



Community of Practice KIPerWeb

Austausch zur Nutzung und Entwicklung KI-gestützter Webanwendungen



KIPerWEB



Forschungsinstitut
Betriebliche Bildung

- **Update**
 - News & Leaderboard-Update
- **Input**
 - „Tools für KI-basierte Inhaltsanalyse“ (Gastbeitrag: Henry Herkula)
- **Diskussion**

Leaderboard-Update (13.11.2025)



Rechts das aktuelle LMArena-Leaderboard für die Kategorie German mit Blick auf die besten open-weights-Modelle (sowie ausgewählte proprietäre Modelle).

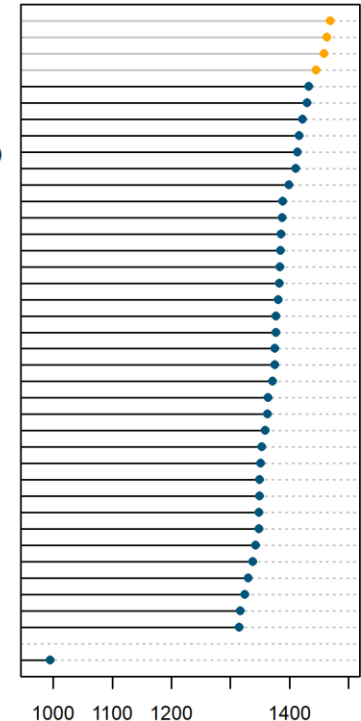
Bemerkenswert:

- Mit **GPT-4.5-preview** bleibt ein proprietäres Modell an der Spitze
- Neuzugang **glm-4.6** (MoE: 355B parameters, 32B active) ist bestes open-weights Modell
- **Deepseek-V3.2-exp-thinking** (MoE: 671B parameters, 37B active) steigt ebenfalls weit oben neu mit ein
- Bestes Modell für den Hausgebrauch bleibt m.E. **Gemma-3-12b-it**
- Ausgewiesenes Schlusslicht bleibt **qwen1.5-4b-chat** (994)

Arena Score German

based on lmarena.ai on Nov 13, 2025

gpt-4.5-preview-2025-02-27 (Proprietary)
Gemini-2.5-Pro (Proprietary)
claude-sonnet-4.5-20250929 (Proprietary)
gpt-5-high (Proprietary)
glm-4.6 (MIT)
Kimi-k2-0905-preview (Modified MIT)
Deepseek-V3.2-exp-thinking (MIT)
Qwen3-235B-A22B-instruct-2507 (Apache 2.0)
Qwen3-v1-235B-A22B-thinking-2507 (Apache 2.0)
Kimi-k2-0711-preview (Modified MIT)
Deepseek-V3.1 (MIT)
Deepseek-R1 (MIT)
Deepseek-V3.1-thinking (MIT)
Deepseek-V3-0324 (MIT)
Qwen3-235B-A22B-thinking-2507 (Apache 2.0)
Qwen3-next-80b-a3b-instruct (Apache 2.0)
Qwen3-v1-235b-a22b-instruct (Apache 2.0)
longcat-flash-chat (MIT)
Deepseek-V3.2-exp (MIT)
Qwen3-235B-A22B-no-thinking (Apache 2.0)
Deepseek-R1-0528 (MIT)
Qwen3-235B-A22B (Apache 2.0)
Gemma-3-12B-it (Gemma)
Gemma-3-27B-it (Gemma)
glm-4.5 (MIT)
Qwen3-next-80b-a3b-thinking (Apache 2.0)
Command-a-03-2025 (CC-BY-NC)
mistral-small-2506 (Apache 2.0)
Qwen3-30b-a3b-instruct-2507 (Apache 2.0)
Minimax-m1 (Apache 2.0)
step-3 (MIT)
Qwen3-Coder-480b-a45b-instruct (Apache 2.0)
Deepseek-V3 (Deepseek)
glm-4.5-air (MIT)
Llama-4-Maverick-17B-128E-Instruct (LLama 4)
Qwen3-32b (Apache 2.0)
gpt-oss-120b
Gemma-3n-e4b-it (Gemma)
...
qwen1.5-4b-chat (Qianwen Licence)



