
Projektbericht: IMARA

Domain-specific GraphRAG pipeline with model fine-tuning

Modul: Abschlussarbeit CAS Machine Learning for Software Engineers (ML4SE)

Datum: [Aktuelles Datum]

Autoren: Marco Allenspach, Lukas Koller, Emanuel Sovrano

Abstract

Die Einführung von Retrieval-Augmented Generation (RAG) markierte einen bedeutenden Meilenstein in der Anwendung grosser Sprachmodelle (LLM), indem generative Fähigkeiten auf faktischen, externen Daten basierten, um Fehlinterpretationen zu vermeiden und die Relevanz zu erhöhen. Um die Schwächen von RAG der ersten Generation durch die Einführung strukturierter, relationaler Kontexte zu beheben, hat sich jedoch mit AI-Native GraphRAG ein weiterentwickeltes Paradigma etabliert.

Der rasante branchenweite Wandel hin zu graphenbasierten Architekturen ist eine notwendige Weiterentwicklung, die auf der Erkenntnis beruht, dass eine KI für effektives Denken ein Modell des Anwendungsbereichs benötigt, nicht nur eine Sammlung von Fakten. Der Fortschritt von unreflektierten LLMs zu grundlegenden RAGs löste das Problem der faktischen Fundierung, doch das Versagen rein vektorbasierter RAGs bei komplexen Anfragen zeigte, dass die Struktur des Wissens ebenso wichtig ist wie sein Inhalt. Ein Wissensgraph liefert diese Struktur und transformiert eine passive Dokumentensammlung in ein aktives, abfragefähiges Modell der Welt.

1. Management Summary

(Ca. 0.5 - 1 Seite) Zusammenfassung des gesamten Projekts: Problemstellung (Extraktion aus komplexen PDFs), gewählter Lösungsansatz (GraphRAG & Fine-tuning) und die wichtigsten Ergebnisse des Benchmarkings.

Traditionelle neuronale Netze eignen sich gut zur Kodierung linearer Beziehungen, doch Daten aus der realen Welt sind in der Regel komplex und multidimensional. Graphen sind besser geeignet, höherdimensionale Verbindungen darzustellen, in denen jeder Knoten mit jedem anderen Knoten in Beziehung steht. Dadurch eignen sich Graphen besser zur Speicherung komplexer Beziehungen aus der realen Welt.

2. Einleitung und Zielsetzung

Das IMARA-Projekt hat zum Ziel, aufzuzeigen wie die Genauigkeit der Abfrage eines graph-basierten RAG-Systems sich verbessert.

Um eine Grundlage für die Messbarkeit zu haben, wurde OpenRAGBench als Referenzdatensatz ausgewählt.

Defining the "AI-Native" GraphRAG Paradigm

AI-Native GraphRAG represents a specific and powerful subset of graph-based RAG systems. Solutions must automate the entire workflow from unstructured data to a natural language answer, abstracting complexities of graph theory and database management.

naives RAG

Die inhärenten Einschränkungen von vektorbasierter RAG

Konventionelle RAG-Architekturen verlassen sich auf Vektorsimilaritätssuche über ein Korpus von geteiltem Text. Dieser Ansatz behandelt Wissen als eine Sammlung von unzusammenhängenden Fakten und hat Schwierigkeiten mit Fragen, die erfordern:

- Synthese von Informationen aus mehreren Quellen
- Verständnis nuancierter Beziehungen zwischen Entitäten
- Durchführung von Multi-Hop-Reasoning Der Kontext, der dem LLM bereitgestellt wird, ist oft eine Liste von Textausschnitten, die keine explizite Darstellung ihrer Verbindungen enthalten.

Kontextuelle Fragmentierung und Blindheit

Das Chunking bricht den natürlichen Informationsfluss willkürlich. Relevanter Kontext kann über verschiedene Chunks, Dokumente oder Abschnitte verstreut sein. Die Vektorschre, die die Anfrage mit jedem Chunk einzeln vergleicht, versagt oft dabei, diesen vollständigen, verteilten Kontext abzurufen, was zu unvollständigen oder oberflächlichen Antworten führt. Sie versteht semantische Ähnlichkeit, ist jedoch blind für explizite Beziehungen wie Kausalität, Abhängigkeit oder Hierarchie.

Empfindlichkeit gegenüber der Chunking-Strategie

Die Leistung ist hochgradig empfindlich gegenüber der Chunking-Strategie (z.B. Chunk-Grösse, Überlappung). Suboptimale Strategien können übermässiges Rauschen einführen (Chunks zu gross) oder kritischen Kontext verlieren (Chunks zu klein), was umfangreiche und brüchige Anpassungen erfordert.

Unfähigkeit, Multi-Hop-Reasoning durchzuführen

Es gibt Schwierigkeiten, komplexe Fragen zu beantworten, die "Multi-Hop"-Reasoning erfordern. Zum Beispiel: "Welche Marketingkampagnen wurden von der in dem Q3-Bericht erwähnten Lieferkettenstörung betroffen?" erfordert die Verknüpfung von Störung → betroffene Produkte → Marketingkampagnen. Eine einfache Vektorschre ist unwahrscheinlich, diese Informationssprünge zu überbrücken.

