Community of Practice KI*PerWeb*

Austausch zur Nutzung und Entwicklung KI-gestützter Webanwendungen

KIPERWEB

f-bb Forschungsinstitut Betriebliche Bildung

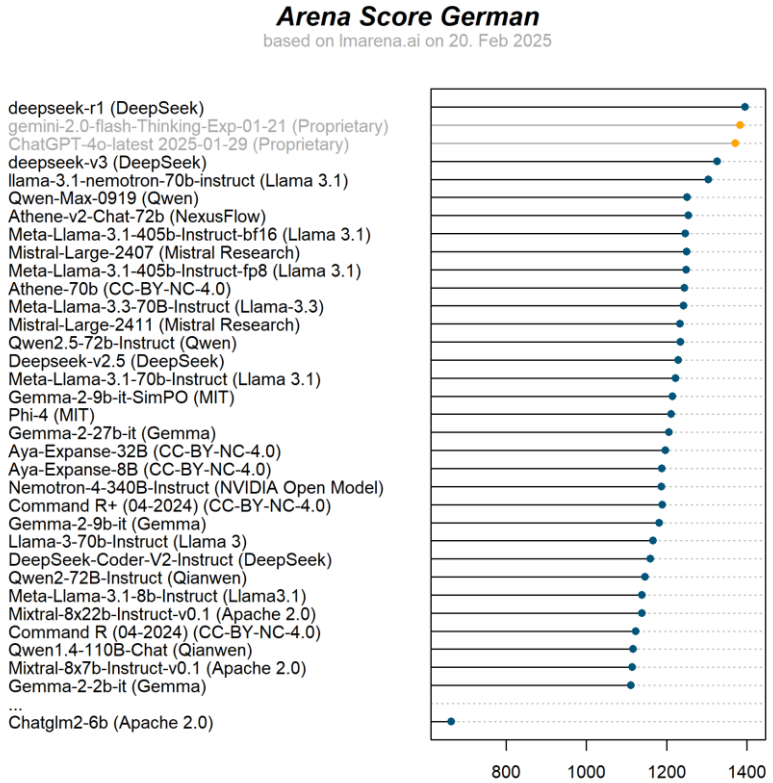
Agenda

- Update
  - News & Leaderboard-Update
- Input
  - „KI-Agenten“
- Diskussion

News & Leaderboard-Update (22.01.2025)

Vergleichsweise wenige Änderungen in Kategorie „German“:

- Trotz neuer Varianten von Gemini und ChatGPT bleibt Deepseek-r1 auf Platz 1.
- Arena-Scores von *nicht*-proprietären Modelle sind rechts ausgewiesen sofern sie mindestens das Niveau von Gemma-2-2b-it erreichen (nebst Schlusslicht Chatglm2-6b):
  - Phi-4 (14B) liegt nun hinter Gemma-2-9b-it-SimPO, welches damit wieder das beste Modell unter freier (MIT-)Lizenz ist
  - Nach wie vor nicht im Rennen, aber trotzdem hervorragend ist Gemma-2-27b-it-SimPo



Datengrundlage: https://chat.lmsys.org/?leaderboard

Fokusthema: AI Agents

- „AI Agent“

(rechts visualisiert von FLUX.1-schnell)

Quelle: https://huggingface.co/spaces/black-forest-labs/FLUX.1-schnell

Intelligente Agenten (vor 2022)

- „Übertragen auf die Informatik ist ein Agent ein Programm, welches im Auftrag anderer selbstständig eine Aufgabe erfüllt. Auftraggeber können dabei sowohl der Mensch als auch ein anderes Programm, möglicherweise ein anderer Agent sein.“ *(Lämmel & Cleve, 2008)*
  - Für die selbstständige Aufgabenerfüllung müssen Agenten idR. nicht nur wahrnehmen und reagieren, sondern planen, kommunizieren und ggf. auch lernen (ibid.)
- „Ein Agent besitzt Sensoren, mit denen er seine Umgebung wahrnehmen kann, und Aktuatoren, durch die er handelt“ (*Russell & Norvig, 2012*)

Quellen: Lämmel, U., & Cleve, J. (2008). Künstliche Intelligenz (Vol. 3). München: Hanser.
Russell, S., & Norvig, P. (2012). *Künstliche Intelligenz* (Vol. 2). München: Pearson Studium.

