Intelligent agents: individual development assignment

Guilherme Amorim

April 2025

Introduction

Intelligent agents[1]

- Sense environment
- Take action autonomously to achieve specific goals

Potential value[2,3]

- Efficiency gains (automation, complex decision-making)
- Foster innovation (new business models)

Use-cases[4-6]

- Healthcare (precision diagnosis, tailored treatment recommendations)
- eCommerce (personalized shopping experiences)
- Finance (fraud detection, algorithmic trading)

Assignment specifications

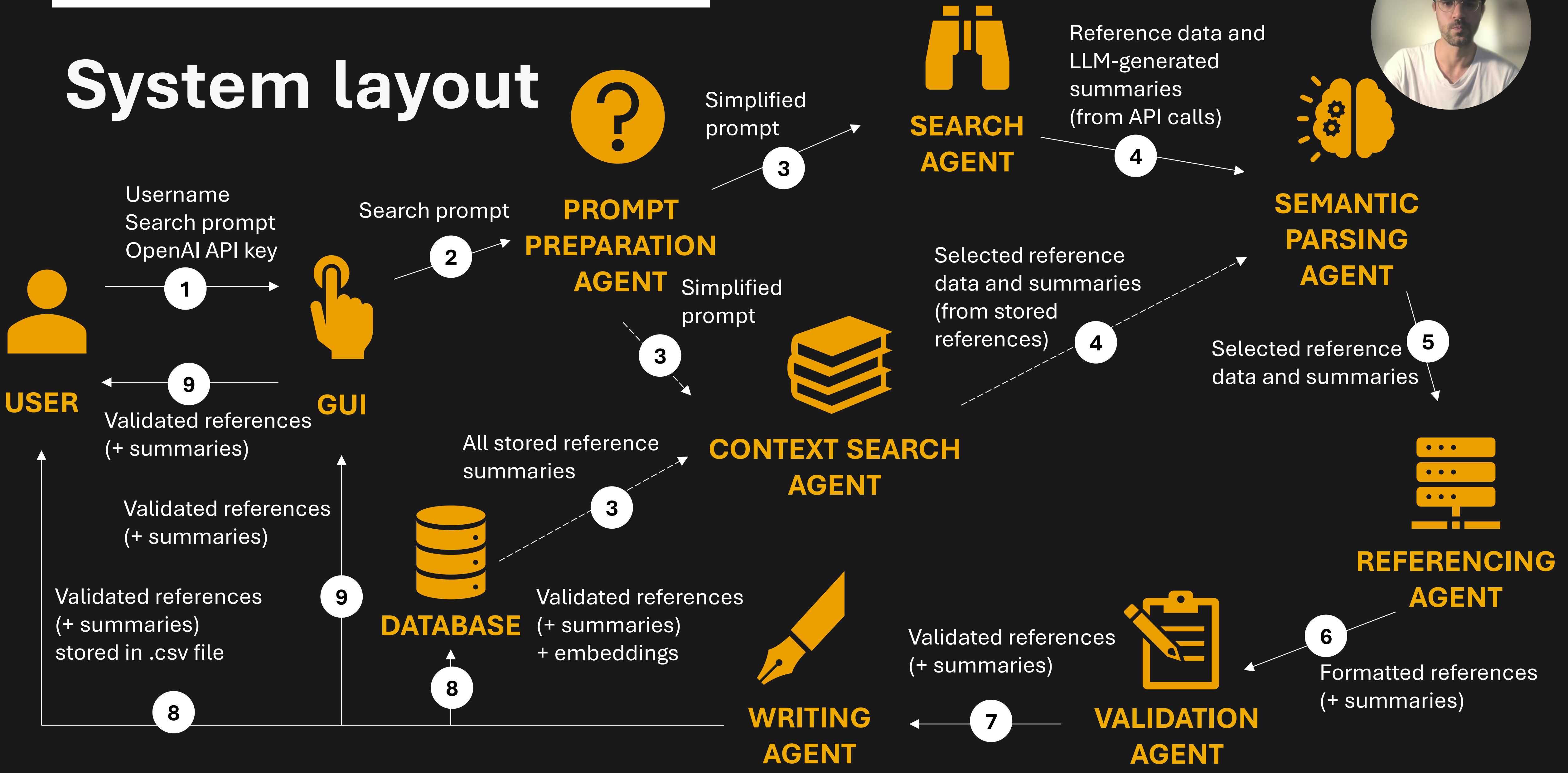
1. The agent can **identify and retrieve data.**
2. The **data is processed** in some way.
2. The processed **information is stored/saved/presented**.