Analogie: Vektorbasierte RAG bietet einem Forscher einen Stapel isolierter Karteikarten, während GraphRAG darauf abzielt, eine umfassende Mindmap zu erstellen und bereitzustellen, die entscheidende Verbindungen aufdeckt.

2.1 Projekttitel: IMARA

2.2 Problemstellung

Die Extraktion und Verarbeitung von Informationen aus unstrukturierten PDF-Dokumenten stellt eine Herausforderung für herkömmliche RAG-Systeme dar.

2.3 Projektziele

- Die Implementation von graphbasierten System und der Vergleich zu klassischen RAG-Systemen
- Der Vergleich zwischen verschiedenen graphbasierten RAG-Systemen
-
-

Graph-basiertes RAG: Aufbau einer Pipeline zur Erstellung dichter Wissensgraphen.

•

Model Fine-tuning: Optimierung eines LLMs (z.B. Qwen) basierend auf dem Graph.

•

Automation: End-to-End Automatisierung der Pipeline. Eine flexible Pipeline bauen, die bei der Evaluation der verschiedenen RAG-Systeme unterstützt.

3. Datenbasis und Vorverarbeitung

3.1 Datenquellen

Beschreibung der verwendeten Datensätze, wie z.B. der **Open RAG Bench Dataset** (Arxiv-Kategorien) oder **PubMedQA**.

3.2 PDF-Extraktion mit Docling

Einsatz des **Docling Toolkits** zur effizienten Konvertierung von Dokumenten in maschinenlesbare Formate (Markdown/JSON).

angetroffene Herausforderungen **Challenge:** Die Qualität der Ergebnisse liegt unter den Erwartungen.

Massnahme 1: Optimierung der Parameter. Die optimierte Version der Parameter ist massiv schneller und viel genauer.

```

SUM
310  ## 6 CONCLUSION
311
312  In this paper, we presented the DocLayNet dataset. It provides the document conversion and layout anal
313
314  From the dataset, we have derived on the one hand reference metrics for human performance on document
315
316  To date, there is still a significant gap between human and ML accuracy on the layout interpretation t
317
318  ## REFERENCES
319
320  + [1] Max Gobel, Tamir Hassan, Emanuele Oro, and Giorgio Orsi. Icdar 2013 table competition. In 2013 I
321  + [2] Christian Clauener, Apostolos Antonacopoulou, and Stefan Fleetzschacher. Icdar 2017 table-based compar
322  - [3] Regino Oten (ICDAR), Juan L. Muletin, and Eric G. Roan. Icaros 2017 data-based comparison of doc
323  + [4] Antonio Jiménez, Peter Song, and Douglas S. Hurni. Scientific literature in the International Co
324  + [5] Antonio Jiménez, Peter Song, and Douglas S. Hurni. Scientific literature in the International Co
325  + [6] Xu Jian, Feng Ding, and Xian Li. Data-Sharpening for Image Annotation and Document Analysis. In 2017
326  + [7] Xiao X. Zhang, and Jian Li. Data-Sharpening for Image Annotation and Document Analysis. In 2017
327
328  + [8] Ross B. Girshick, Andrew Tompson, Trevor Darrell, J. Hindradt Technical Journal, Rich feature hierarc
329  + [9] Ross B. Girshick, Fast R-CNN, J. Hindradt Technical Journal, International Conference on Com
330  + [10] Shaqiang Ren, Ross B. Girshick, and J. Hindradt Technical Journal. Fast-recon-towards-the-world
331  + [11] Kaiming He, Georgia Gkioxari, Piotr Dollar, and J. Hindradt Technical Journal. IEEE Internati
332
333
334  Figure 6: Example of a large human-annotated dataset for document-layout analysis. (A) The dataset co
335
336  <!-- image -->
337
338  + [12] Mai Thanh Minh, Marc, albinvki, fatih, oleg, and wanghao yang. Ultralytics/yolov5: v6.0 - Yo
339
340
341  + [13] Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier,
342  + [14] Nicolas, Serebryak, and Sergey Zagoruyko. End-to-end object detection with transformers.
343  + [15] Mingxing Tan, Jianguo Peng, and Quoc V. Le. EfficientDet: Scalable and efficient object det
344  + [16] Taung-Yi Lin, Michael Maire, and Peter Pichler, de. Jean de Lavizier, Paul D. Dolart, and C. Ro
345  + [17] Yukinao, James, Hui, Pichler, and Ross Law. (2016). Cross-domain object detection using a large
346  + [18] Yuheng Xu, Minghao Liu, Lei Li, and Fei Wu. (2016). Pre-training of text and layout for docu
347  + [19] Yuheng Xu, Minghao Liu, Lei Li, and Fei Wu. (2016). Pre-training of text and layout for docu
348  + [20] Zheng, X., and Liu, X. (2016). Pre-training of text and layout for document image understanding.
349  + [21] Zheng, X., and Liu, X. (2016). Pre-training of text and layout for document image understanding.
350  + [22] Shoubin Li, Xuyan Ma, Shuaigun Pan, Jun Hu, Lin Shi, and Qing Wang. Vrlayout: Fusion of visual
351  + [23] Peng Zhang, Can Li, Liang Qiao, and Zhi Cheng, Shiliang Pu, Yi Niu, and Fei Wu. Vsr: A unified
352  + [24] Peter W J Staats, Michele Dolfi, Christoph Auer, and Costas Bekas. Corpus conversion service: A
353
354  + [25] Connor Shorten and Taghi M. Khoshgoftaar. A Survey on image data augmentation for deep l

