ENTWURF UND UMSETZUNG EINER GRAPHBASIERTEN MULTI-AGENTEN-ARCHITEKTUR ZUR VERBESSERTEN GENERIERUNG VON BLOGBEITRÄGEN MIT LARGE LANGUAGE MODELS UNTER NUTZUNG VON RETRIEVAL-AUGMENTED GENERATION

Bachelorarbeit von Marc Rodenbäck
Betreuer: Dipl.-Ing. Dr.techn. Marian Lux

MOTIVATION

Praktischer Zugang zu tieferem Verständnis von…

- …Large Language Model Architektur und Funktionsweise
- …Multi-Agenten-Systemen und arbeitsteiliger KI-Verarbeitung
- …Orchestrierung komplexer KI-Workflows
- …Prompt Engineering

GAP-ANALYSE

Die ursprüngliche Version zeigte sich in vielen Aspekten…

- …starr
- …ineffizient
- …fehleranfällig
- …intransparent
- …schwer bedienbar
- …schwer erweiterbar
- …schwer wartbar

GRUNDLAGEN

MODELLE & STEUERUNG

Large Language Models & Tokenization[1]

- Generierung von Sprache basierend auf probabilistischer Token-Vorhersage
- **Token-Limits:** Das Kontextfenster begrenzt die Menge an verarbeitbaren Informationen pro Durchlauf strikt

Prompt-Engineering[2]

- Die semantische Programmierung und Steuerung des LLM-Verhaltens
- **Chain-of-Thought (CoT):** Strukturierte Anweisungen zwingen das Modell zu transparenten Zwischenschritten, was Halluzinationen signifikant reduziert

WISSEN & ORCHESTRIERUNG

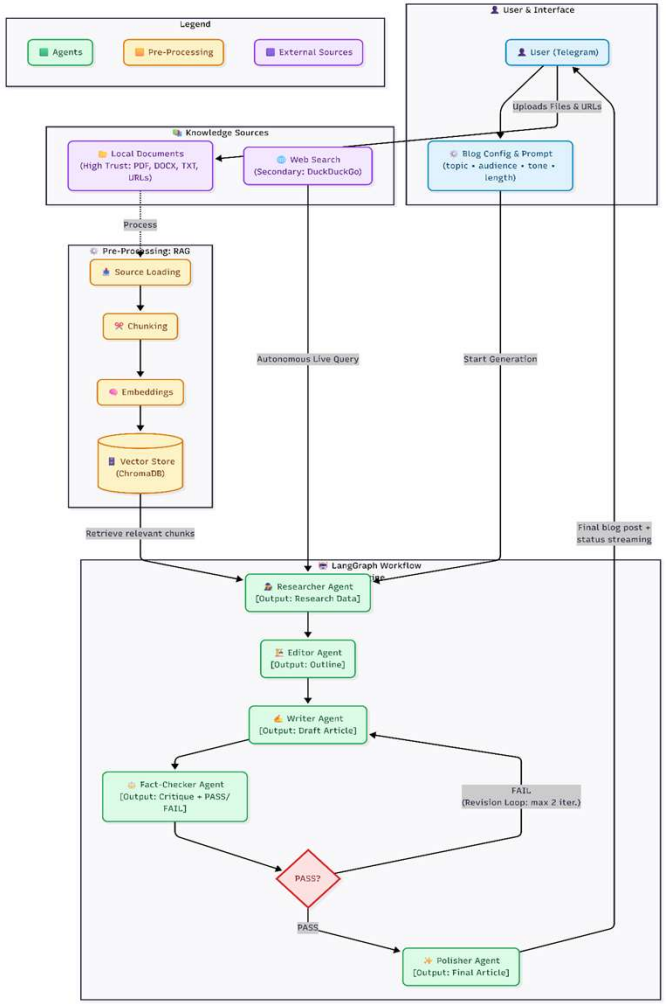
Retrieval-Augmented Generation (RAG)[3]

- Dynamische Injektion von externen, verifizierten Quellen in den LLM-Prompt
- **Ziel:** Überwindung des statischen Trainingswissens und „Erdung“ der Textgenerierung auf Faktenbasis

Multi-Agenten-Systeme (MAS) & Self-Reflection[4]

- **Orchestrierung:** Aufteilung komplexer Aufgaben in spezialisierte Agentenrollen (z.B. Researcher, Writer)
- **Self-Reflection:** Programmatische Feedback-Schleifen erlauben es dem System, eigene Outputs iterativ zu evaluieren und selbstständig zu korrigieren

