Natural Language Processing

Exercise – Summer term 2026

klara.gutekunst@uni-kassel.de

Organization

Studienleistung

- The *Studienleistung* is required to be eligible for the exam.
- Active participation in the exercise sessions is expected.
- Programming project [Modulhandbuch] in groups of **2–3 students**.
- Code must be submitted at the end of the semester.

Organization

Studienleistung

Programming Project: **Forensic Linguist Agent**

- Develop a modular agent system for forensic authorship analysis.
- Focus on *idiolectal style*: how a text is written rather than what it is about.
- A coordinator agent selects and executes reusable linguistic analysis skills.
- Ensure a transparent and reproducible workflow with inspectable intermediate results and uncertainty.
- Evaluate the system on authorship datasets and compare agent-based workflows to simpler baselines.

Agentic Workflow

What is an agent?

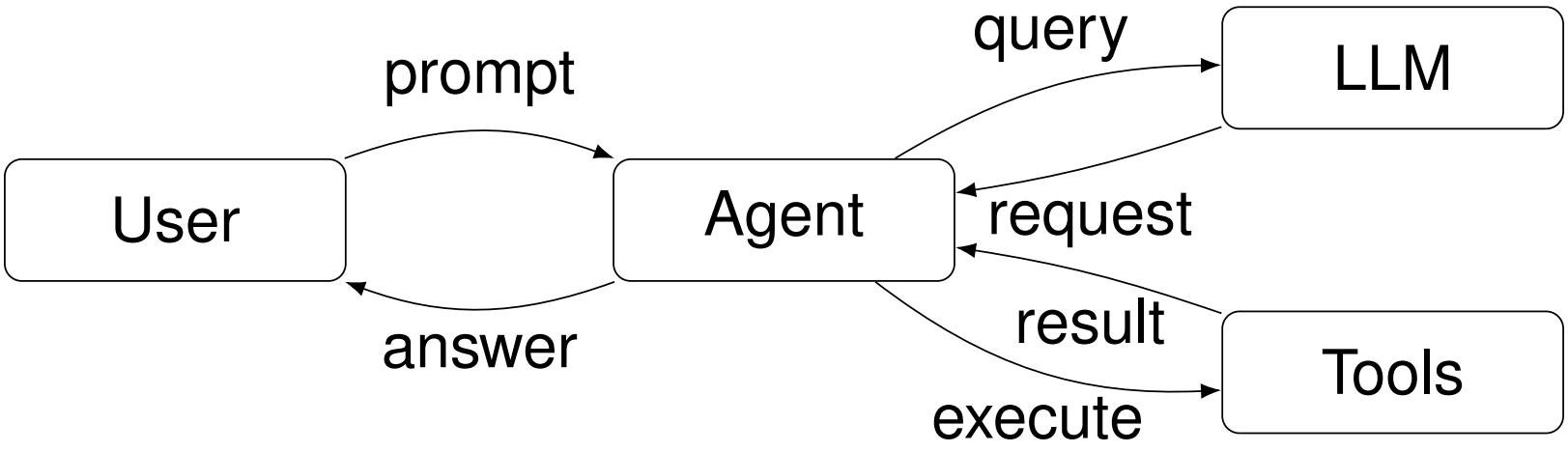
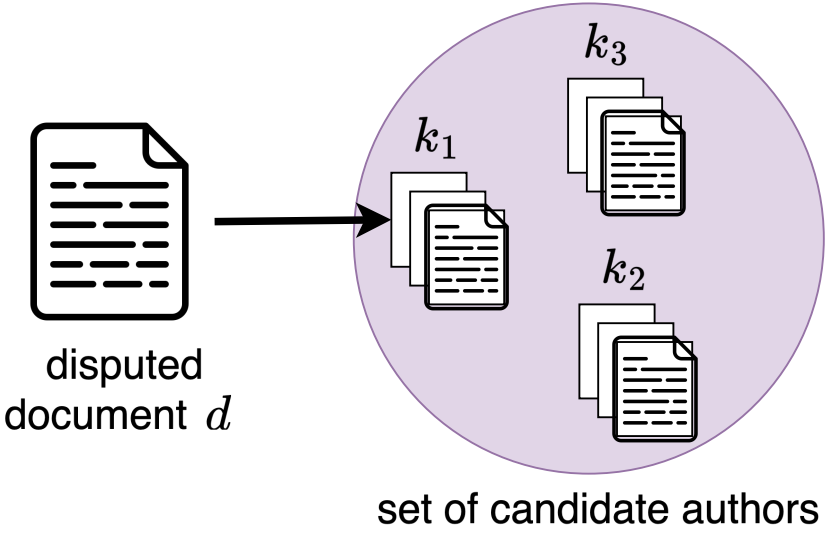
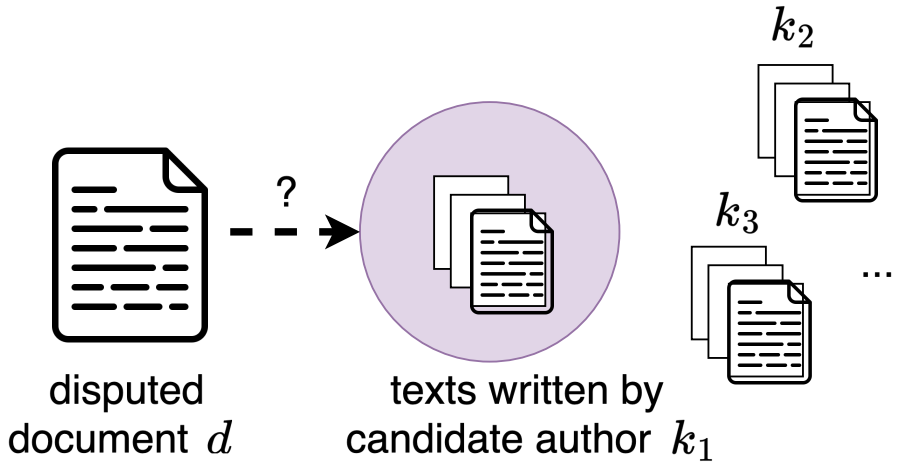