```

```

SUM
310  ## 6 CONCLUSION
311
312  In this paper, we presented the DocLayNet dataset. It provides the document conversion and layout anal
313
314  From the dataset, we have derived on the one hand reference metrics for human performance on document
315
316  To date, there is still a significant gap between human and ML accuracy on the layout interpretation t
317
318  ## REFERENCES
319
320  + [1] Max Gobel, Tamir Hassan, Emanuele Oro, and Giorgio Orsi. Icdar 2013 table competition. In 2013 I
321  + [2] Christian Clauener, Apostolos Antonacopoulou, and Stefan Fleetzschacher. Icdar 2017 competition on
322  + [3] Hervé Delajet, Jean-Luc Meunier, Liangcai Gao, Yilin Huang, Yu Fang, Florian Kleber, and Eva-Mari
323  + [4] Antonio Jiménez, Peter Song, and Douglas Burdick. Competition on scientific literature par
324  + [5] Antonio Jiménez, Peter Song, and Douglas Burdick. Competition on scientific literature par
325  + [6] Xu Jian, Feng Ding, and Xian Li. Data-Sharpening for Image Annotation and Document Analysis. In 2017
326  + [7] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A b
327  + [8] Riaz Ahmad, Muhammad Tanvir Afzal, and M. Qadir. Information extraction from pdf sources based o
328  + [9] Ross B. Girshick, Fast R-CNN, J. Hindradt Technical Journal, International Conference on Computer Visi
329  + [10] Shaqiang Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object d
330  + [11] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross B. Girshick. Mask R-CNN, In IEEE Internati
331  + [12] Glenn Jocher, Alex Stooke, Ayush Chaurasia, Jitendra Boorevci, NanoCode12, TaoXie, Yonghui Xiong, K
332
333
334  Figure 6: Example of a large human-annotated dataset for document-layout analysis. (A, D) exhibit fav
335
336  <!-- image -->
337
338  + [12] Diazoom, Mai Thanh Minh, Marc, albinvki, fatih, oleg, and wanghao yang. Ultralytics/yolov5: v6.0 - Yo
339
340
341  + [13] Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Ser
342  + [14] Mingxing Tan, Jianguo Peng, and Quoc V. Le. EfficientDet: Scalable and efficient object detecti
343  + [15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hay
344  + [16] Nicolae Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Ser
345  + [17] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detection2, 2019.
346  + [18] Nikita Radivilov, Cesat Barisip, Mahya Lysak, Viktor Korotchiyanik, Ahmed Wassar, Andrii Tarv
347  + [19] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutnet Pre-training
348
349
350  + [20] Shoubin Li, Xuyan Ma, Shuaigun Pan, Jun Hu, Lin Shi, and Qing Wang. Vrlayout: Fusion of visual
351  + [21] Peng Zhang, Can Li, Liang Qiao, Shanzhuan Cheng, Shiliang Pu, Yi Niu, and Fei Wu. Vsr: A unified
352  + [22] Peter W J Staats, Michele Dolfi, Christoph Auer, and Costas Bekas. Corpus conversion service: A
353
354  + [23] Connor Shorten and Taghi M. Khoshgoftaar. A Survey on image data augmentation for deep learning

```

Die Unterschiede sind z.T. ganze Tabellen.

90	*We did not control the document selection with regard to language. The vast majority of documents cont
91	*To ensure that future benchmarks in the document-layout analysis can be easily compared, we split up D
92	93
94	*e.g. AAPL from https://www.bloomberg.com/
95	*Table 1 shows the overall frequency of documents among the different sets. We ensure that subsets are
96	97
98	*In order to accommodate the different types of models currently in use by the community, we provide Doc
99	100
101	*Despite being cost-inducing and far less scalable than other languages, human annotation has several b
102	103
104	The annotation campaign was carried out in four phases. In phase one, we identified and prepared the d
105	106
107	108
109	110
111	112
113	114
115	116
117	118
119	120
121	122
123	124
125	126
127	128
129	130
131	132
133	134
135	136
137	138
139	the textual content of an element, which goes beyond visual layout recognition, in particular outside
140	*At first sight, the task of visual document-layout interpretation appears intuitive enough to obtain p
141	142
143	*Obviously, this is an issue with plausible annotations. For example, examples of plausible but inconsi
144	145
146	* (1) Every list-item is an individual object with class <code>label List-item</code> . This definition is differen
147	* (2) A list-item is a paragraph with handing indentation. Single elements can be manipulated as List-
	- (3) For every <code>Caption</code> , there must be exactly one corresponding <code>Picture</code> or <code>Table</code> .
	- (4) Connected sub-pictures are grouped together in one <code>Picture</code> object.

