles in Agentic Systems. [arXiv:2407.13032](#), 2024.

5. Evaluating Agent-to-Agent Negotiation Interfaces

- Experimental topic
- Student: Yen-An Chen
- Mentor: Aaron Steiner

Some papers as starting point

- Derouiche, H., Brahmi, Z., & Mazeni, H. (2025): Agentic AI frameworks: Architectures, protocols, and design challenges. arXiv preprint arXiv:2508.10146
- <https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/>
- <https://github.com/a2aproject/A2A>

6. RAG over Heterogeneous Corporate Data

- Experimental topic
- Student: Can Guo
- Mentor: Aaron Steiner

Some papers as starting point

- Yu, X., Jian, P., & Chen, C. (2025): TableRAG: A retrieval augmented generation framework for heterogeneous document reasoning. arXiv preprint arXiv:2506.10380
- Choi, N., Byun, G., Chung, A., Paek, E. S., Lee, S., & Choi, J. D. (2025): Reference-aligned retrieval-augmented question answering over heterogeneous proprietary documents. arXiv preprint arXiv:2502.19596

7. Optimizing RAG Pipelines with Preprocessing and Structured Representations for the WebMail Use Case

- Experimental topic
- Student: Yu-Wei Liu
- Mentor: Aaron Steiner

Some papers as starting point

- Lyu et al.: DeepShop: A Benchmark for Deep Research Shopping Agents, arXiv:2506.02839, 2025.
- Lewis et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in Advances in Neural Information Processing Systems, 2020, pp. 9459–9474.

8. Fact Verification using RAG

- Literature topic
- Student: Jacqueline Scherr
- Mentor: Christian Bizer

Some papers as starting point

- Dmonte et al.: Claim Verification in the Age of Large Language Models: A Survey. arXiv:2408.14317, 2024.
- Ge et al.: Resolving Conflicting Evidence in Automated Fact-Checking: A Study on Retrieval-Augmented LLMs. arXiv:2505.17762, 2025.

9. Variant Detection in Product Data

- Experimental topic
- Student: Emil Schallwig
- Mentor: Christian Bizer

Some papers as starting point

- Vidal et al.: Learning Variant Product Relationship and Variation Attributes from E-Commerce Website Structures. In Generative AI for E-Commerce at CIKM, 2024.
- West et al.: Interpretable Methods for Identifying Product Variants, in Companion Proceedings of the Web Conference 2020, in WWW '20. 2020, pp. 448–453.

3. How to Structure Your Paper / Presentation

Goals of Literature and Experimental Papers

– Goals of Literature Papers

1. describe the **problem / task** and give overview of **state-of-the-art**
2. describe selected **existing methods/systems** in detail,
3. compare the methods/systems and their **evaluation** using a **systematic set of comparison criteria**

– Goals of Experimental Papers

1. **describe state of the art** concerning your problem area
2. summarize the **evaluation tasks and results** from the papers
3. design **experimental setup** to evaluate methods(s) on a different task
4. Perform an **error analysis** to understand weaknesses of methods
5. compare **your results** to the **results from the paper**

How to Structure Your Literature Paper?

1. Introduction and Problem Statement
 - Which problem/task is addressed? Why is the problem important?
 - Structure of your paper
2. Description of Existing Approaches
 - Overview of **state of the art** concerning the problem/task
 - Detailed description of **selected methods** (likely two)
 - Comparison of the selected methods using a **set of comparison criteria**
3. Evaluation
 - Comparison and **critical discussion of the evaluation tasks**, metrics
 - Comparison of the evaluation results using a **set of comparison criteria**
4. Conclusion
 - What did the comparison of the methods and evaluation results show?
 - Can something be concluded for future work?
5. Bibliography (**20 - 30 references**)

How to Structure Your Experimental Paper?

