TUBAF
Die Ressourcenuniversität.
Seit 1765.

ALIMA

Von der KI zur Praxis – LLM unterstützte Sacherschließung in Bibliotheken

Conrad Hübler
Universitätsbibliothek "Georgius Agricola"

27. Januar 2026

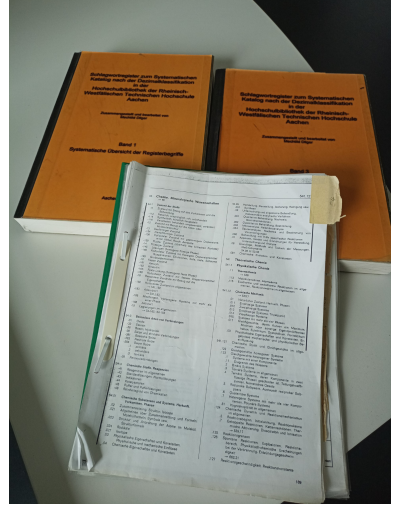
Ausgangslage an der Universitätsbibliothek

- Routineaufgabe: Erschließung von Neuerwerbungen
  - Verbale Erschließung: GND-Schlagworte
  - Klassifikatorische Erschließung: DK/RVK/DDC
- Probleme im Alltag:
  - Veraltete DK-Register (1960er!)
  - Interdisziplinäre Forschung
  - Manuelle Katalog-Recherche
  - Fachwissen erforderlich

Neuerwerbungen

Ausgangslage an der Universitätsbibliothek

- Routineaufgabe: Erschließung von Neuerwerbungen
  - Verbale Erschließung: GND-Schlagworte
  - Klassifikatorische Erschließung: DK/RVK/DDC
- Probleme im Alltag:
  - Veraltete DK-Register (1960er!)
  - Interdisziplinäre Forschung
  - Manuelle Katalog-Recherche
  - Fachwissen erforderlich



Gedruckte DK-Klassifikationen

Ausgangslage an der Universitätsbibliothek

- Routineaufgabe: Erschließung von Neuerwerbungen
  - Verbale Erschließung: GND-Schlagworte
  - Klassifikatorische Erschließung: DK/RVK/DDC
- Probleme im Alltag:
  - Veraltete DK-Register (1960er!)
  - Interdisziplinäre Forschung
  - Manuelle Katalog-Recherche
  - Fachwissen erforderlich

Automatisierte Erschließung (DALL·E 3 via OmniGPT)

Können wir LLMs für automatisierte Sacherschließung nutzen?

Probleme mit naiver LLM-Nutzung

Halluzinationen

- LLM erfindet Schlagworte, die nicht in der GND existieren
- Wahrscheinlichkeitsbasiert, nicht faktisch korrekt

Kontextproblem

- Zu viele mögliche Klassifikationen
- LLM hat wenig Kontext über verfügbare Klassen

Retraining/Feintuning aufwendig

- GND wird ständig aktualisiert
- Ein speziell trainiertes/feingetuntes Modell wäre zu aufwendig
- Qualität der Trainingsdaten

Multi-Label-Problem

- Extreme Anzahl möglicher Label-Kombinationen

Lösung: Retrieval Augmented Generation (RAG)

ALIMA – AI-powered Library Indexing and Metadata Assignment

Was ist ALIMA?

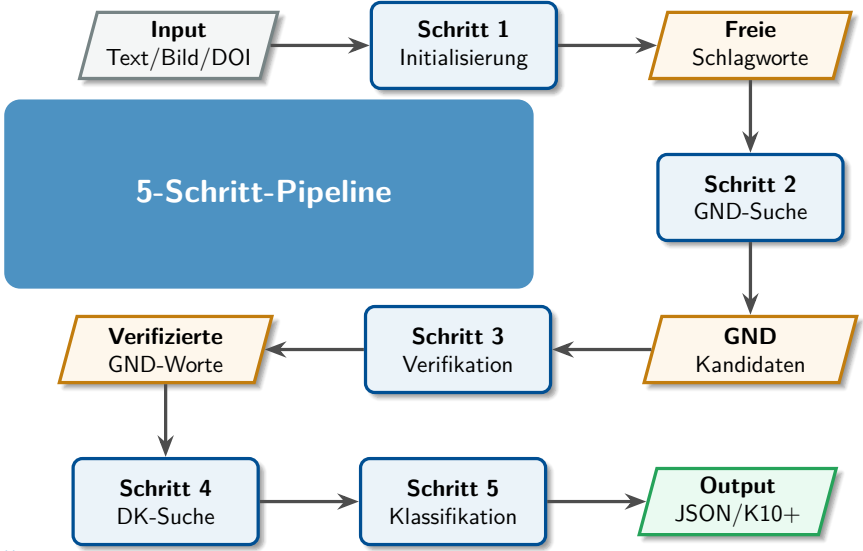
- Python-basierte Desktop-Anwendung + Command Line Interface +WebApp
- Verbindet LLMs mit bibliothekarischen Systemen
- Unterstützt GND-konforme Schlagworte und DK/RVK-Klassifikation
- Flexible Multi-Provider-Unterstützung