90	*We did not control the document selection with regard to language. The vast majority of documents cont
91	*To ensure that future benchmarks in the document-layout analysis community can be easily compared, we
92	93
94	*e.g. AAPL from https://www.bloomberg.com/
95	*Table 1 shows the overall frequency and distribution of the labels among the different sets. Important
96	97
98	*In order to accommodate the different types of models currently in use by the community, we provide Do
99	100
101	*Despite being cost-intense and far less scalable than automation, human annotation has several benefit
102	103
104	105
105	*The annotation campaign was carried out in four phases. In phase one, we identified and prepared the d
106	107
107	108
108	109
109	110
110	111
111	112
112	113
113	114
114	115
115	116
116	117
117	118
118	119
119	120
120	121
121	122
122	123
123	124
124	125
125	126
126	127
127	128
128	129
129	130
130	131
131	132
132	133
133	134
134	135
135	136
136	137
137	138
138	the textual content of an element, which goes beyond visual layout recognition, in particular outside
139	*At first sight, the task of visual document-layout interpretation appears intuitive enough to obtain p
140	141
141	*Obviously, this is an issue with plausible annotations. For example, examples of plausible but inconsi
142	143
143	* (1) Every list-item is an individual object with class <code>label List-item</code> . This definition is differen
144	* (2) A list-item is a paragraph with handing indentation. Single elements can be manipulated as List-
145	- (3) For every <code>Caption</code> , there must be exactly one corresponding <code>Picture</code> or <code>Table</code> .
146	- (4) Connected sub-pictures are grouped together in one <code>Picture</code> object.
147	148

Problematische Parameter:

```

226     params = {
227         "pipeline": "vlm",
228         "from_formats": ["docx", "pptx", "html", "image", "pdf", "asciidoc", "md", "xlsx"],
229         "to_formats": ["md", "json", "html", "text", "doctags"], # Option "html_split_page"
230         "image_export_mode": "placeholder", # Allowed values: "placeholder", "embedded", "referenced". Optional, defaults to em
231         "do_ocr": True,
232         "force_ocr": False,
233         "ocr_engine": "easyocr",
234         "ocr_lang": ["en"], # en, fr, de, es
235         "pdf_backend": "dlparse_v4",
236         "table_mode": "accurate",
237         "abort_on_error": False,
238
239         "do_table_structure": True, # default is True
240         "include_images": True, # default is True
241         # "do_code_enrichment": True, # default is False
242         # "do_formula_enrichment": True, # default is False
243         # "do_picture_classification": True, # default is False
244         "do_picture_description": True, # default is False
245         "picture_description_api": None, #"http://localhost:11435/v1/",
246         #"vlm\_pipeline\_model": "granite3.2-vision:2b",
247         #"vlm\_pipeline\_model\_api": "http://localhost:11434/v1/chat/completions", # vlm_pipeline_model_api,
248

```

erfolgreiche Parameter:

```

73     parameters = {
74         "from_formats": ["docx", "pptx", "html", "image", "pdf", "asciidoc", "md", "xlsx"],
75         "to_formats": ["md", "json", "html", "text", "doctags"], # Option "html_split_page"
76         "image_export_mode": "placeholder", # Allowed values: placeholder, embedded, referenced. Optional, defaults to embedded.
77         "do_ocr": True,
78         "force_ocr": False,
79         "ocr_engine": "easyocr",
80         "ocr_lang": ["en"],
81         "pdf_backend": "dlparse_v4",
82         "table_mode": "accurate",
83         "abort_on_error": False,
84         # "do_table_structure": True, # default is True
85         # "include_images": True, # default is True
86         # "do_code_enrichment": True, # default is False
87         # "do_formula_enrichment": True, # default is False
88         # "do_picture_classification": True, # default is False
89         # "do_picture_description": True, # default is False
90         # "picture_description_api": "http://localhost:11434/v1/chat/completions",
91         # "vlm_pipeline_model": "granite3.2-vision:2b",
92         # "vlm_pipeline_model_api": vlm_pipeline_model_api,
93
94     }
95     # "target": "zip",

```

Challenge: Die 16GB VRAM waren nicht genug, um alle features von docling zu unterstützen. Das verursachte periodische Endless-loop's in Docling serve. **Massnahme 1:** Der Verzicht auf die Container-Version "Docling

serve" und die Verwendung direkt in Python. **Massnahme 2:** Die Ausführung von Docling auf der CPU, um das VRAM-Limit zu umgehen

Challenge: Die 16GB VRAM waren nicht genug, um alle features von docling zu unterstützen. Das verursachte periodische Endless-loop's in Docling serve. Die cloudcode_cli.exe in der VSCode-Umgebung hat durch einen extremen RAM-Verbrauch im Hintergrund die Ausführung von docling verhindert. freeze, not started, ... <https://forum.cursor.com/t/high-memory-consumption-on-cloudcode-cli/106122> **Massnahme 1:** Ein Uninstall von cloudcode_cli.exe war unumgänglich.