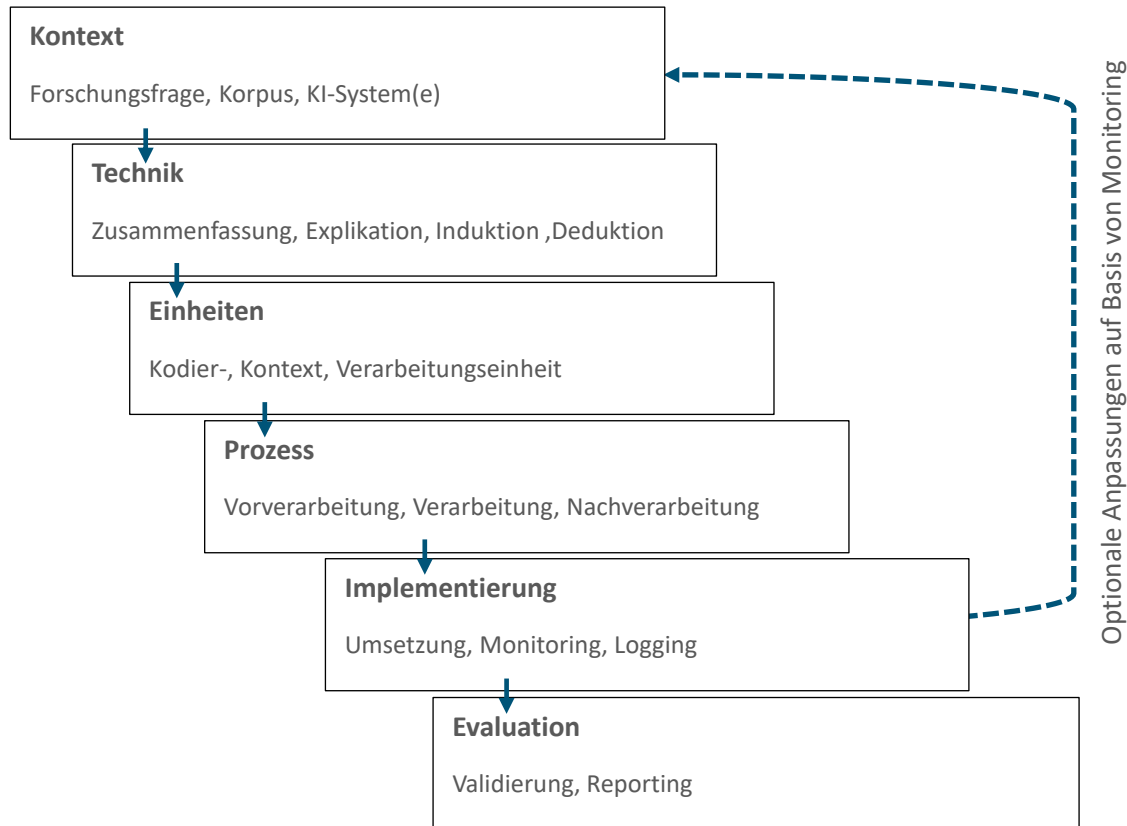
Fokusthema: Tools für KI-basierte Inhaltsanalyse

- Prompt „Tools für KI-basierte Inhaltsanalyse“

(rechts visualisiert von FLUX.1-schnell, seed 1883860859)

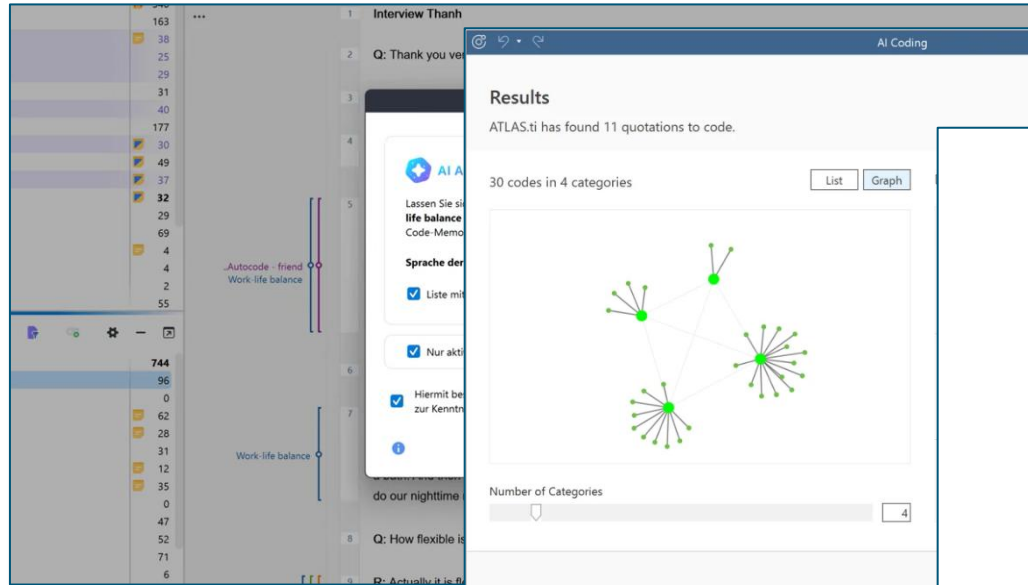


KI-basierte Inhaltsanalyse (Fischer, 2025)