1. Introduction and Problem Statement

- Which problem is addressed? What is the **overall approach** for addressing it?
- Overview of the existing methods/papers and their evaluation (3 pages+)
- Structure of your paper

2. Description of Your Experimental Design

- How to you select **examples** for which **challenges**?
- Which **method/language model combinations** do you test?

3. Presentation of Experimental Results

- Present the **results** of your experiments (tables containing values and deltas).
- Present the results of your **error analysis** (types of errors, frequency of these types)

4. Conclusion

- What did the experiments and the error analysis show?
- How to your results compare to the experiments presented in the papers?

5. Bibliography (**10 - 20 references**)

Learn from Examples

- Read **survey articles and previous experimental papers** and identify the structure from the previous slides
 - Why can this paragraph be found at that position?
 - What is the purpose of some section / subsection?
- Some relevant surveys
 1. Wang, et al: **A Survey on Large Language Model based Autonomous Agents**. arXiv:2308.11432, 2023.
 2. Sager et al.: **AI Agents for Computer Use**. arXiv:2501.16150, 2025.
 3. Mialon, et al.: **Augmented Language Models: a Survey**. arXiv:2302.0784
 4. Zhao, et al.: **A survey of Large Language Models**. arXiv:2303.18223
- Textbook on how to write a thesis
 - Zobel: Writing for Computer Science, 3rd Edition, Springer 2014.

Citing Different Types of Publications

1. Journal article, conference and workshop paper
 - Good to cite (cite at least **10 papers** about specific methods)
2. Survey articles
 - Good to cite as overviews for specific topics, but prefer individual papers as reference for specific systems (cite at least **2-3 surveys** in your introduction)
3. Books (sometimes cited)
 - Textbooks
 - Collections of articles/papers => Cite specific paper in book
4. Websites
 - better not cited, exceptions are, e.g., documents like W3C Specifications
 - Do not cite Wikipedia, ever!
 - **Use footnotes** to refer to project pages, download pages, or technical documentation
5. Slide sets (especially from our lectures)
 - **Never cite!**

How to Find Relevant Publications?

1. Start with gathering relevant papers from the **surveys**
 1. Wang, et al: A Survey on Large Language Model based Autonomous Agents. arXiv:2308.11432. 2023.
 2. Sager et al.: AI Agents for Computer Use. arXiv:2501.16150, 2025.
2. **Exploit references:** Given a relevant document x
 - Follow references in the past: papers y that x has cited
 - Follow references in the future: papers y that cited x („**cited by**” functionality in Google scholar)
3. **Use Google Scholar or Semantic Scholar**
 - we use it a lot ourselves

Using LLMs to Write Scientific Texts

- **you are responsible** for the scientific quality of your text!
- LLM tend to make too bold statements and draw conclusions without proper evidence, which you need to correct:
 - “comprehensive study”, “rigorous evaluation”, “move the state-of-the-art”
 - “the experiments showed ...” Did they really show this?
- When you ask LLMs to discuss related work, they tend to write shallow texts focusing on arbitrary aspects. Thus:
 1. collaborate with the LLM to determine relevant comparison criteria
 2. ask the LLM to discuss related work along these criteria
 3. **verify** whether the discussion makes sense and if it reflects the actual content of the papers
 4. **improve** the discussion, refine the criteria, including additional relevant papers that LLMs has missed

Using the LLM to Help You with the Error Analysis

- Error Analysis Process
 1. Sample a subset of the errors (30 to 200 errors)
 2. Determine a set of error classes by analyzing the errors (5-10 classes)
 3. Determine the frequency of each error class
 4. Present the results as a table with the columns Error class, Frequency

- You can collaborate with an LLM in the process
 - Error Classes
 - Determine error classes yourself
 - Have the LLM propose error classes given the errors
 - Combine your ideas and the LLM-generated classes
 - Frequency of the different types of errors
 - You need to annotate a relevant subset of examples yourself
 - Afterwards, you can ask the LLM to the whole set
 - You need to report the **correlation** between your categorizations and the categorizations generated by the LLM

4. Questions?