Challenge: Das parsen von Formeln in Docling mit CPU oder GPU ist sehr langsam. Den Verzicht auf die Extraktion der Formeln war keine Option, da eine maximale Qualität des Extrakts abgestrebt wurde, um die over-all Performance nicht zu beeinträchtigen.

Docling Log Ausschnitt:

```
[WindowsPath('C:/Users/ML4SE/Desktop/openspec_demo/configs/data/OpenRAGBench/pdfs/2411.02951v2.pdf')]  
2025-12-17 19:08:35,249 - INFO - detected formats: [<InputFormat.PDF: 'pdf'>]  
2025-12-17 19:08:35,259 - INFO - Going to convert document batch...  
2025-12-17 19:08:35,260 - INFO - Processing document 2411.02951v2.pdf  
2025-12-18 01:37:07,514 - INFO - Finished converting document 2411.02951v2.pdf in 23312.29 sec.  
mpve the source file to the target directory  
2025-12-18 01:37:07,940 - INFO - Processed 1 docs, of which 0 failed and 0 were partially converted.  
2025-12-18 01:37:07,948 - INFO - Document conversion complete in 203589.20 seconds. it successfully completed 1 out of 287  
[WindowsPath('C:/Users/ML4SE/Desktop/openspec_demo/configs/data/OpenRAGBench/pdfs/2411.03001v2.pdf')]  
2025-12-18 01:37:07,968 - INFO - detected formats: [<InputFormat.PDF: 'pdf'>]  
2025-12-18 01:37:07,972 - INFO - Going to convert document batch...  
2025-12-18 01:37:07,973 - INFO - Processing document 2411.03001v2.pdf  
2025-12-18 14:22:26,866 - INFO - Finished converting document 2411.03001v2.pdf in 45918.92 sec.  
mpve the source file to the target directory  
2025-12-18 14:22:27,152 - INFO - Processed 1 docs, of which 0 failed and 0 were partially converted.  
2025-12-18 14:22:27,160 - INFO - Document conversion complete in 249508.41 seconds. it successfully completed 1 out of 286  
[WindowsPath('C:/Users/ML4SE/Desktop/openspec_demo/configs/data/OpenRAGBench/pdfs/2411.03166v3.pdf')]  
2025-12-18 14:22:27,193 - INFO - detected formats: [<InputFormat.PDF: 'pdf'>]  
2025-12-18 14:22:27,201 - INFO - Going to convert document batch...  
2025-12-18 14:22:27,202 - INFO - Processing document 2411.03166v3.pdf  
2025-12-19 03:50:46,515 - INFO - Finished converting document 2411.03166v3.pdf in 48499.35 sec.  
mpve the source file to the target directory  
2025-12-19 03:50:47,201 - INFO - Processed 1 docs, of which 0 failed and 0 were partially converted.  
2025-12-19 03:50:47,229 - INFO - Document conversion complete in 298008.48
```

```

seconds. it successfully completed 1 out of 285
[WindowsPath('C:/Users/ML4SE/Desktop/openspec_demo/configs/data/OpenRAGBench/pdfs/
2411.03257v3.pdf')]
2025-12-19 03:50:47,249 - INFO - detected formats: [<InputFormat.PDF: 'pdf'>]
2025-12-19 03:50:47,257 - INFO - Going to convert document batch...
2025-12-19 03:50:47,259 - INFO - Processing document 2411.03257v3.pdf
2025-12-19 23:49:15,094 - INFO - Finished converting document 2411.03257v3.pdf in
71907.86 sec.
mpve the source file to the target directory
2025-12-19 23:49:17,939 - INFO - Processed 1 docs, of which 0 failed and 0 were
partially converted.
2025-12-19 23:49:18,034 - INFO - Document conversion complete in 369919.29
seconds. it successfully completed 1 out of 284

```

Massnahme 1: Einen zweiten Rechner 100% dafür einsetzen.

4. Methodik und Architektur

4.1 Graph-Konstruktion

4.1.1 LeanRAG Ansatz

Detaillierung der Triple-Extraktion und der hierarchischen Retrieval-Struktur.

- **Extraktion:** Umwandlung von Text in Entitäten und Relationen.
-

Aggregation: Semantische Aggregation zur Reduzierung von Redundanz.

leanRAG Workflow

file_chunk.py

1. chunk raw input token-based with 512 Tokens and 64 Tokens overlap

Method 1: CommonKG

CommonKG/create_kg.py 2. create a list of match words (entities) for each chunk 3. create a list of "all entities" based on the match words without duplicates

4. "new triples" have "subject, predicate, object" triples init with corresponding reference to the chunk of origin
5. "next layer entities"
6. "new triples descriptions"

CommonKG/deal_triple.py 7. summarize descriptions => relation.jsonl

Method 2: GraphRAG

GraphExtraction/chunk.py 2. loads the chunks 3. performs a "triple extraction" => entity.jsonl, relation.jsonl