Kernidee: Pipeline

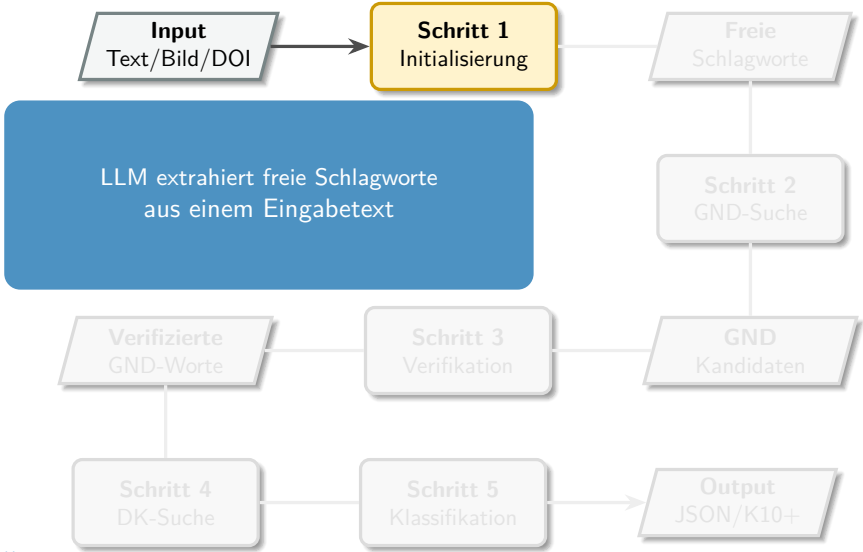
- Text → freie Schlagworte → GND-Suche → verifizierte Schlagworte → DK-Klassifikation
- Jeder Schritt spezifiziert den Kontext
- Halluzinationen werden minimiert

DALL·E 3 via OmniGPT

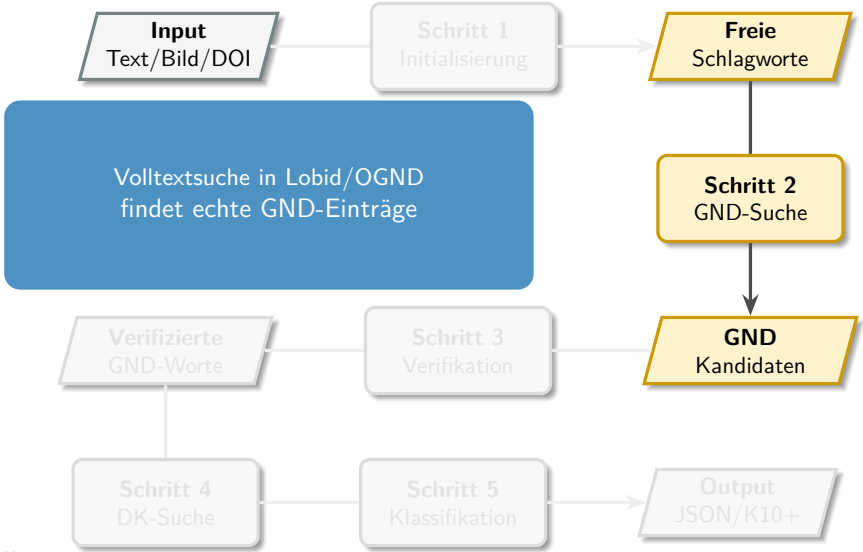
Die ALIMA-Pipeline – Schritt für Schritt



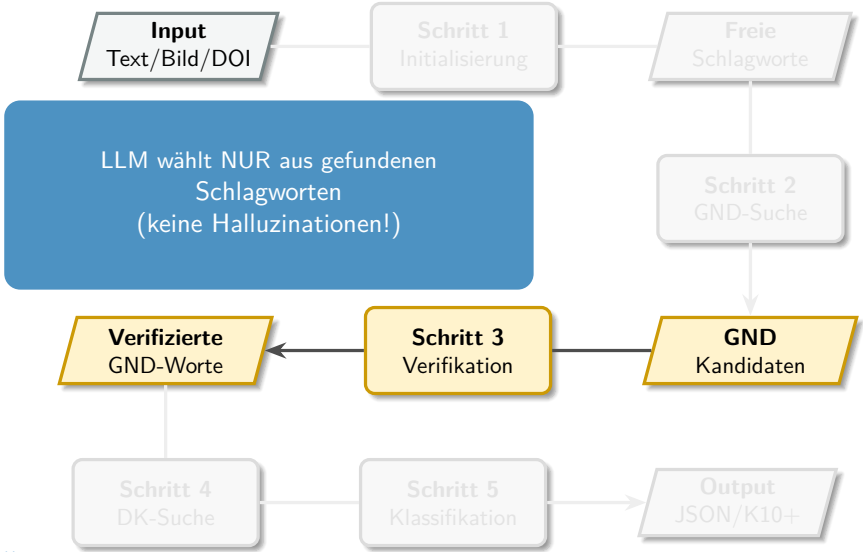
Die ALIMA-Pipeline – Schritt für Schritt