Chosen domain: academic research online

- finding results on a website based on search terms (e.g., social media or a search engine)
- extracting the data and
- sending to an offline location.

[1]Russel and Norvig, 2021; [2]Brynjolfsson and McAfee, 2014; [3]Chui, Manyika and Miremadi, 2016; [4]Topol, 2019; [5]Cao, 2021; [6]Salau *et al.*, 2022

System layout



USER

1 Username
Search prompt
OpenAI API key

GUI

2 Search prompt

PROMPT PREPARATION AGENT

3 Simplified prompt

SEARCH AGENT

3 Simplified prompt

CONTEXT SEARCH AGENT

3 All stored reference summaries

4 Reference data and LLM-generated summaries (from API calls)

4 Selected reference data and summaries (from stored references)

SEMANTIC PARSING AGENT

5 Selected reference data and summaries

REFERENCING AGENT

6 Formatted references (+ summaries)

VALIDATION AGENT

7 Validated references (+ summaries)

WRITING AGENT

8 Validated references (+ summaries) + embeddings

DATABASE

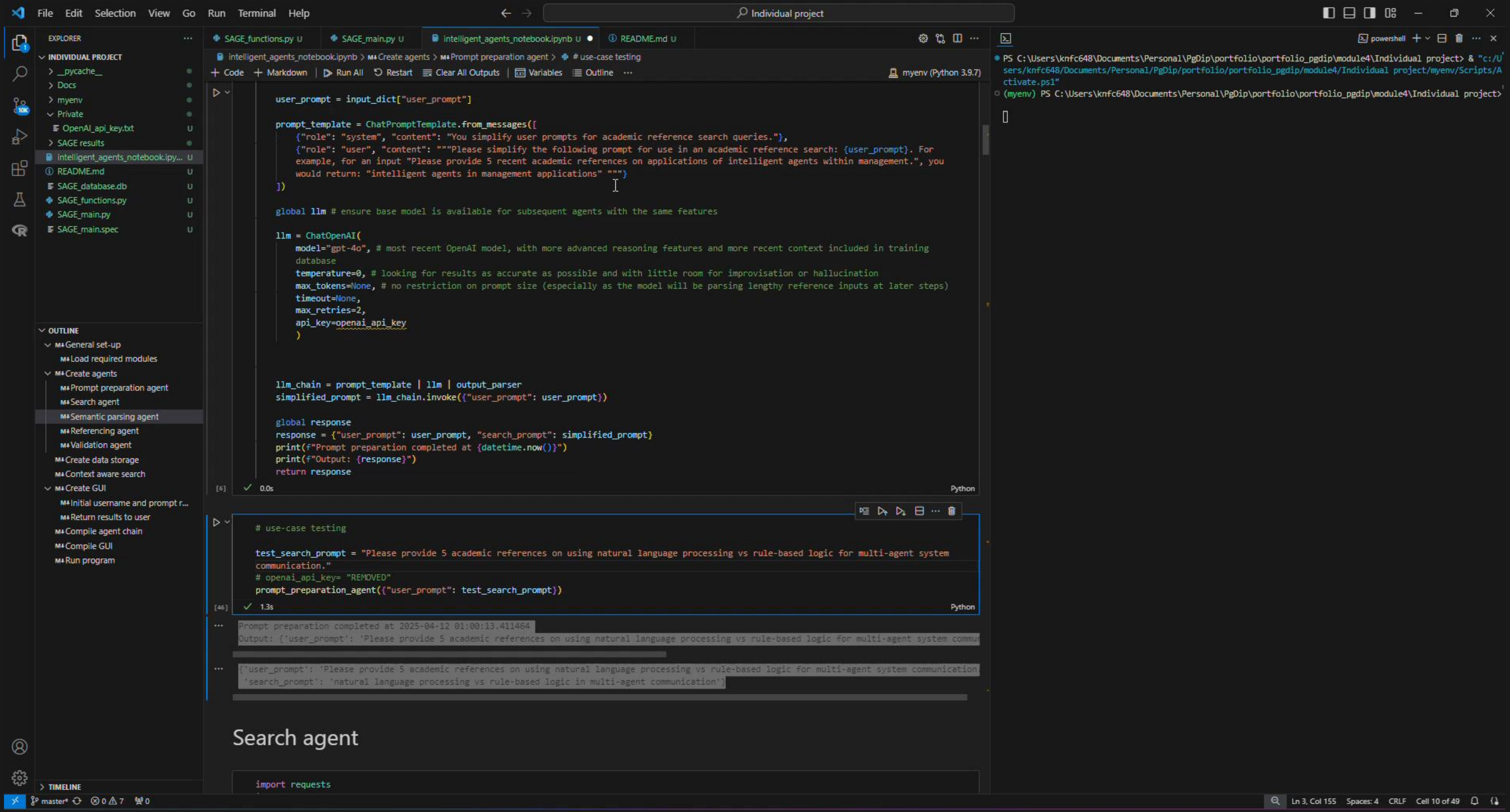
8 Validated references (+ summaries) stored in .csv file