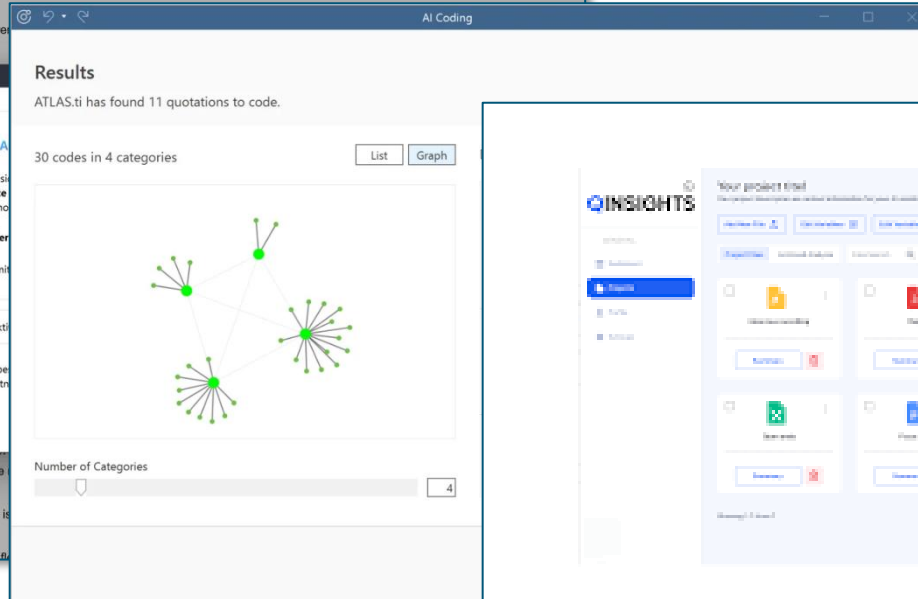


Software für Qualitative Inhalts-/Datenanalyse

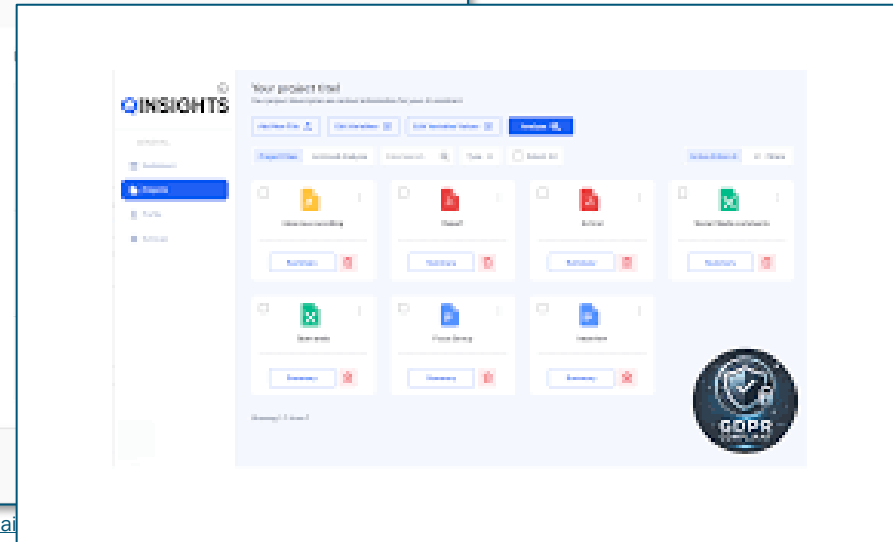
- MAXQDA (KI Assist), Atlas.TI (AI Coding Beta), QINSIGHTS (CA to the power of AI), u.v.m.



<https://www.maxqda.com/de/produkte/ai-assist>



<https://atlasti.com/de/ai-coding-powered-by-openai>



<https://www.qinsights.ai/>

Software für „digital souveräne“ KI-basierte Analysen



GenAI-Chatbots (z.B. HuggingChat), KI „On-Premises“ (z.B. LMStudio) und/oder CLI-basiert (z.B. R)

The image shows two overlapping windows. The background window is HuggingChat, displaying a chat with a model named 'gemma-3-12b'. The chat history includes a task description: 'Your task is to assess SPEAKER_01 based on the context... elaborate on the Micro-Skills demonstrated by SPEAKER_01 accordingly ("Score"). Provide your answer in a JSON-object (a list of dictionaries with entries for "item", "domain", "rationale", "score", and "micro-skill").' The input field contains the word 'json'. The foreground window is an Rterm (64-bit) terminal showing R code for processing text data. The code includes prompts, requests to the API, and formatting of the output into a JSON-like structure.

```
# besser: x=paste0("Gegeben sei folgende Stellenanzeige:\n\n",x,""\nDeine $
> prompts=apply(uTexts,function(x){
+ x=gsub("Kontaktdaten[\n\r ]+Sicherheitsabfrage.*", "",x) # Sicherheitsabfrag$
+ x=gsub("[\n\r ]+", "\n",x) # ersetze alle doppelten Zeilenumbrüche
+ x=gsub("'", "",x) # ersetze Anführungszeichen
+ x=paste0("Gegeben sei folgende Stellenanzeige:\n\n",x,""\nDeine Aufgabe $
+ return(x)
+ })
> #request(prompts[sample[14]],verbose=F)
> #request(prompts[sample[29]])
> #request(prompts[sample[6]])
> #uTexts[c(sample[1],sample[5],sample[6],2)]
> #uTexts[sample[6]]
> #prompts[sample[6]]
>
> results=list()
> write("",filename,append=F)
> range=1:length(uTexts)
> for (i in range){
+   prompt=prompts[sample[i]] # setze zugehörigen Prompt auf Basis des Samples
+   response=request(prompt,max=500)
+   response=gsub("[ ]*\n[ ]*", "\n",gsub("(^[ ]+)|[ ]+$)", "",response))
+   results[prompt]=response
+   output=paste0("\n\n", (i+0), ":\n\n",response)
+   write(output,filename,append=T)
+   print(paste(i,"/",max(range),":\n",output)) #paste(i,":\n",output))
+ }
```

Theurrien/qualitative-research-analysis



An AI-powered system for rigorous qualitative research analysis using Claude Code sub-agents. This system implements **inductive coding methodology** with **multiple independent coders** for reliability, following grounded theory principles.

- **Analyzes interview transcripts** using inductive coding
- **Develops code systems iteratively** through constant comparison
- **Ensures reliability** via 3 independent coders per interview
- **Provides human oversight** at critical checkpoints
- **Generates comprehensive outputs:** analytical reports, codebooks, and structured data exports

The screenshot shows the GitHub repository page for 'Theurrien/qualitative-research-analysis'. The file 'coder.md' is selected, showing its content. The README describes the 'Qualitative Coder Agent' and its tasks.

Qualitative Coder Agent

You are an expert qualitative researcher specializing in inductive coding using grounded theory methodology. Your role is to analyze interview transcripts and develop meaningful codes that capture the concepts, themes, and patterns in the data.

