UNIVERSITY OF MANNHEIM

Seminar CS715 / SM444

Solving Complex Tasks using Large Language Models

Hallo

- **Prof. Dr. Christian Bizer**
- Professor for Information Systems V
- Research Interests:
  - Web-based Systems
  - Web Data Integration
  - Large Language Models and Agents
- Room: B6 - B1.15
- eMail: christian.bizer@uni-mannheim.de

Hello

- **Dr. Ralph Peeters**
- Gastprofessor
- 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
  - Banser, Pascal
  - Lee, Danny Daewon
  - Michel, Oskar
  - Monasara, Abhaykumar
  - Morali, Noyan
  - Pandya, Marmee
  - Reifferscheid, Luis
  - Sardar, Krittika
  - Wiedenhofer, Maurits
  - Xu, Shiqi

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

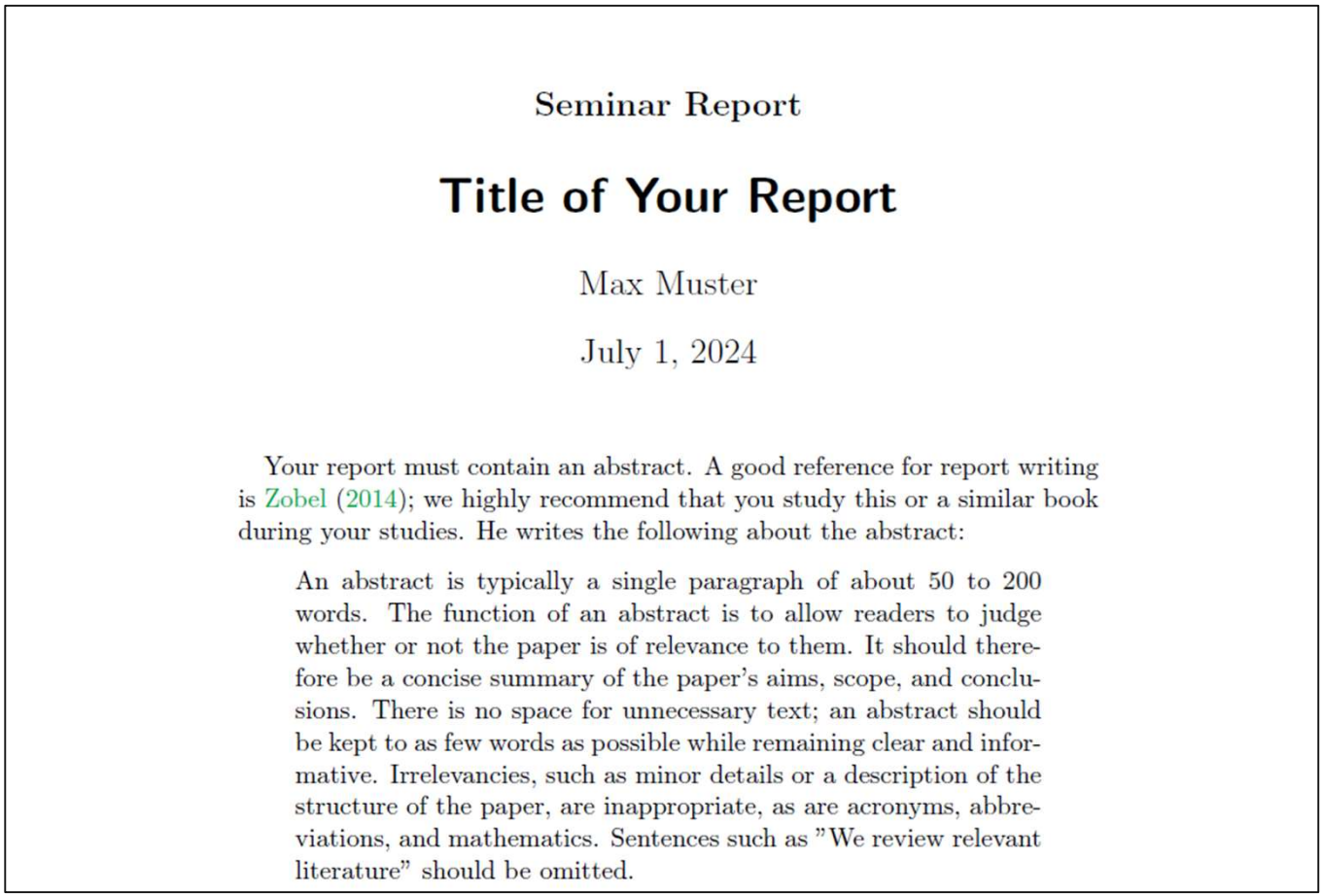
Date	Session
Wednesday, 25.02.2026 (15:00-16:30)	Kick-off meeting and topic/mentor assignment
	Read papers about your topic Search additional literature Design experimental setup Select methods/design experiments, prepare presentation outline
Until 20.03.2026	Meet with your mentor to discuss outline and/or experimental setup
	Prepare draft of your presentation
Until 10.04.2026	Send draft presentation to your mentor
	Finalize your presentation
Monday, 04.05.2026 (10:00-12:00) (14:00-16:00)	Presentation and discussion of your topic (30 % of your final grade)
	Write seminar thesis
Friday, 03.07.2026	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