Agents-Example: Wikipedia-Agent (April 2023)

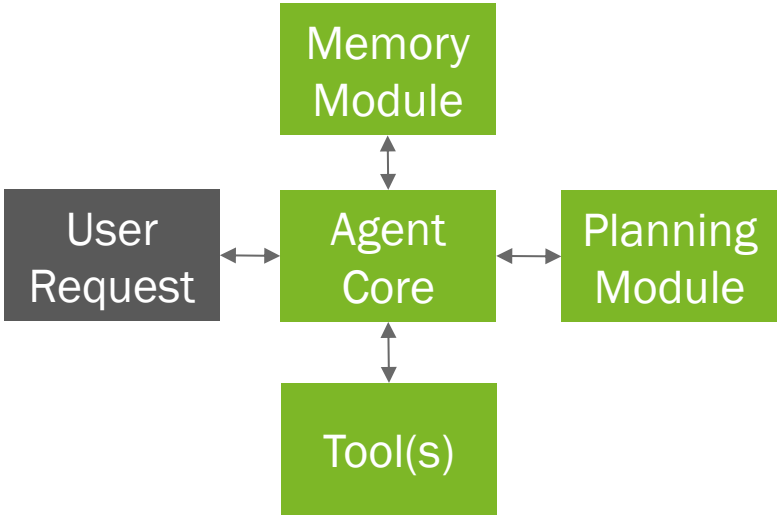
```python
from langchain.agents import Tool, initialize_agent
from langchain.utilities import WikipediaAPIWrapper
tools=[Tool(
  name="Wikipedia",
  func=WikipediaAPIWrapper(top_k_results=2).run,
  description="A wrapper around Wikipedia."+
    " Useful for when you need to answer general questions about"+
    " people, places, companies, historical events, or other subjects."+
    " Input should be a search query.")]
agent = initialize_agent(tools, llm, agent="zero-shot-react-description", verbose=True)
agent("What is the meaning of life?")
```

Source:
https://github.com/AndreasFischer1985/code-snippets/blob/master/py/LangChain_HuggingFace_examples.py