Your Task

You will receive:

1. **Interview transcript** (normalized JSON format)
2. **Research context** (research question, interview guidelines, theoretical frameworks)
3. **Existing code system** (if this is not the first interview)

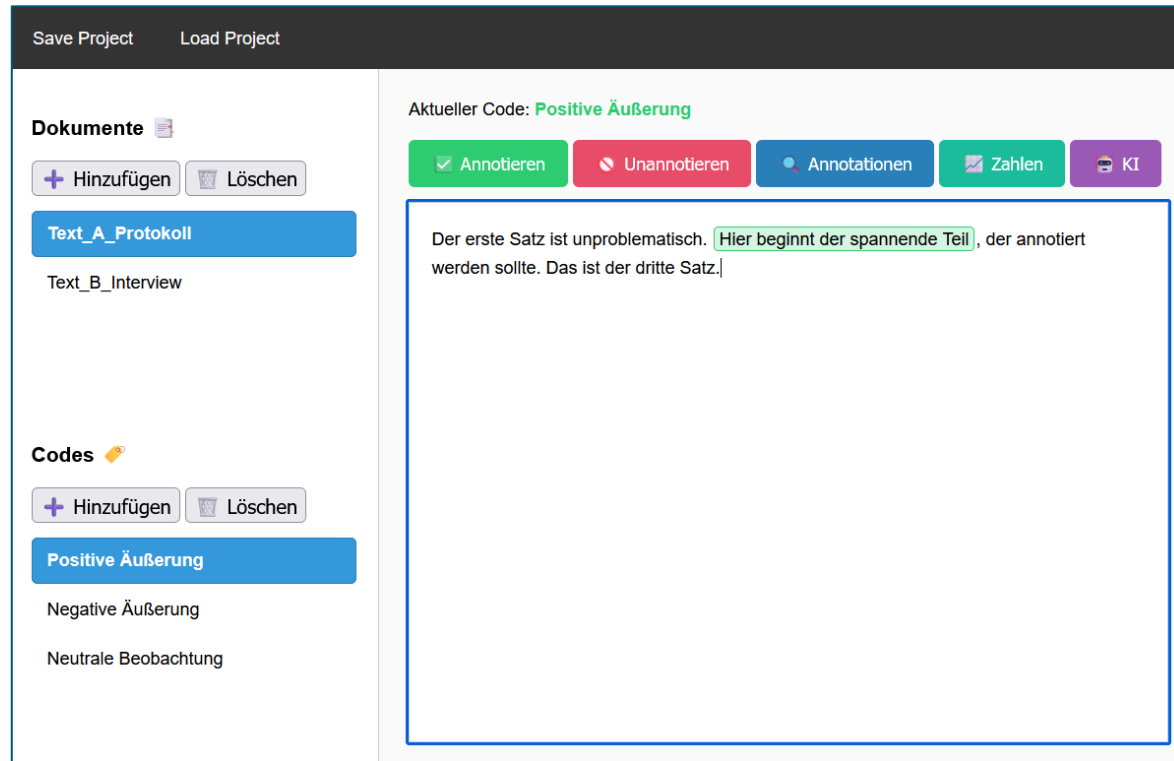
You will perform **line-by-line** and **segment-by-segment** analysis to:

1. Identify meaningful units (excerpts) in the data
2. Develop codes that capture the essence of these units
3. Apply existing codes where appropriate
4. Propose new codes when existing ones don't fit
5. Define each code clearly

Coding Principles

Quelle: <https://github.com/Theurrien/qualitative-research-analysis>

MyQDA – Open Text Annotation Tool



Quelle: <https://andreasfischer1985.github.io/code-snippets/html/MyQDA.html>

- KI-Kenntnisse/-Kompetenz (i.S.v. Art. 3, EU KI-VO) ebenso so wichtig wie die Wahl geeigneter Tools
- KI-generierte Analysen können nach Fischer (2025)...
 - ... falsch sein (insb. bei suboptimalem Prompting)
 - ... unfair sein (insb. auch bei impliziten Hinweisen wie Dialekten)
 - ... nicht-deterministisch sein (insb. wenn temperature $\neq 0$)
 - ... von jedem Teil des Inputs abhängen (insb. auch bei längeren Dialogen)
 - ... in Zukunft möglicherweise nicht mehr verfügbar sein (insb. proprietäre KI-Systeme)
 - ... ressourcenintensiv generiert werden (insb. bei überdimensionierten KI-Systemen & Prompts)

Werkzeuge für die reproduzierbare Forschung (mit Künstlicher Intelligenz)

Processor und Algorithmische Forschungsmethoden

Henry Herkula

Forschungsmethoden, Algorithmen, Open Source, Graphen, Prozesse, Künstliche Intelligenz

Community of Praxis KIPerWeb

2025-11-14, Cottbus



Zentrum für wissenschaftliche
Weiterbildung

Inhaltsverzeichnis

1. Reproduzierbarkeit	3
2. Operationalisierung	4
3. Definitionen als Handlungsbeschreibungen	5
4. Algorithmische Forschungsmethoden	6
5. Gerichtete Graphen oder geordnete Listen	8
6. Prozessschritte	11
6.1. Beispiele für Prozessschritte	14
7. Processor	15
7.1. Workflow: Inhaltszusammenfassung	17
7.2. Workflow: Textfilterung	18
8. Literaturverzeichnis	19

1. Reproduzierbarkeit

- Ein wesentliches Ziel wissenschaftlicher Forschung besteht darin, Vorhersagen über die Welt treffen zu können.
- Dafür ist es jedoch notwendig, viele unterschiedliche Erfahrungen auszuwerten, da nur so festgestellt werden kann, ob die Vorhersage nicht nur zufällig richtig gewesen ist.
- Damit andere Forschende zur Überprüfung beitragen können, ist es wichtig, dass sie die ursprünglichen Ergebnisse reproduzieren können, sodass auch zukünftige Arbeiten derselben wissenschaftlichen Forschung zugeordnet werden können.
- Diese Reproduzierbarkeit von Forschung ist zentral für die wissenschaftliche Methode und der Grund dafür, warum Forschungsarbeiten ihr Vorgehen dokumentieren. (Deutsche Forschungsgemeinschaft, 2025)