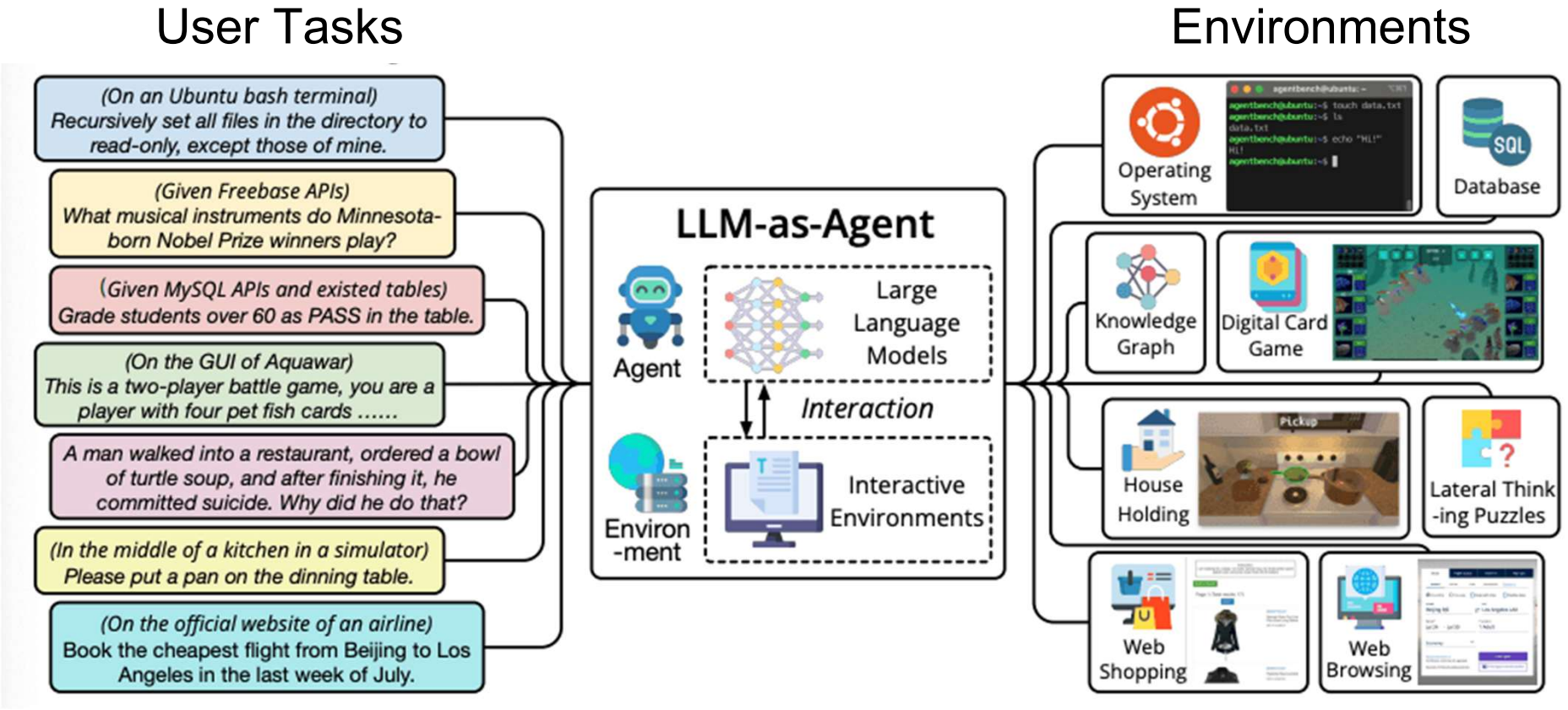
Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen 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	++