LLM-Agents

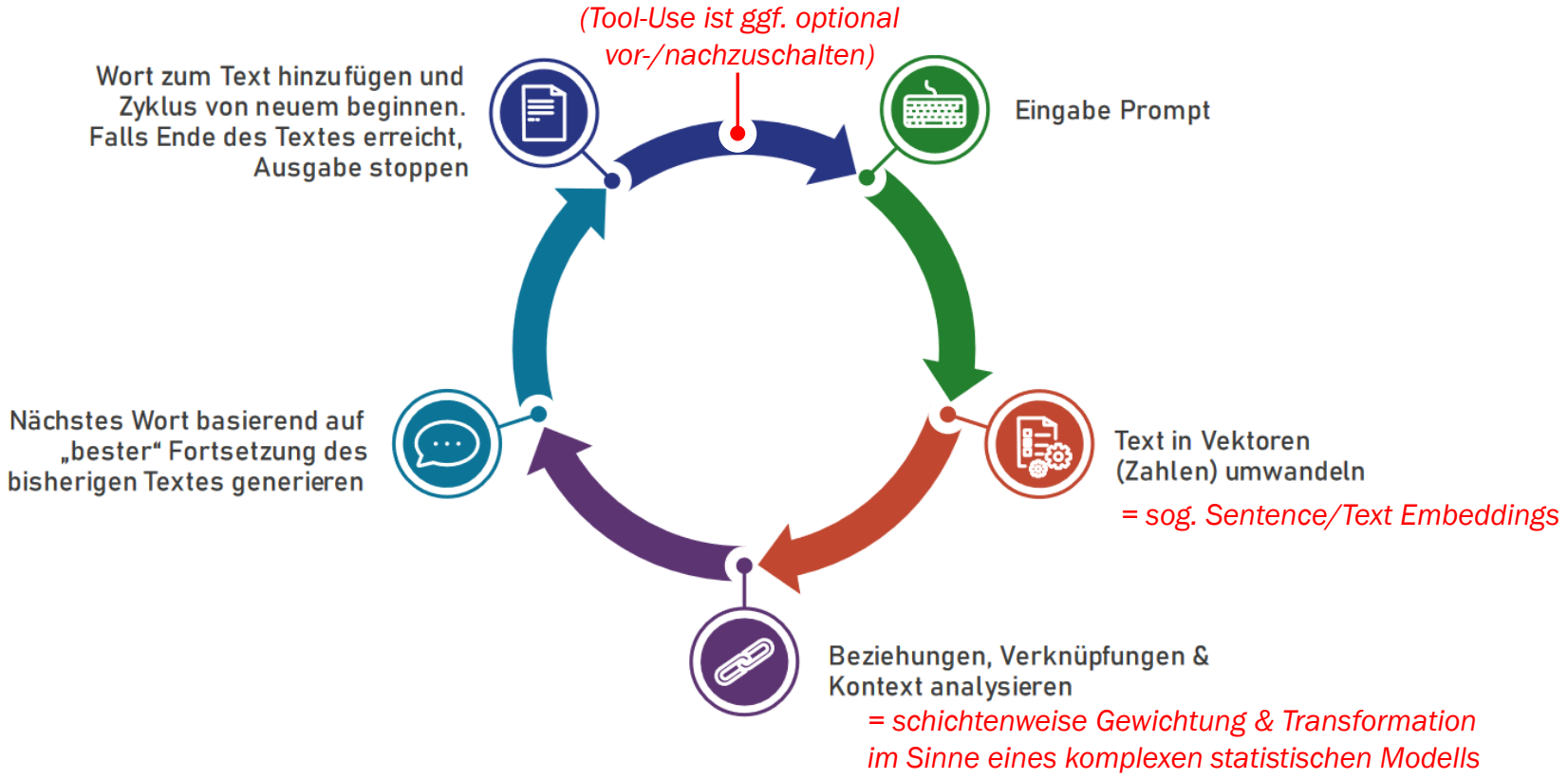
- **LLM-powered agents** “can be described as a system that can use an LLM to reason through a problem, create a plan to solve the problem, and execute the plan with the help of a set of tools. In short, agents are a system with complex reasoning capabilities, memory, and the means to execute tasks.” (NVIDIA, 2023)
- Vgl. Session 4 zu Tool-Use:

https://github.com/AndreasFischer1985/KIPerWeb/blob/main/cop/slides/240614-4-CoP_KIPerWeb.pdf



https://developer.nvidia.com/blog/introduction-to-llm-agents/

Exkurs zur Funktionsweise von LLMs



Quelle: https://www.pedocs.de/volltexte/2024/28659/pdf/Huebsch_et_al_2024_Sprachgewandt_in_die_Zukunft.pdf