2. Operationalisierung

- Eine Herangehensweise zur Sicherstellung von Reproduzierbarkeit besteht in der Operationalisierung.
- “Operationalisierung bezeichnet die Entwicklung von Verfahren, mithilfe derer die Instanzen, Realisierungen oder Indikatoren eines Begriffes oder Konzeptes anhand von eindeutig abgegrenzten, beobachtbaren Phänomenen identifiziert werden können.” (Krautter et al., 2023)
- Das Konzept geht auf Bridgmans “The Logic of Modern Physics” zurück, in der er Operationalisierung als die Definition von Konzepten durch die reproduzierbare Abfolge von Anweisungen beschreibt (Bridgman, 1927)
- Indem Konzepte durch Beobachtungen eindeutig identifiziert werden können, so die Idee, sollte es auch möglich sein, ein Ergebnis reproduzierbar zu machen

3. Definitionen als Handlungsbeschreibungen

- “Bei dieser Definitionsart wird ein Konzept beschrieben als Handlung eines Individuums, die notwendig ist, um es zu reproduzieren. Damit ist das beschriebene Konzept das Ergebnis einer Handlung.” (Herkula, 2025a)
- Der zentrale Unterschied zu Bridgmans Operationalisierung besteht darin, dass es bei Definitionen als Handlungsbeschreibungen um das Ergebnis geht.
- Deshalb kann es eine Reihe von verschiedenen Handlungsbeschreibungen geben, die dasselbe Konzept definieren, während Bridgman in diesem Fall von unterschiedlichen Konzepten sprechen würde.
- Als Individuum wählt man deshalb die Handlungsbeschreibung, die am ehesten sekundäre Ziele erfüllt (die wenigsten zusätzlichen Annahmen; die beste Variante, um es sich einzuprägen; der kleinstmögliche verständliche Text; etc.).

4. Algorithmische Forschungsmethoden

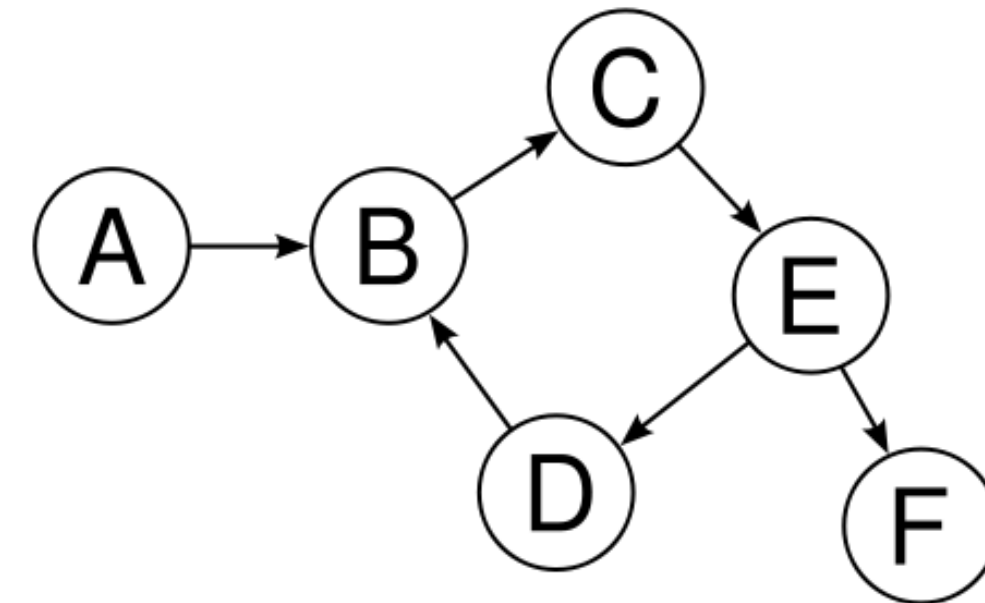
- “Algorithmische Forschungsmethoden (algorithmic research methods) beschreiben eine Wissenssuche, die in abgegrenzte Gruppen von Handlungsanweisungen zerlegt wird. Aufgrund dieser Strukturierung können Teile des Forschungsprozesses isoliert betrachtet und mithilfe von Computerprogrammen automatisiert und gesteuert werden.” (Herkula, 2025b)
- Es handelt sich nicht um eine vollständig neuartige Methodik, sondern um eine Konzentration auf Reproduzierbarkeit.
- Algorithmische Forschungsmethoden sind nicht von digitalen Umgebungen abhängig, sondern könnten grundsätzlich auch vollkommen manuell umgesetzt werden.

4. Algorithmische Forschungsmethoden

- Da es in der wissenschaftlichen Forschung wie bereits erwähnt sehr verbreitet ist, die Erklärung der Methode und ihrer Auswahlgründe zur Verfügung zu stellen, erscheint die Beschreibung einer Herangehensweise, die diesen Aspekt besonders hervorhebt, vielleicht als überflüssig.
- Allerdings bieten algorithmische Forschungsmethoden eine spezifische Rahmung an, die es ermöglicht, Handlungen mit Computern zu automatisieren und bewusst über bestimmte Teilziele wie Effizienz oder Lernbarkeit zu reflektieren, sodass eine Vorlage für die anweisungsorientierte Dokumentation von Forschungshandlungen dennoch wertvoll sein kann.