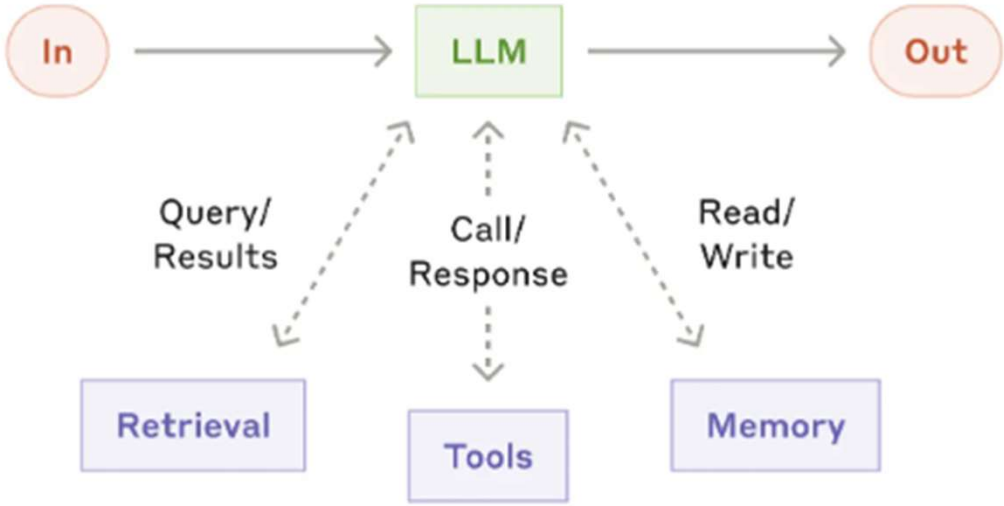
2. Introduction to LLM-based Agents

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



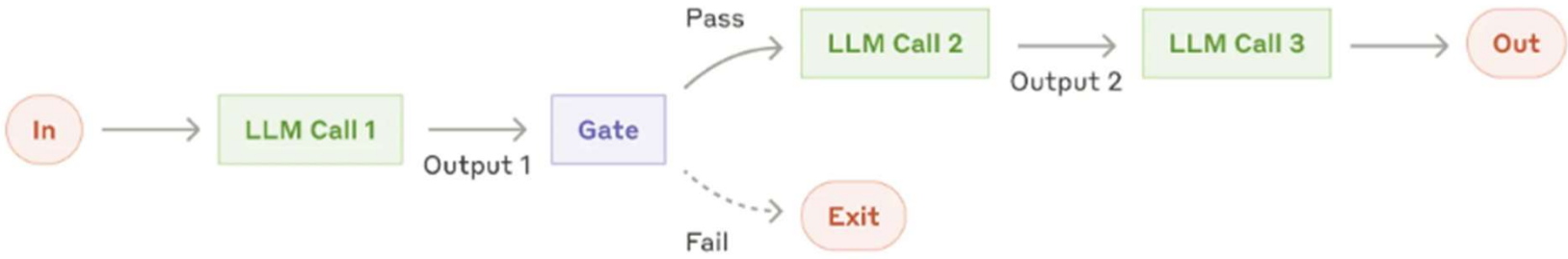
Tool Usage

- LLM agents use **tools**
  - search the Web or document repositories (RAG agents)
  - 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:

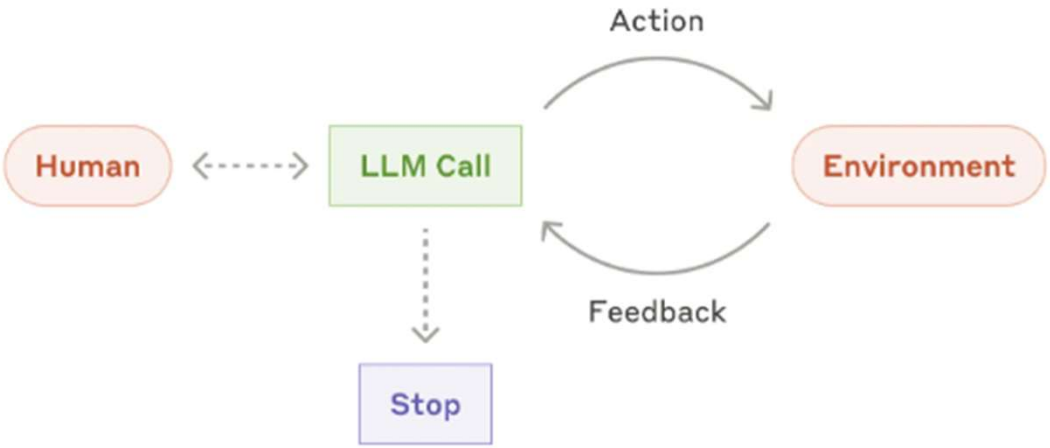


- example where prompt chaining is useful:
  - generating marketing text, then translating it into a different language
  - search tasks that involve gathering and analyzing information from multiple sources for possible relevant information
- the workflows are hard-coded
  - using frameworks like LangGraph