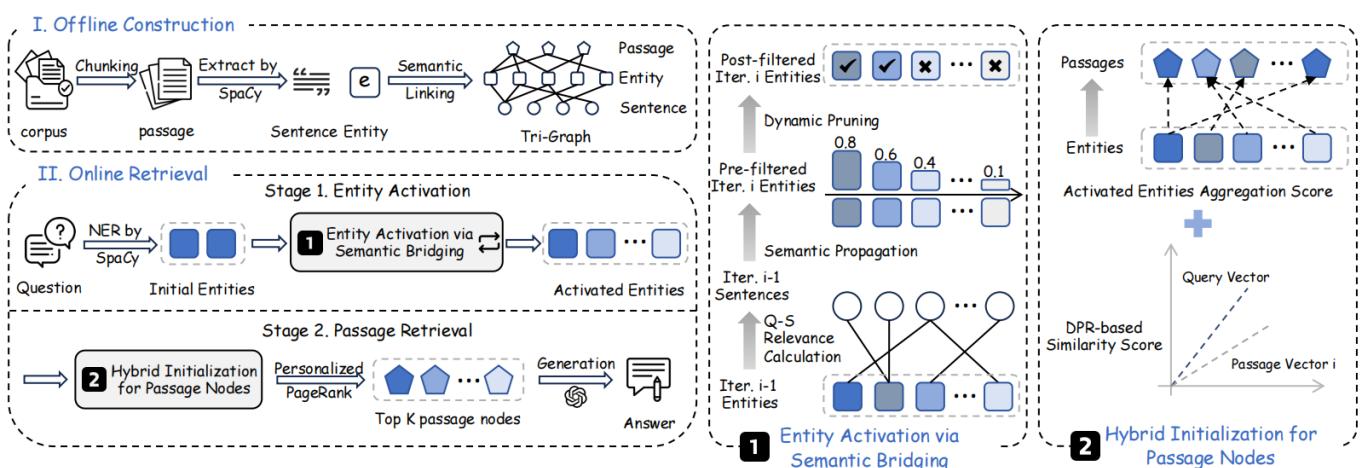
GraphExtraction/deal_triple.py 4. deal with duplicates of entries and relations

build_graph.py

1. generating embeddings
2. clustering lables (based on the embeddings)
3. layer 1 clustering
4. layer 2 clustering
5. building vector DB

4.1.2 LinearRAG

LinearRAG: Linear Graph Retrieval-Augmented Generation on Large-scale Corpora - A relation-free graph construction method for efficient GraphRAG.



Context-Preserving: Relation-free graph construction, relying on lightweight entity recognition and semantic linking to achieve comprehensive contextual comprehension. Complex Reasoning: Enables deep retrieval via semantic bridging, achieving multi-hop reasoning in a single retrieval pass without requiring explicit relational graphs. High Scalability: Zero LLM token consumption, faster processing speed, and linear time/space complexity.

Graphbuilding:

1. => load data
2. chunking data
3. get named entities - SpacyNER (Named Entity Recognition)
4. sentence splitting
5. get passages
6. get embeddings(sentences, entities, passages)
7. build graph => LinearRAG.graphml => ner_results.json => passage_embedding.parquet => sentence_embedding.parquet => entity_embedding.parquet

Retreival:

1. retrieval_results = qa(question)

linearRAG Results

LinearRAG, Dataset: 2wikimultihop, Results with local GPT-OSS-20b Model

[passage] Loaded 658 records from ./import\2wikimultihop\passage_embedding.parquet [entity] Loaded 40320 records from ./import\2wikimultihop\entity_embedding.parquet [sentence] Loaded 21206 records from ./import\2wikimultihop\sentence_embedding.parquet

2025-12-09 12:16:23,189 - INFO - Evaluation Results: 2025-12-09 12:16:23,191 - INFO - LLM Accuracy: 0.7350 (735.0/1000) 2025-12-09 12:16:23,191 - INFO - Contain Accuracy: 0.7210 (721/1000)

LinearRAG, Dataset: 2wikimultihop, Results with online gpt-4o-mini Model

[passage] Loaded 658 records from ./import\2wikimultihop\passage_embedding.parquet [entity] Loaded 40320 records from ./import\2wikimultihop\entity_embedding.parquet [sentence] Loaded 21206 records from ./import\2wikimultihop\sentence_embedding.parquet Retrieving: 100%

██████████ | 1000/1000 [02:43<00:00, 6.12it/s] QA Reading (Parallel): 100%

1000/1000 [03:48<00:00, 4.37it/s] Evaluating samples: 100%

██████████ | 1000/1000 [00:40<00:00, 24.70sample/s,

LLM_Acc=0.639, Contain_Acc=0.693] 2025-12-09 13:34:30,325 - INFO - Evaluation Results: 2025-12-09 13:34:30,325 - INFO - LLM Accuracy: 0.6390 (639.0/1000) 2025-12-09 13:34:30,325 - INFO - Contain Accuracy: 0.6930 (693/1000)

LinearRAG, Dataset: 2wikimultihop, Results with remote gemma3:17b Model

[passage] Loaded 658 records from ./import\2wikimultihop\passage_embedding.parquet [entity] Loaded 40320 records from ./import\2wikimultihop\entity_embedding.parquet [sentence] Loaded 21206 records from ./import\2wikimultihop\sentence_embedding.parquet Retrieving: 100%