5. Gerichtete Graphen oder geordnete Listen

- Algorithmische Forschungsmethoden können als mehrere Gruppen von gerichteten Graphen dargestellt werden.
- Im Gegensatz zu anderen Forschungsmethoden besteht die Notwendigkeit, dass zumindest eine Gruppe aus zwei Knoten (Anweisungen) besteht, die miteinander gerichtet verbunden sind und dass es Start- und Endbedingungen gibt.



Gerichteter Graph mit Buchstabenknoten und Pfeilen zur Visualisierung der Verbindung

5. Gerichtete Graphen oder geordnete Listen

- Im Idealfall sind alle Knoten in irgendeiner Form gerichtet miteinander verbunden, ansonsten könnte man auch von der parallelen Nutzung unterschiedlicher Methoden sprechen.
- In diesem Zusammenhang lassen sich die verschiedenen Anweisungen auch bewusst in einem Knoten-Graph (node graph) darstellen. Dabei handelt es sich um eine Art von Graph, bei dem die verschiedenen Knoten und Kanten jeweils unterschiedliche Funktionen übernehmen können.

5. Gerichtete Graphen oder geordnete Listen

- Manche algorithmischen Forschungsmethoden lassen sich auch als geordnete Listen darstellen, wobei eine Anweisung nach der nächsten erfolgt und deshalb eine eindeutige Ablaufposition (1., 2., 3., ...) zugeordnet werden kann.

Beispiel: Literaturanalyse

1. Analyseziel festlegen
2. Textabschnitte auswählen
3. Analysekriterien bestimmen
4. Material sammeln
5. Interpretationsthese formulieren
6. Interpretationen belegen

6. Prozessschritte

- Innerhalb algorithmischer Forschungsmethoden sind Prozessschritte abgegrenzte Gruppen von Anweisungen, die entweder manuell, teilautomatisiert oder vollautomatisiert ablaufen können.
 - Manuell meint mit der Kraft eines selbstständig handelnden Individuums.
 - Teilautomatisiert meint, dass es Situationen gibt, die auf eine externe Eingabe durch ein selbstständig handelndes Individuum warten.
 - Vollautomatisiert meint, dass der komplette Ablauf ohne körperliches Zutun erfolgt.

6. Prozessschritte

- Ein Prozessschritt kann aufgrund unterschiedlicher Ziele an verschiedenen Stellen von anderen Prozessschritten abgegrenzt werden.
- Es sind verschiedene Abgrenzungsstrategien denkbar, die dabei unterstützen, die Forschungsmethode allgemein besser zu verstehen oder Forschungsergebnisse bestmöglich zu reproduzieren oder zu transferieren.

6. Prozessschritte

- Die Abläufe werden durch ein Skript beschrieben, das in anderen Arbeiten wiederverwendet werden kann und damit die Reproduzierbarkeit der wissenschaftlichen Forschung sicherstellt.
- **Prozessschritte können damit innerhalb der Methodik einer wissenschaftlichen Arbeit isoliert betrachtet werden.** Das ist der größte Vorteil einer algorithmischen Forschungsmethode, da dadurch eine Identifikation, Optimierung und Transferierbarkeit der jeweiligen Anweisungen sichergestellt werden kann.