Figure: Simplified agentic workflow. A user prompt is processed by an agent that queries an LLM. The LLM may request tool calls, which are executed by the agent and returned as results. This loop continues until the agent produces a final answer.

Authorship Analysis Tasks



(a) Authorship Attribution



(b) Authorship Verification

Authorship Analysis

- **Goal:** Determine who wrote a given text based on writing style.
- Focus on style (idiolect), not content
- Features include [Stamatatos, Neal et al., Bouanani et al.]:
  - character n-grams
  - error patterns
  - frequent words
  - vocabulary richness
  - ...
- Applications: plagiarism detection, forensic linguistics, ...

Studienleistung

Details

Your tasks include:

- Implement tools for linguistic analysis (e.g., linguistic feature extraction, stylometric analysis).
- Persist imediate results and uncertainty in a transparent way.
- Connect with a backend LLM.
- Identify risks and limitations of the agentic workflow and implement mitigation strategies.
- Evaluate the agent on benchmark datasets and compare to non-agentic baselines.
- Write an 4-page report summarizing your approach, results, and insights in a two-column format including figures, tables, and references.

Studienleistung

Useful Tools/Libraries

- NLTK for tokenization and n-gram-based analysis.
- spaCy for linguistic preprocessing and syntactic annotation.
- scikit-learn for baselines, feature processing, and evaluation.
- n8n for workflow automation.
- LangChain as engineering platform for agents.
- Visual Studio Code with Dev Containers for reproducible development.
- Visual Studio Code extension Continue for connecting the project to the private OpenAI-compatible Webis-hosted LLM backend.
- Agentic Engineering Post
- Medium Article on Skills
- Blabladador API Access
- . . .

Studienleistung

Next Steps

- Form groups of **2–3 students**.
- One group member should submit the following information for the entire group by **May 05, 2026** (via email or Discord):
  - Names
  - Matriculation numbers
  - Email addresses
  - Group name
  - URL of a shared GitHub repository for the project
    - Use Dev Containers
- After registration, you will receive an API key for accessing Webis-hosted LLMs.
  - Handle the API key securely (`.env` file) and do not share it with unauthorized parties.

LLMs hosted by Webis

Python example

```
from openai import OpenAI
client = OpenAI(
  base_url="https://chat.web.webis.de/openai/",
  api_key="YOUR_API_KEY",
)
models = client.models.list()

print("Available models:")
for m in models.data:
  print(m.id)

print("Example response:")
response = client.chat.completions.create(
  model="deepseek-r1-8b",
  messages=[
    {"role": "user", "content": "Why is the sky blue?"}
  ],
)
print(response.choices[0].message.content)
```