██████████ | 1000/1000 [03:10<00:00, 5.24it/s] QA Reading (Parallel): 100%

██████████ | 1000/1000 [1:22:15<00:00, 4.94s/it] Evaluating samples: 100%

██████████ | 1000/1000 [03:24<00:00, 4.88sample/s, LLM_Acc=0.240, Contain_Acc=0.351] 2025-12-09 19:02:34,979 - INFO - Evaluation Results: 2025-12-09 19:02:34,980 - INFO - LLM Accuracy: 0.2400 (240.0/1000) 2025-12-09 19:02:34,981 - INFO - Contain Accuracy: 0.3510 (351/1000)

LinearRAG, Dataset: 2wikimultihop, Results with online gpt-4o Model

[passage] Loaded 658 records from ./import\2wikimultihop\passage_embedding.parquet [entity] Loaded 40320 records from ./import\2wikimultihop\entity_embedding.parquet [sentence] Loaded 21206 records from ./import\2wikimultihop\sentence_embedding.parquet Retrieving: 100%

██████████ | 1000/1000 [03:00<00:00, 5.55it/s] QA Reading (Parallel): 100%

██████████ | 1000/1000 [03:29<00:00, 4.78it/s] Evaluating samples: 100%

██████████ | 1000/1000 [00:40<00:00, 24.96sample/s, LLM_Acc=0.590, Contain_Acc=0.755] 2025-12-09 19:32:14,264 - INFO - Evaluation Results:

2025-12-09 19:32:14,264 - INFO - LLM Accuracy: 0.5900 (590.0/1000) 2025-12-09 19:32:14,265 - INFO - Contain Accuracy: 0.7550 (755/1000)

LinearRAG, Dataset: hotpotqa, Results with local GPT-OSS-20b Model

[passage] Loaded 1311 records from ./import\hotpotqa\passage_embedding.parquet [entity] Loaded 66846 records from ./import\hotpotqa\entity_embedding.parquet [sentence] Loaded 38455 records from ./import\hotpotqa\sentence_embedding.parquet Retrieving: 100%

Reading (Parallel): 100%

| 1000/1000 [1:51:26<00:00, 6.69s/it] Evaluating samples: 100%

1000/1000 [24:59<00:00, 1.50s/sample, LLM_Acc=0.771, Contain_Acc=0.662] 2025-12-10 20:59:41,463 - INFO - Evaluation Results: 2025-12-10 20:59:41,463 - INFO - LLM Accuracy: 0.7710 (771.0/1000) 2025-12-10 20:59:41,463 - INFO - Contain Accuracy: 0.6620 (662/1000)

LinearRAG, Dataset: musique, Results with local GPT-OSS-20b Model

[passage] Loaded 1354 records from ./import\musique\passage_embedding.parquet [entity] Loaded 67532 records from ./import\musique\entity_embedding.parquet [sentence] Loaded 39110 records from ./import\musique\sentence_embedding.parquet Retrieving: 100%

| 1000/1000 [03:15<00:00, 5.13it/s] QA Reading (Parallel): 100%

| 1000/1000 [3:51:21<00:00, 13.88s/it] Evaluating samples: 100%

| 1000/1000 [17:39<00:00, 1.06s/sample, LLM_Acc=0.642, Contain_Acc=0.317] 2025-12-11 02:00:28,341 - INFO - Evaluation Results: 2025-12-11 02:00:28,342 - INFO - LLM Accuracy: 0.6420 (642.0/1000) 2025-12-11 02:00:28,342 - INFO - Contain Accuracy: 0.3170 (317/1000)

LinearRAG, Dataset: medical, Results with local GPT-OSS-20b Model

[passage] Loaded 225 records from ./import\medical\passage_embedding.parquet [entity] Loaded 9033 records from ./import\medical\entity_embedding.parquet [sentence] Loaded 8985 records from ./import\medical\sentence_embedding.parquet Retrieving: 100%

| 2062/2062 [06:03<00:00, 5.67it/s] QA Reading (Parallel): 100%

| 2062/2062 [10:51<00:00, 3.17it/s] Evaluating samples: 100%

| 2062/2062 [01:26<00:00, 23.72sample/s, LLM_Acc=0.694, Contain_Acc=0.032] 2025-12-11 09:33:43,939 - INFO - Evaluation Results: 2025-12-11 09:33:43,939 - INFO - LLM Accuracy: 0.6940 (1431.0/2062) 2025-12-11 09:33:43,939 - INFO - Contain Accuracy: 0.0320 (66/2062)

4.1.3 GraphMERT

GraphMERT: Effiziente und skalierbare Gewinnung zuverlässiger Wissensgraphen aus unstrukturierten Daten

Ein einfaches Beispiel für eine Testimplementierung des Princeton GraphMERT-Papers.