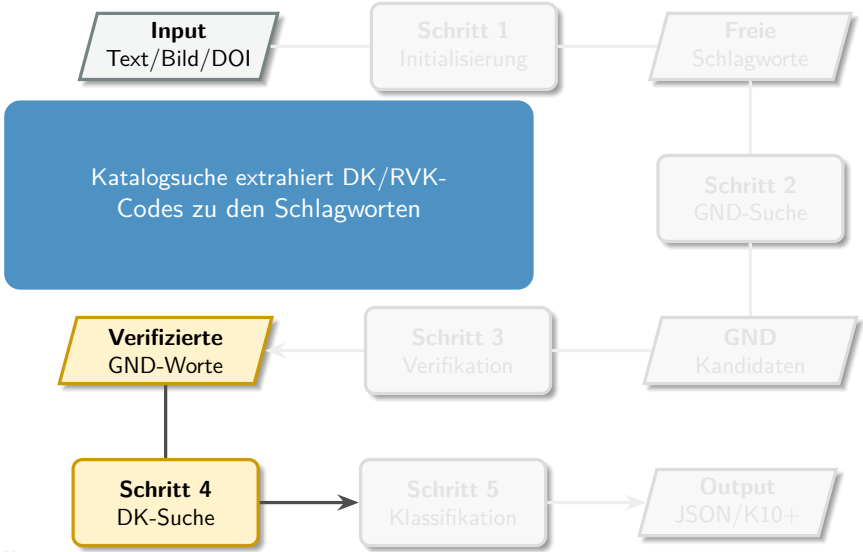
Die ALIMA-Pipeline – Schritt für Schritt



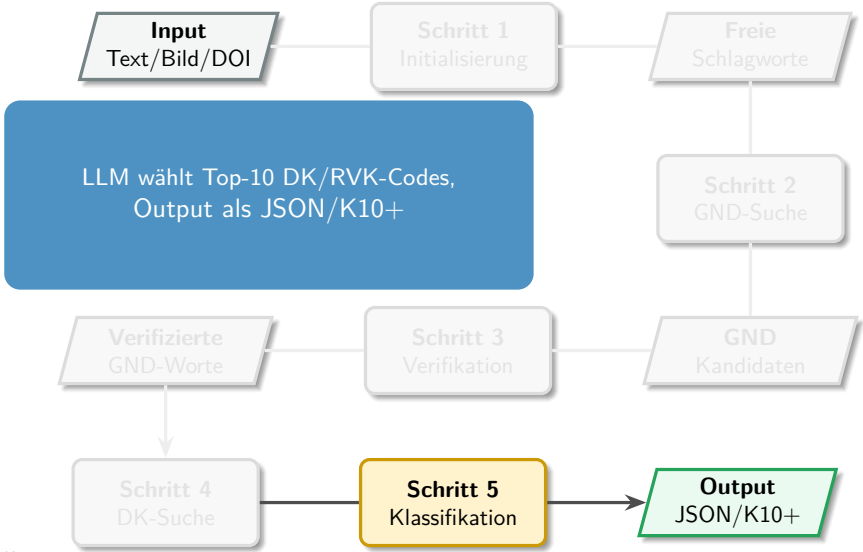
Die ALIMA-Pipeline – Schritt für Schritt



Die ALIMA-Pipeline – Schritt für Schritt



Die ALIMA-Pipeline – Schritt für Schritt



Technischer Stack und Provider

Kernkomponenten:

- **Programmiersprache:** Python 3.8+ -> LLM-Coding mit Claude/Qwen
- **GUI:** PyQt6
- **Datenquellen:** Lobid-API, OGND, lokale Sqlite3/MariaDB
- **Web:** WebApp mit FastAPI/Uvicorn

LLM-Provider:

- **Ollama** (lokale Modelle)
- **OpenAI** bzw **GWDG**
- **Google** und **Anthropic API** theoretisch nutzbar, aber nicht weiter gepflegt

Getestete Modelle:

- *Gemini 1.5/2* und *DeepSeek R1* Anfang 2025
- *Gemma 3 27B* (kostenlos via GWDG oder ollama)
- *Cogito v1 14B/32B* (Ollama) Gute Qualität für kleine Modelle
- *Nemotron Nano 3* (Ollama) Gute Qualität, aktueller Favourit