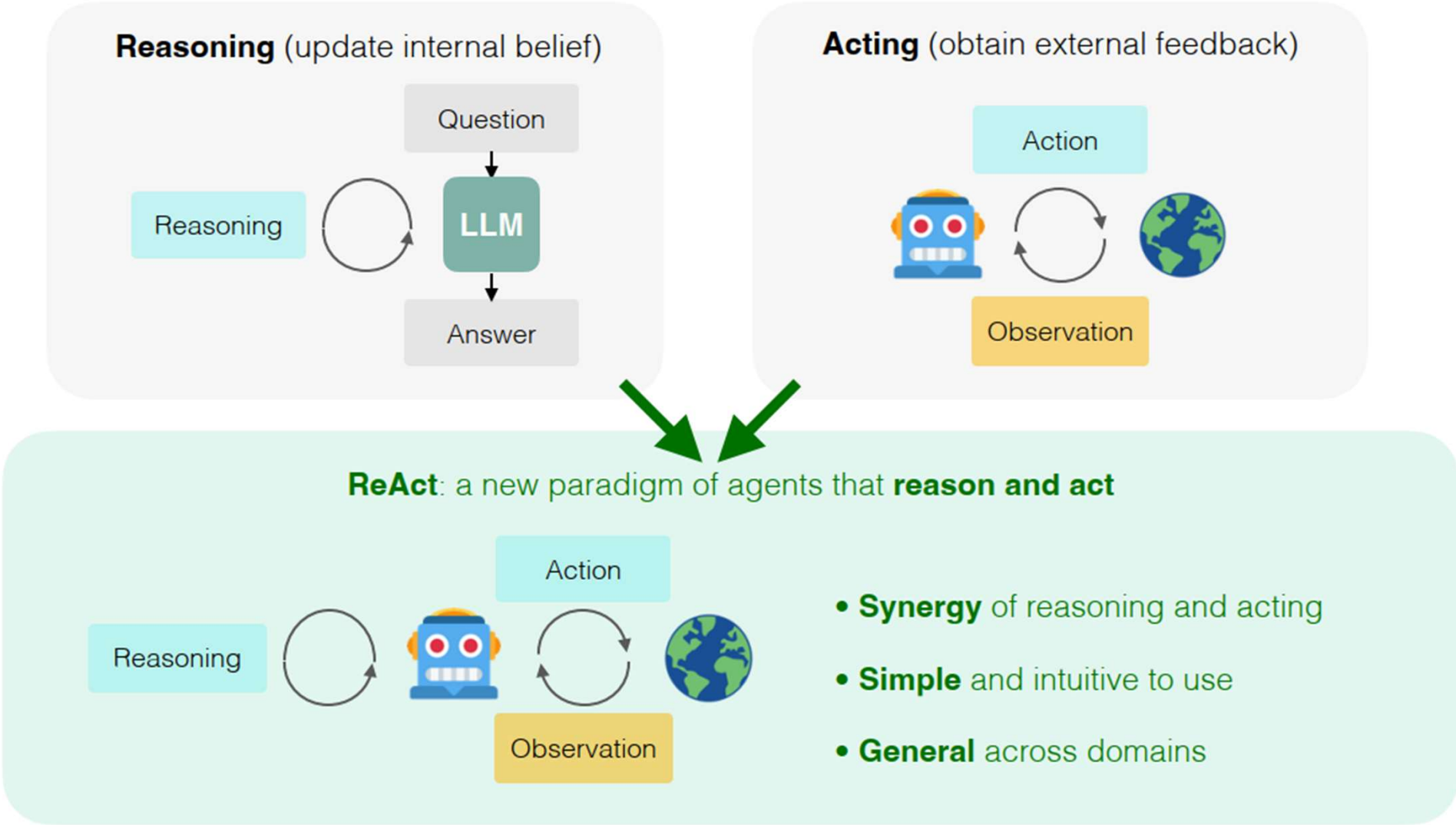
https://www.anthropic.com/research/building-effective-agents

LLM 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"]