ARCHITEKTUR


Legend
Agents
Pre-Processing
External Sources
User & Interface
User (Telegram)
Uploads Files & URLs
Knowledge Sources
Local Documents
(High Trust: PDF, DOCX, TXT, URLs)
Web Search
(Secondary: DuckDuckGo)
Blog Config & Prompt
(topic • audience • tone • length)
Process
Pre-Processing: RAG
Source Loading
Chunking
Embeddings
Vector Store
(ChromaDB)
Autonomous Live Query
Start Generation
Retrieve relevant chunks
Final blog post + status streaming
LangGraph Workflow
Researcher Agent
[Output: Research Data]
Editor Agent
[Output: Outline]
Writer Agent
[Output: Draft Article]
Fact-Checker Agent
[Output: Critique + PASS/FAIL]
FAIL
(Revision Loop: max 2 iter.)
PASS?
PASS
Polisher Agent
[Output: Final Article]

DEMO

EVALUIERUNG

Kriterium	Alte Version	Neue Version
Framework-Architektur	**CrewAI:** API-Zwang (Vendor Lock-in), Blackbox-Prompts & versteckter Datenfluss	**LangGraph:** LLM-agnostisch, volle Prompt-Kontrolle & explizites State Management, maximale Transparenz
Workflow & Fehlerkorrektur	Strikt linear ohne echte Revision	Zyklischer State-Graph mit aktivem Fact-Checker-Loop
KI-Modell-Strategie	“One-Size-Fits-All”(1 Modell für alle Tasks)	„Triple-Model“ (Logik: Qwen, Kreativ: Gemma, Freechat: Llama)
Usability (Telegram)	Mühsame Texteingaben & keine flexible Navigation, Output als Textnachricht	Interaktive Buttons & flüssige Navigation (Back/Restart), Output als Markdown-file
Lesbarkeit & UI-Design (Telegram)	Unstrukturierte Textblöcke & „Blackbox“-Wartezeit	Strukturierte Texte, Emojis & Live-Status-Streaming
Output-Qualität	Risiko von „Meta-Bleeding“ & Halluzinationen, oft sehr langsam (bis zu über 1h) oder Abbruch (Endlosschleife)	„Prompt Hardening“ verspricht saubere, verifizierte Texte, Ergebnis in 7-14 Minuten

FAZIT

METHODIK

- **Frontend-First-Ansatz:** hat Auswertung & Debugging stark beschleunigt
- **AI-Assisted Engineering (Gemini 3.1 Pro):** besonders hilfreich bei Prompt-Engineering & Test-Design („AI versteht AI“)
- **Developer Experience (DX):** Einsatz visueller Marker (Emojis/Highlights) hat Lesbarkeit der System-Logs optimiert
- **Iterative Auswertung:** Systematische Testreihen, transparent dokumentiert und versioniert, haben geholfen Fortschritt zu tracken (*/Test_Cases*)
- **Spätes Upgrade der Dependencies:** Unvorhergesehene API-Zwänge in neuen CrewAI-Versionen erzwangen unter Zeitdruck eine komplexe Architektur-Migration auf LangGraph

ERGEBNIS

User-Sicht

- ✓ **Maximale Usability & Transparenz:** Interaktive Buttons, flüssige Navigation und Live-Status-Streaming beenden mühsame Texteingaben und intransparenter Wartezeiten
- ✓ **Lesbarkeit & Vertrauen**: „Prompt Hardening“ eliminiert störendes Meta-Bleeding, während die aktive Faktenprüfung Halluzinationen drastisch reduziert und saubere Texte liefert

Entwickler-Sicht

- ✓ **Zukunftssicherheit & Erweiterbarkeit:** Die LLM-agnostische LangGraph-Architektur verhindert Vendor-Lock-in und lässt sich modular um neue Modelle, Agenten und Workflows erweitern
- ✓ **Kontrolle & Robustheit:** Explizites State-Management und der zyklische Fact-Checker-Loop ersetzen die fehleranfällige „Blackbox“ durch nachvollziehbare Selbstreparatur

FUTURE WORK

Backend

- **Graphen-Skalierung:**
  - Dynamisches Hinzufügen/Entfernen von Agenten inkl. passender Modelle
  - Neuausrichtung/Parallelisierung von Workflows
  - „Human-in-the-Loop“-Integration
- **Prompt-Spezialisierung:** Echte strukturelle Trennung und Verarbeitung von „Topic“ (Wissen) vs. „Task“ (Handlung)

Frontend

- **UI-Deep-Links:** Direkte Sprungmarken zur gezielten Parameter-Anpassung im Bestätigungs-Schritt.
- **Erweiterte Parameter:** Einführung neuer Metadaten (z.B. Zielgruppe, Ausgabeformat, Struktur).

LITERATUR

- [1] **Zhao, W. X. et al. (2023).** *A Survey of Large Language Models*. arXiv preprint arXiv:2303.18223.
- [2] **Wei, J. et al. (2022).** *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv preprint arXiv:2201.11903.
- [3] **Lewis, P. et al. (2020).** *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv preprint arXiv:2005.11401.
- [4] **Shinn, N. et al. (2023).** *Reflexion: Language Agents with Verbal Reinforcement Learning*. arXiv preprint arXiv:2303.11366

DANKE FÜR DIE AUFMERKSAMKEIT!