9 Validated references (+ summaries)

9 Validated references (+ summaries)

Benji, 2025; Woolridge, 2009; Juziuk, Weyns and Holvoet, 2014; Russel and Norvig, 2021

Application demo



```python
    user_prompt = input_dict["user_prompt"]

    prompt_template = ChatPromptTemplate.from_messages([
        {"role": "system", "content": "You simplify user prompts for academic reference search queries."},
        {"role": "user", "content": """Please simplify the following prompt for use in an academic reference search: {user_prompt}. For
        example, for an input "Please provide 5 recent academic references on applications of intelligent agents within management.", you
        would return: "intelligent agents in management applications" """}
    ])

    global llm # ensure base model is available for subsequent agents with the same features

    llm = ChatOpenAI(
        model="gpt-4o", # most recent OpenAI model, with more advanced reasoning features and more recent context included in training
        database
        temperature=0, # looking for results as accurate as possible and with little room for improvisation or hallucination
        max_tokens=None, # no restriction on prompt size (especially as the model will be parsing lengthy reference inputs at later steps)
        timeout=None,
        max_retries=2,
        api_key=openai_api_key
        )


    llm_chain = prompt_template | llm | output_parser
    simplified_prompt = llm_chain.invoke({"user_prompt": user_prompt})

    global response
    response = {"user_prompt": user_prompt, "search_prompt": simplified_prompt}
    print(f"Prompt preparation completed at {datetime.now()}")
    print(f"Output: {response}")
    return response
```

[6] ✓ 0.0s Python

```python
# use-case testing

test_search_prompt = "Please provide 5 academic references on using natural language processing vs rule-based logic for multi-agent system
communication."
# openai_api_key= "REMOVED"
prompt_preparation_agent({"user_prompt": test_search_prompt})
```

[46] ✓ 1.3s Python

```
Prompt preparation completed at 2025-04-12 01:00:13.411464
Output: {'user_prompt': 'Please provide 5 academic references on using natural language processing vs rule-based logic for multi-agent system commu
```

```
{'user_prompt': 'Please provide 5 academic references on using natural language processing vs rule-based logic for multi-agent system communication.
 'search_prompt': 'natural language processing vs rule-based logic in multi-agent communication'}
```