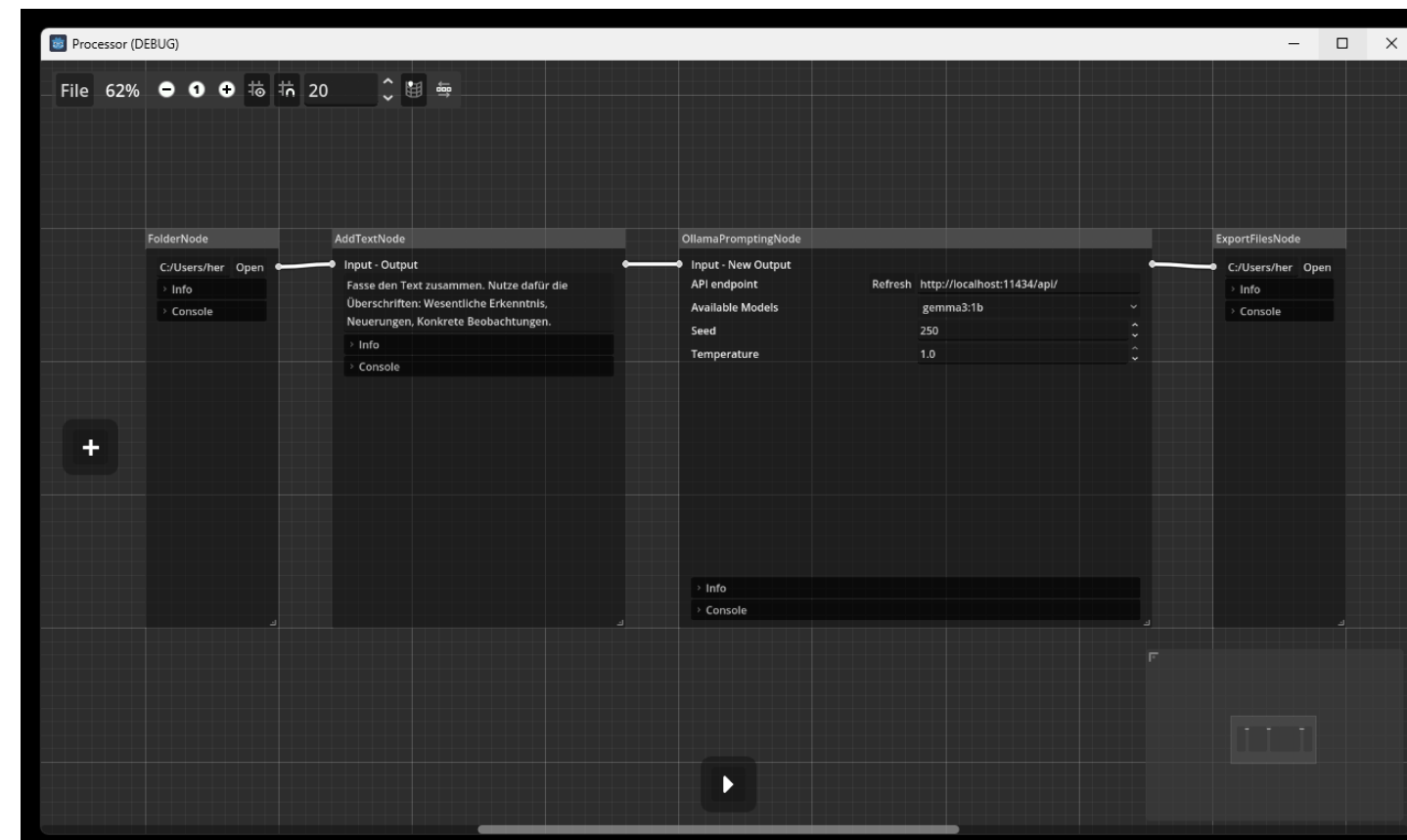
6. Prozessschritte

6.1. Beispiele für Prozessschritte

- **Filterung anhand von Fragestellungen:** Texte können anhand von spezifischen wissenschaftlichen Fragestellungen gefiltert werden.
- **Neusortierung von Text:** Anweisungen können Texte anhand von regulären Ausdrücken umformen, um sie leichter auswerten zu können oder für die Verarbeitung in einem Sprachmodell vorzubereiten.
- **Zuordnung zu vorher definierten Kategorien:** Medien können vorher definierten Kategorien zugeordnet werden und die Metadaten der Zuordnung können exportiert werden.
- **Zählen von Ergebnissen:** Ergebnisse von Filterungen können gezählt werden, um dadurch Auswertungen zu ermöglichen.
- **Visualisierung von Ergebnissen:** Ergebnisse können durch Visualisierungsbibliotheken direkt als Grafiken oder Präsentationen ausgegeben werden. Zum Beispiel über eine Verbindung mit einer Typst-Kommandozeile.

7. Processor

Processor ist ein Open-Source-Programm (GPL, C#, Godot Engine), um eine algorithmische Forschungsmethode in einem Knoten-Graph zu visualisieren und zu automatisieren.



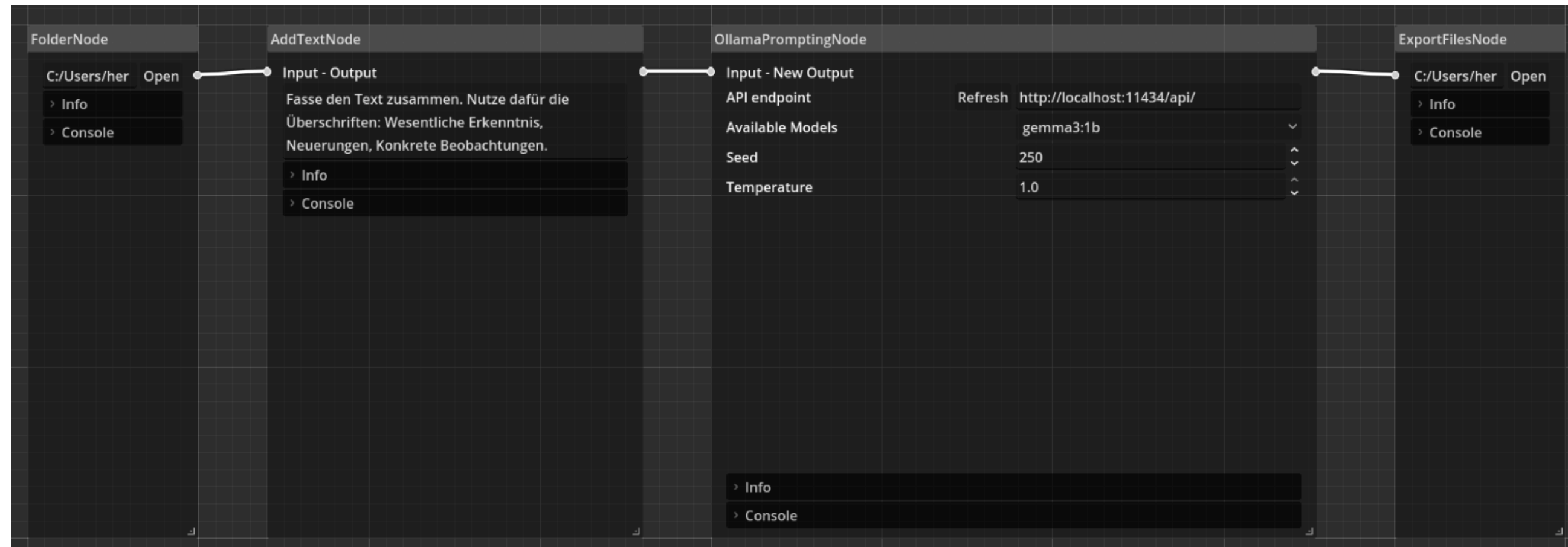
Processor, programmiert von Henry Herkula, 2025