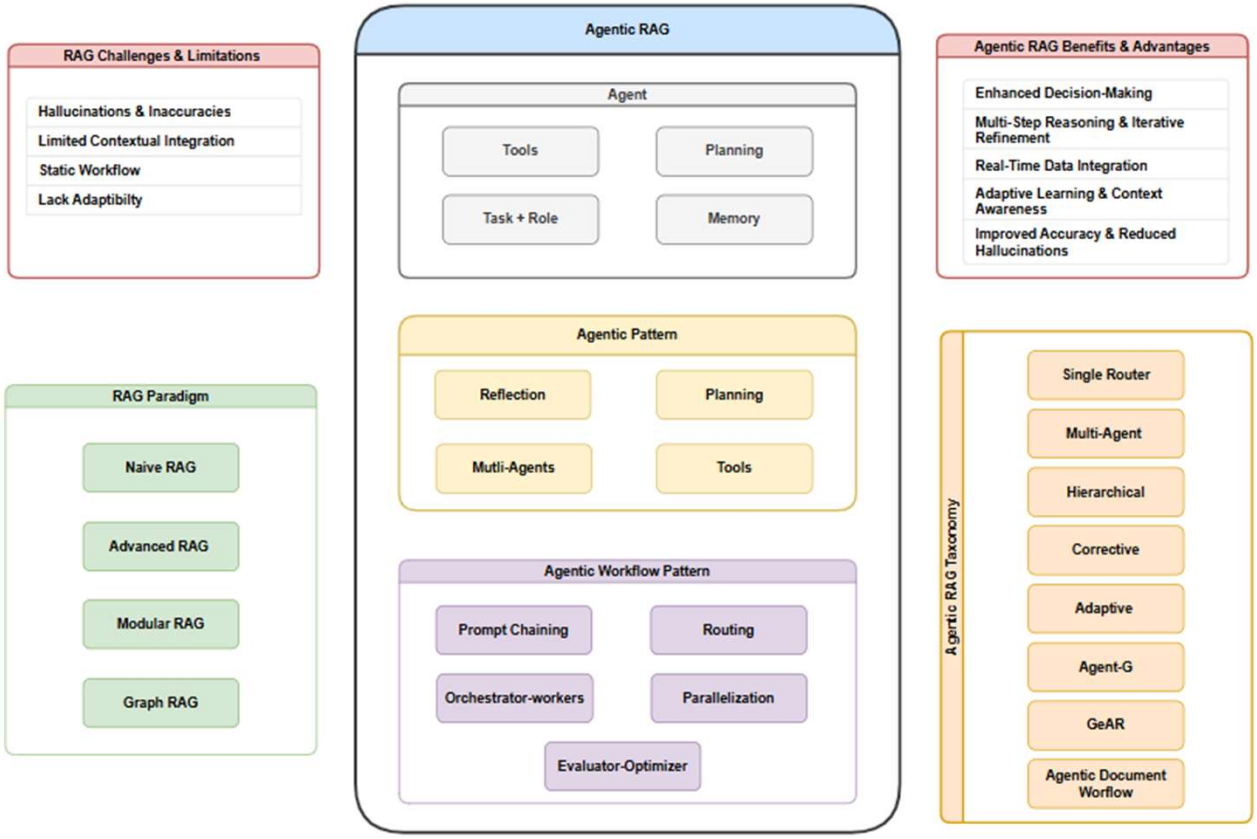
Observation: 7683000000000

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."]

Focus of the Seminar: RAG and Data Agents

RAG Agents

RAG agents use complex workflows to retrieve information and generate output using the retrieved information.