Search agent

```python
import requests
```

```
PS C:\Users\knfc648\Documents\Personal\PgDip\portfolio\portfolio_pgdip\module4\Individual project> & "c:/Users/knfc648/Documents/Personal/PgDip/portfolio/portfolio_pgdip/module4/Individual project/myenv/Scripts/Activate.ps1"
(myenv) PS C:\Users\knfc648\Documents\Personal\PgDip\portfolio\portfolio_pgdip\module4\Individual project>
```

Development issues and troubleshooting

LLM reference search

✗ Hallucinations[1]
Outdated references
Unable to circumvent with prompt engineernig

✓ Add API calls to structured databases
(LLM for prompt simplification
and reference prioritising)

Referencing

✗ Complex referencing rules
(depending on citation source and chosen format)

✓ Simple and versatile scheme

Reference validation

✗ No clear way to identify problematic data
(e.g. pages, authors)[2]
Complex architecture required to prompt user for corrections

✓ Simple data type checks and URL validation
LLM cross-check (formatting)

LLM output parsing

✗ Always return string outputs
Difficult to ensure correct formating[3]

✓ Extensive data formatting enforcement
(lists/dictionaries)

Reference storage

✗ Avoid saving duplicated references

⏭ Removing duplicates before storage
(but how best to identify? DOI? combinations?)[4]

Context search

✗ Possibly helpful, but difficult to distinguish when old vs new references need to be prioritised

⏭ Allow user to turn context-aware search off

Plus: setting up API connections, managing dependencies, library updates, langchain documentation, etc

[1]Maes, 2024; [2]Choi et al., 2023; [3]Liu et al., 2024; [4]McKeown and Mir, 2021

Critical reflection

STRENGTHS

- Natural language queries (simplify user interactions)[1]
- API calls (overcome LLM hallucination issues, provide solid foundation for LLM parsing)[1, 2]
- Flexible and scalable database connections
- NLP-based agents (reference prioritisation, flexible formatting implementation)[3]
- Data validation (structured and flexible scheme)
- LLM-based reference summaries (simplify inspection)
- Retrieval of public data only
(no privacy/ethical concerns)[4]
- Local data storage
(low cost, high degree of user control)
- Modular code implementation

WEAKNESSES

- Archaic GUI
- Multiple dependencies
- Limited literature database scope
- Suboptimal reference validation scope
(and limited capacity to handle and resolve issues)
- NLP-based agents
(difficulty handling ambiguous or poorly-defined prompts, LLM bias, possibility of hallucination)[2,5]
- Poor scalability and real-time processing capability
- Simple in-memory vector database indexing strategy
- Limited error handling / debugging features

[1] Benji, 2025; [2] Maes, 2024; [3] Lalwani et al., 2024; [4] Data protection explained - European Commission, no date; [5] Yadav, Patel and Shah, 2021

Possible improvements

Area	Improvement
Referencing	Implement a formal reference format (and allow user to select others)
Search prompt	Integrate LLM-generated disambiguation or specification requests to the user
LLM agent	Allow user to edit default prompt instructions, or specify specific LLM of interest
Verification	Integrate user verification / approval before saving references
Validation	Additional reference validation rules or more complex validation steps
Context-aware search	Turn on / off depending on user needs
Database indexing	Develop indexing strategy to avoid saving duplicates
API handling	Batch processing, exponential back-off, usage/monitoring alerts
Error management	Comprehensive exception handling, diagnostic logging, and fallback recovery procedures

Conclusions and learning reflections

- Developed a multi-agent system that successfully:
  - ✓ finds results on (a) website(s) based on search terms (CrossRef, arXiv, Pubmed)
  - ✓ extracts and processes the data (API calls + LLM and rule-based processing)
  - ✓ stores the data in an offline location (structured + vector database, .csv file, and GUI)
- Explored many important and new computer science concepts (data validation and parsing, object-oriented programming, control flow, error handling/debugging/testing, GUI, etc.)
- Experienced advantages and pitfalls of different agent-based architectures (rule-based vs NLP) and challenges of working with multi-agent systems
- Identified strengths, weaknesses, and possible improvements for my proposed implementation
- Wrote my first computer program (with a little help from a few LLMs)