Bedienung: GUI, WebApp und CLI

Grafische Oberfläche (PyQt6):

- Pipeline-Tab mit Schritt-für-Schritt-Analyse
- Review-Tab zur Überprüfung und Änderung der Ergebnisse

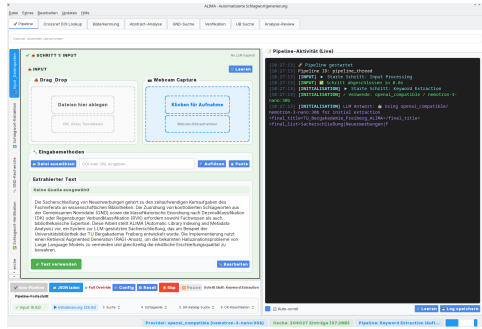
WebApp:

- Browser-basierte Verwendung
- Für Fernzugriff / Scheduler in Arbeit

Kommandozeile (CLI):

- Für Automatisierung und Scripting
- Batch-Verarbeitung ganzer Dokumenten-Listen
- Protokoll-Anzeige und Export (CSV, K10+)

- Json als Austauschformat für alle Frontends



PyQt6 Gui

Bedienung: GUI, WebApp und CLI

Grafische Oberfläche (PyQt6):

- Pipeline-Tab mit Schritt-für-Schritt-Analyse
- Review-Tab zur Überprüfung und Änderung der Ergebnisse

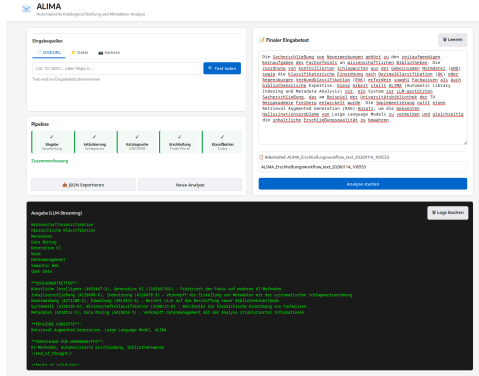
WebApp:

- Browser-basierte Verwendung
- Für Fernzugriff / Scheduler in Arbeit

Kommandozeile (CLI):

- Für Automatisierung und Scripting
- Batch-Verarbeitung ganzer Dokumenten-Listen
- Protokoll-Anzeige und Export (CSV, K10+)

- Json als Austauschformat für alle Frontends



Webapp via Python Webserver

Qualität der Ergebnisse

Aktuelle Situation (Januar 2026):

- Tool wird an der UB Freiberg genutzt
- *Sehr gute Ergebnisse* mit großen kommerziellen Modellen wie Gemini 2.x
- Auch lokale Modelle liefern zufriedenstellende Qualität

Wichtige Erkenntnisse:

- Modellgröße und Kontextfenster haben einen wichtigen Einfluss
- Kleine und freie Modelle ($< 70B$ Parameter) funktionieren für alle Schritte
- Prompt-Engineering und Kontext-Curation sind kritisch
- Ollama als LLM-Anbieter hat einen Bug bei der Reproduzierbarkeit

Mehrwert:

- Zeitersparnis durch paralleles Arbeiten
- Konsistenz: Keine manuelle Varianz mehr bei korrekter LLM-Verwendung
- Fachwissen: System nutzt verfügbares Domänenwissen aus dem Katalog

Geplante Erweiterungen und Verfügbarkeit

Weitere Entwicklung:

- Optimierung der GND-Schlagwort-zu-Klassifikation-Zuordnung
- Einfachere Batch-Verarbeitung
- Integration weiterer bibliothekarischer Datenquellen
- Iterative Verfeinerung des Ergebnis

Open Source:

- Code auf GitHub verfügbar (LGPL-3.0)
- Dokumentation und Workflow können andere Bibliotheken nutzen

Fazit: Von der KI zur Praxis

- **Problem:** Manuelle Sacherschließung ist zeitaufwendig und erfordert Spezialwissen
- **Lösung:** ALIMA – eine RAG-basierte Pipeline mit LLMs
- **Innovation:** Jeder Schritt ist transparent und verifizierbar, Halluzinationen werden minimiert
- **Praktisch:** Nutzt echte bibliothekarische Datenbanken (GND, Kataloge)
- **Flexibel:** GUI, WebApp, CLI für verschiedene Workflows
- **Offen:** Open Source, für andere Institutionen anpassbar

ALIMA zeigt: KI-Einsatz in Bibliotheken ist praktisch machbar!

Vielen Dank für die Aufmerksamkeit!

Kontakt: Conrad Hübler, TU Bergakademie Freiberg

Code: github.com/conradhuebler/ALIMA
Lizenz: LGPL-3.0