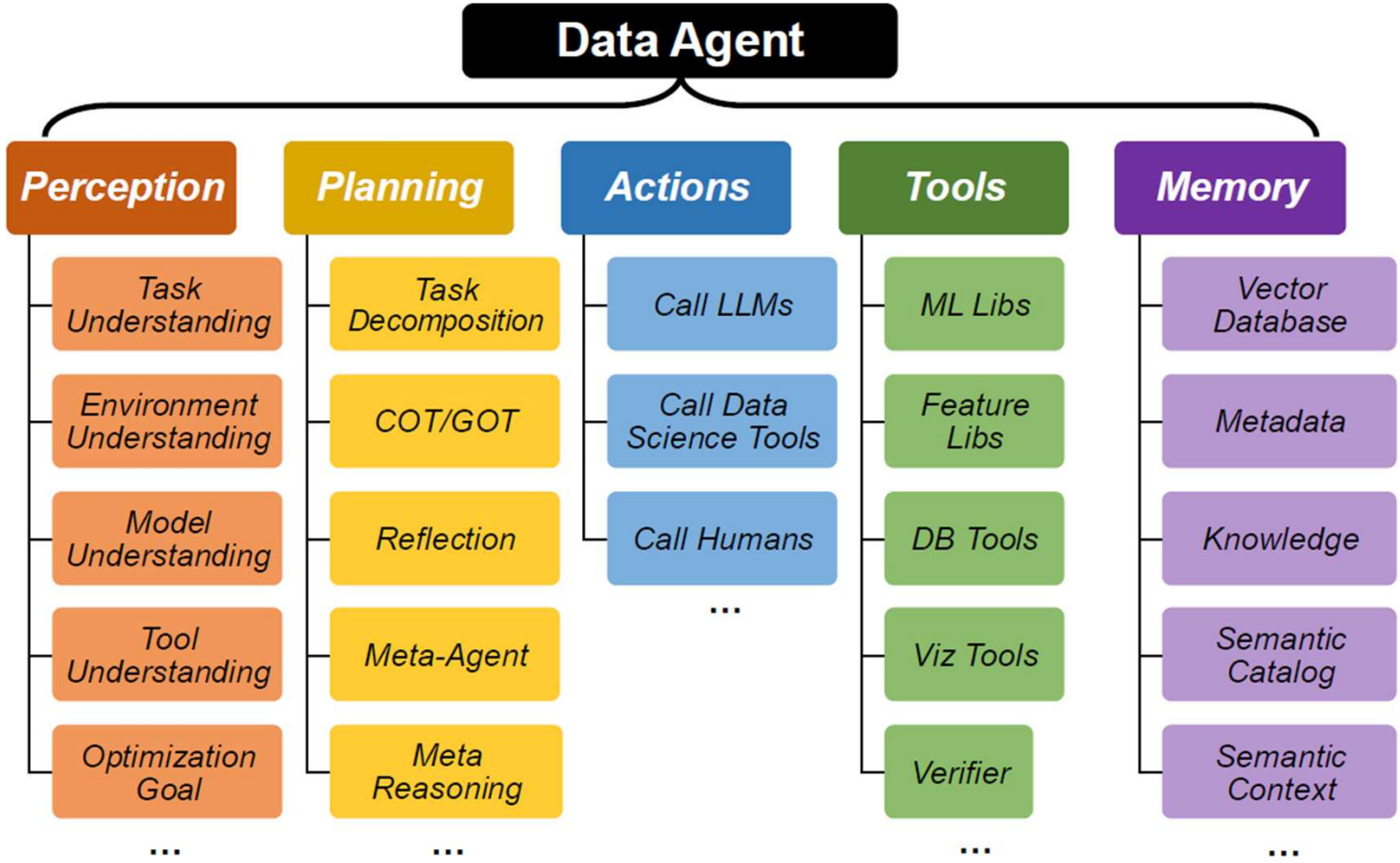


Overview of Agentic RAG

Singht et al.: Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. https://arxiv.org/abs/2501.09136, 2025. https://github.com/asinghcsu/AgenticRAG-Survey

Data Agents

Agents that collect, integrate, and prepare data for analytical tasks.



Zhu et al.: A Survey of Data Agents: Emerging Paradigm or Overstated Hype? arXiv preprint arXiv:2510.23587, 2025.
https://github.com/HKUSTDial/awesome-data-agents

2. Seminar Topics and Topic Assignment

- The seminar features literature (1) as well as experimental topics (9).

- 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. Wikipedia Article Generation Using Web-RAG

- Experimental topic
- Student: Maurits Wiedenhofer
- Mentor: Christian Bizer

Some papers as starting point

- Zhang, J. et al., 2025. WIKIGENBENCH: Exploring full-length Wikipedia generation under real-world scenario. In Proceedings of the 31st International Conference on Computational Linguistics, pp. 5191-5210.
- Yang, Z., et al., 2025. WikiAutoGen: Towards Multi-Modal Wikipedia-Style Article Generation. arXiv preprint arXiv:2503.19065
- Reeves, N. and Simperl, E., 2025. Machines in the Margins: A Systematic Review of Automated Content Generation for Wikipedia.

2. Verifying Scientific Claims using Web-RAG Agents

- Experimental topic
- Student: Oskar Michel
- Mentor: Christian Bizer

Some papers as starting point

- Wadden, D. et al., 2020. Fact or Fiction: Verifying Scientific Claims. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7534–7550.
- Dmonte et al. 2024, Claim Verification in the Age of Large Language Models: A Survey. arXiv:2408.14317, 2024.
- Asai, A., He, J., Shao, R. et al. Synthesizing scientific literature with retrieval-augmented language models. Nature (2026).