References

- Benji, N. (2025) *LLMs Vs. Deterministic Logic — Overcoming Rule-Based Evaluation Challenges*, *Medium*. Available at: https://blog.gopenai.com/llms-vs-deterministic-logic-overcoming-rule-based-evaluation-challenges-8c5fb7e8fe46 (Accessed: 13 April 2025).
- Brynjolfsson, E. and McAfee, A. (2014) *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. 1st edn. W. W. Norton & Company.
- Cao, L. (2021) ‘AI in Finance: Challenges, Techniques and Opportunities’. arXiv. Available at: https://doi.org/10.48550/arXiv.2107.09051.
- Choi, W. *et al.* (2023) ‘Building an annotated corpus for automatic metadata extraction from multilingual journal article references’, *PLOS ONE*, 18(1), p. e0280637. Available at: https://doi.org/10.1371/journal.pone.0280637.
- Chui, M., Manyika, J. and Miremadi, M. (2016) *Where machines could replace humans—and where they can’t (yet) | McKinsey*, *McKinsey Digital*. Available at: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/where-machines-could-replace-humans-and-where-they-cant-yet (Accessed: 13 April 2025).
- *Data protection explained - European Commission* (no date). Available at: https://commission.europa.eu/law/law-topic/data-protection/data-protection-explained_en (Accessed: 13 April 2025).
- Juziuk, J., Weyns, D. and Holvoet, T. (2014) ‘Design Patterns for Multi-agent Systems: A Systematic Literature Review’, in O. Shehory and A. Sturm (eds) *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks*. Berlin, Heidelberg: Springer, pp. 79–99. Available at: https://doi.org/10.1007/978-3-642-54432-3_5.
- Lalwani, A. *et al.* (2025) ‘Autoformalizing Natural Language to First-Order Logic: A Case Study in Logical Fallacy Detection’. arXiv. Available at: https://doi.org/10.48550/arXiv.2405.02318.
- Liu, Y. *et al.* (2024) ‘Are LLMs good at structured outputs? A benchmark for evaluating structured output capabilities in LLMs’, *Information Processing & Management*, 61(5), p. 103809. Available at: https://doi.org/10.1016/j.ipm.2024.103809.
- Maes, S. (2024) ‘Fixing Reference Hallucinations of LLMs’. Available at: https://zenodo.org/records/14791389 (Accessed: 13 April 2025).
- McKeown, S. and Mir, Z.M. (2021) ‘Considerations for conducting systematic reviews: evaluating the performance of different methods for de-duplicating references’, *Systematic Reviews*, 10(1), p. 38. Available at: https://doi.org/10.1186/s13643-021-01583-y.
- Russel, S. and Norvig, P. (2021) *Artificial intelligence: a modern approach*. 4th edn. Upper Saddle River, NJ : Prentice Hall: Pearson. Available at: https://doi.org/10.1109/MSP.2017.2765202.
- Salau, L. *et al.* (2022) ‘State-of-the-Art Survey on Deep Learning-Based Recommender Systems for E-Learning’, *Applied Sciences*, 12(23), p. 11996. Available at: https://doi.org/10.3390/app122311996.
- Topol, E. (2019) *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again.* 1st edn. NY: Basic Books. Available at: https://psnet.ahrq.gov/issue/deep-medicine-how-artificial-intelligence-can-make-healthcare-human-again (Accessed: 13 April 2025).
- Woolridge, M. (2009) *An Introduction to MultiAgent Systems, 2nd Edition | Wiley*. 2nd edn. Chichester: John Wiley & Sons. Available at: https://www.wiley.com/en-us/An+Introduction+to+MultiAgent+Systems%2C+2nd+Edition-p-9780470519462 (Accessed: 11 February 2025).
- Yadav, A., Patel, A. and Shah, M. (2021) ‘A comprehensive review on resolving ambiguities in natural language processing’, *AI Open*, 2, pp. 85–92. Available at: https://doi.org/10.1016/j.aiopen.2021.05.001.