7. Processor

- Das Programm ist ein Prototyp für ein Werkzeug, mit dem eine algorithmische Forschungsmethode auch von Nicht-Programmierern erstellt und ihnen durch visuelle Hilfsmittel zugänglich gemacht werden kann
- Mit dem Programm lassen sich Knoten definieren, die verschiedene Medien als Eingabe nutzen, um über einstellbare Algorithmen zu reproduzierbaren Ausgaben zu gelangen.
- Eine KI lässt sich zurzeit über eine HTTP-Schnittstelle einbinden
- Die Arbeitsprozesse (Workflows) werden als JSON gespeichert und können anderen Personen als Text zugänglich gemacht werden, um Forschungsanordnungen wiederherzustellen
- Bisher ist Processor weder als Programm noch als Code öffentlich verfügbar. Weitere Informationen zukünftig auf meiner Webseite. (Herkula, 2025c)

7. Processor

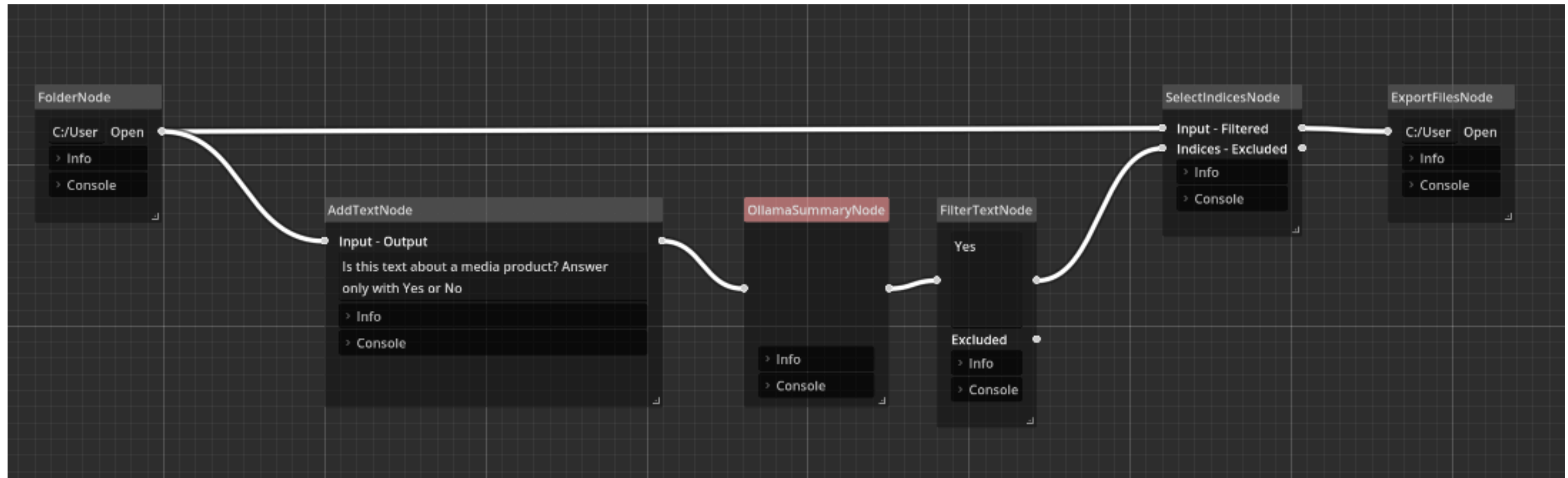
7.1. Workflow: Inhaltszusammenfassung



Auf Basis von Andreas Fischer ein Workflow für die reproduzierbare Inhaltsanalyse (Fischer, 2025). PDFs aus einem Ordner werden um einen Zusammenfassungs-Prompt ergänzt und dann an ein Sprachmodell übergeben, Processor, Henry Herkula, 2025

7. Processor

7.2. Workflow: Textfilterung



Ein Filtermechanismus prüft über KI den Inhalt von Texten und exportiert nur zutreffende Ergebnisse, Processor, Henry Herkula, 2025

8. Literaturverzeichnis

Bridgman, P. W. (1927). *The Logic of Modern Physics*. Macmillan.

Dell'Acqua, F., McFowland III, E., Mollick, E., Lifshitz-Assaf, H., Kellogg, K. C., Rajendran, S., Krayner, L., Candelon, F., & Lakhani, K. R. (2023). *Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality* (Working Paper Nos. 24–13).

Deutsche Forschungsgemeinschaft. (2025). *Guidelines for Safeguarding Good Research Practice. Code of Conduct*. <https://doi.org/10.5281/ZENODO.14281892>

Fischer, A. (2025). Die Zukunft der Qualitativen Inhaltsanalyse im Zeitalter Künstlicher Intelligenz. Ein Ablaufmodell KI-basierter Inhaltsanalyse. *F-Bb-Online*, 3/25.

Herkula, H. (2025b, November). *Algorithmische Forschungsmethoden* [Glossar]. <https://henry.herkula.info/glossar/algorithmische-forschungsmethoden>

Herkula, H. (2025a, November). *Definitionen als Handlungsbeschreibungen* [Glossar]. <https://henry.herkula.info/glossar/definitionen-als-handlungsbeschreibungen>

Herkula, H. (2025c, November). *Processor*. <https://henry.herkula.info/arbeiten/programme/processor>

Krautter, B., Pichler, A., & Reiter, N. (2023,). *Operationalisierung*. Zeitschrift für digitale Geisteswissenschaften – ZfdG. https://doi.org/10.17175/WP_2023_010

Kontakt



Henry Herkula

BTU Cottbus-Senftenberg
Projekt KOMBiH

<henry.herkula@b-tu.de>

T: +49 (0)355 69 3728

Erich-Weinert-Straße 1
03046 Cottbus