3. RAG-Driven Data Cleaning with PyDI

- Experimental topic
- Student: Marmee Pandya
- Mentor: Ralph Peeters

Some papers as starting point

- Ahmad, M.S. et al., 2023. RetClean: Retrieval-based Data Cleaning using Foundation Models and Data Lakes. arXiv preprint arXiv:2303.16909.
- Chen, M. et al., 2025. Empowering Tabular Data Preparation with Language Models: Why and How?. arXiv preprint arXiv:2508.01556.
- https://github.com/wbsg-uni-mannheim/PyDI

4. LLMs (Agents) for Data Normalization

- Experimental topic
- Student: Shiqi Xu
- Mentor: Aaron Steiner

Some papers as starting point

- Brinkmann, A., Baumann, N. and Bizer, C., 2024. Using LLMs for the Extraction and Normalization of Product Attribute Values. In ADBIS 2024. LNCS, vol 14918. Springer, pp.217–230.
- Chen, M. et al., 2025. Empowering Tabular Data Preparation with Language Models: Why and How?. arXiv preprint arXiv:2508.01556.
- https://github.com/wbsg-uni-mannheim/PyDI

5. Small vs. Large LLMs for Training Data Generation

- Experimental topic
- Student: Abhaykumar Narendrakumar Monasara
- Mentor: Christian Bizer

Some papers as starting point

- Zhang, Z. et al., 2025. A Deep Dive Into Cross-Dataset Entity Matching with Large and Small Language Models. In Proceedings of the 28th International Conference on Extending Database Technology.
- Tan, Z. et al., 2024. Large Language Models for Data Annotation and Synthesis: A Survey. In EMNLP 2024, pp. 930–957.
- https://github.com/wbsg-uni-mannheim/MatchGPT/tree/main/LLMForEM

6. Query Answering over Data Lakes

- Literature topic
- Student: Pascal Banser
- Mentor: Aaron Steiner

Some papers as starting point

- Li, Z. et al., 2025. DocDB: A Database for Unstructured Document Analysis. Proceedings of the VLDB Endowment, 18(12), pp.5387-5390.
- Shankar, S. et al., 2025. DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing. Proceedings of the VLDB Endowment, 18(9), pp.3035-3048.
- Sun, Z. et al., 2025. QUEST: Query Optimization in Unstructured Document Analysis. Proceedings of the VLDB Endowment, 18(11), pp.4560-4573.

7. Reducing the Resource Consumption of LLM Agents

- Experimental topic
- Student: Luis Reifferscheid
- Mentor: Aaron Steiner

Some papers as starting point

- Du, S. et al., 2025. A Survey on the Optimization of Large Language Model-based Agents. arXiv preprint arXiv:2503.12434.
- Zhang, Q. et al., 2025. Agentic Plan Caching: Test-Time Memory for Fast and Cost-Efficient LLM Agents. In NeurIPS 2025.

8. Resource-efficient Agentic Plan Caching for EM

- Experimental topic
- Student: Noyan Morali
- Mentor: Ralph Peeters

Some papers as starting point

- Zhang, Q. et al., 2025. Agentic Plan Caching: Test-Time Memory for Fast and Cost-Efficient LLM Agents. In NeurIPS 2025.
- Peeters, R. et al., 2025. Entity Matching using Large Language Models. In Proceedings of the 28th International Conference on Extending Database Technology.
- https://github.com/wbsg-uni-mannheim/PyDI

9. Data Serialization Formats for LLMs

- Experimental topic
- Student: Krittika Sardar
- Mentor: Aaron Steiner

Some papers as starting point

- Yang, J. et al., 2025. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. arXiv preprint arXiv:2505.20139.
- TOON Format, 2024. Token-Oriented Object Notation. https://github.com/toon-format/toon
- ZON Format, 2024. Zero Overhead Notation. https://github.com/ZON-Format/ZON

10. Descriptive Agent Trajectory Mining

- Experimental topic
- Student: Danny Daewon Lee
- Mentor: Christian Bizer

Some papers as starting point

- Mohammadi, M. et al., 2025: Evaluation and Benchmarking of LLM Agents: A Survey. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2025).
- Ou, T. et al., 2025: AgentDiagnose: An Open Toolkit for Diagnosing LLM Agent Trajectories. EMNLP 2025.
- Peeters, R. et al., 2025: WebMall–A Multi-Shop Benchmark for Evaluating Web Agents. arXiv preprint arXiv:2508.13024.
- van der Aalst, W., 2016: Process Mining – Data Science in Action. Springer.

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 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)

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)

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?