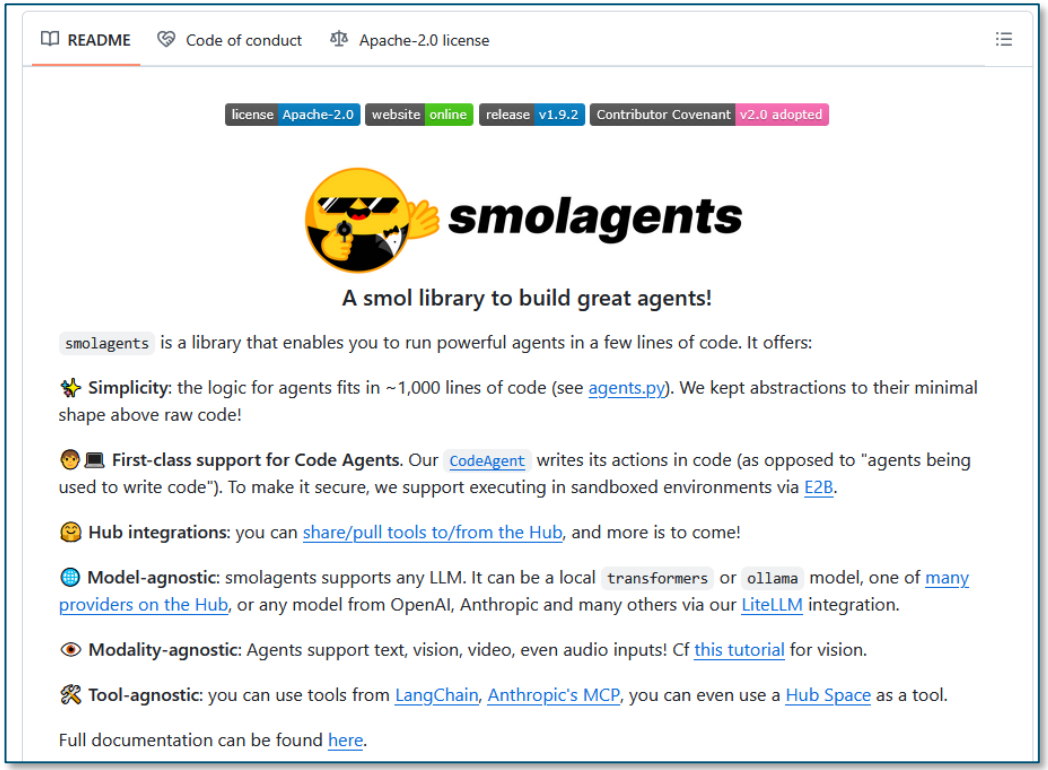
Huggingface 2023: Transformers Agents

- We provide two types of agents, based on the main Agent class:
  - **CodeAgent** acts in one shot, generating code to solve the task, then executes it at once.
  - **ReactAgent** acts step by step, each step consisting of one thought, then one tool call and execution. It has two classes:
    - ReactJsonAgent writes its tool calls in JSON.
    - ReactCodeAgent writes its tool calls in Python code.

Source: https://twitter.com/huggingface/status/1656334778407297027

https://huggingface.co/docs/transformers/main_classes/agent

Huggingface 2025: smolagents



README | Code of conduct | Apache-2.0 license

license Apache-2.0 | website online | release v1.9.2 | Contributor Covenant v2.0 adopted

smolagents

A smol library to build great agents!

`smolagents` is a library that enables you to run powerful agents in a few lines of code. It offers:

✨ **Simplicity**: the logic for agents fits in ~1,000 lines of code (see agents.py). We kept abstractions to their minimal shape above raw code!

🧑‍💻 **First-class support for Code Agents**. Our `CodeAgent` writes its actions in code (as opposed to "agents being used to write code"). To make it secure, we support executing in sandboxed environments via E2B.

🤗 **Hub integrations**: you can share/pull tools to/from the Hub, and more is to come!

🌐 **Model-agnostic**: smolagents supports any LLM. It can be a local `transformers` or `ollama` model, one of many providers on the Hub, or any model from OpenAI, Anthropic and many others via our LiteLLM integration.

👁️ **Modality-agnostic**: Agents support text, vision, video, even audio inputs! Cf this tutorial for vision.

🛠️ **Tool-agnostic**: you can use tools from LangChain, Anthropic's MCP, you can even use a Hub Space as a tool.

Full documentation can be found here.