<https://arxiv.org/abs/2510.09580>

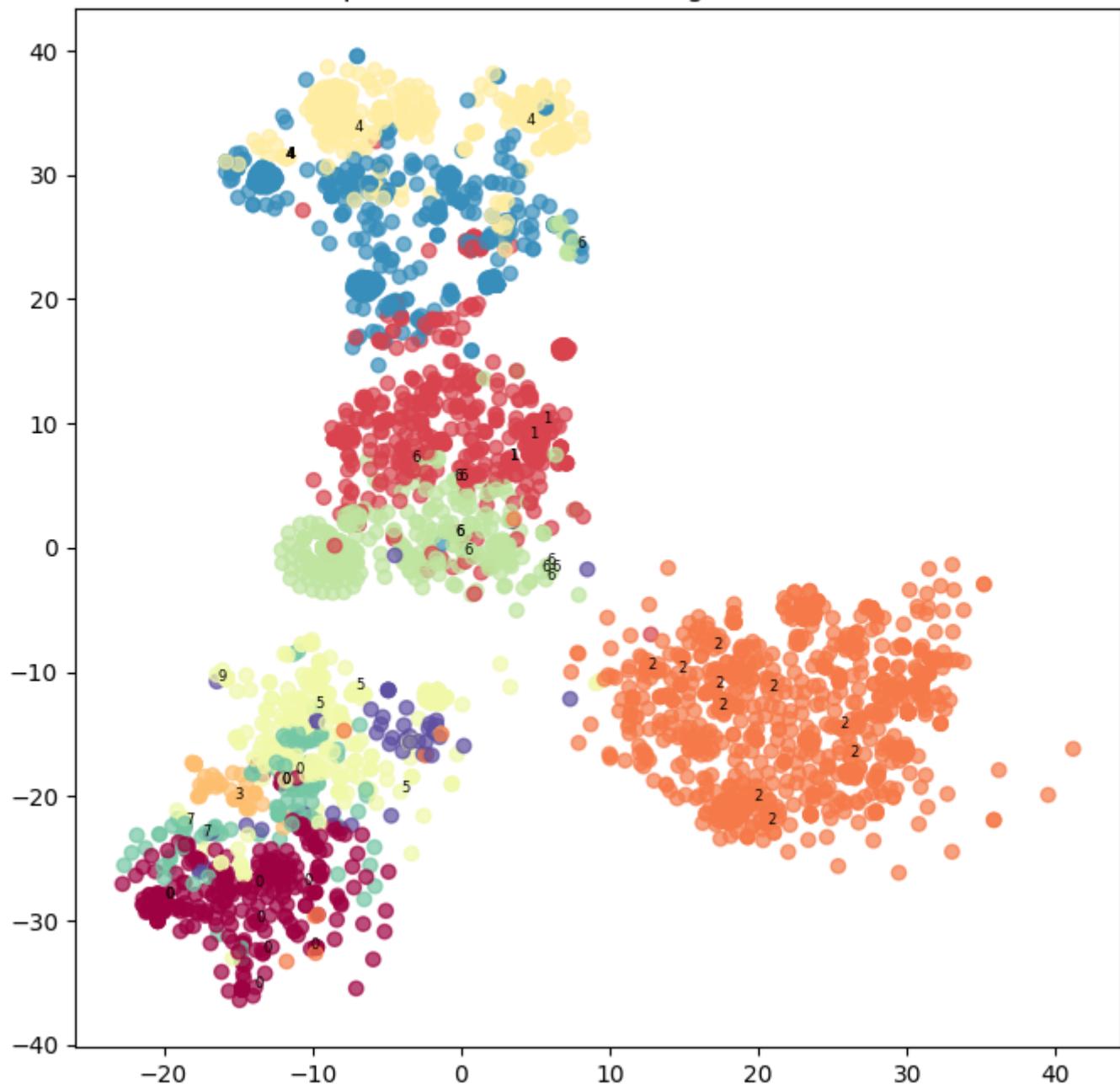
Seit fast drei Jahrzehnten erforschen Wissenschaftler Anwendungen neurosymbolischer künstlicher Intelligenz (KI), da symbolische Komponenten Abstraktion und neuronale Komponenten Generalisierung ermöglichen. Die Kombination beider Komponenten verspricht rasante Fortschritte in der KI. Dieses Potenzial konnte das Feld jedoch bisher nicht ausschöpfen, da die meisten neurosymbolischen KI-Frameworks nicht skalierbar sind. Zudem schränken die impliziten Repräsentationen und das approximative Schliessen neuronaler Ansätze Interpretierbarkeit und Vertrauen ein. Wissensgraphen (KGs), die als Goldstandard für die Repräsentation expliziten semantischen Wissens gelten, können die symbolische Seite abdecken. Die automatische Ableitung zuverlässiger KGs aus Textkorpora stellt jedoch weiterhin eine Herausforderung dar. Wir begegnen diesen Herausforderungen mit GraphMERT, einem kompakten, rein grafischen Encoder-Modell, das hochwertige KGs aus unstrukturierten Textkorpora und seinen eigenen internen Repräsentationen generiert.

GraphMERT und sein äquivalenter Wissensgraph bilden einen modularen neurosymbolischen Stack: neuronales Lernen von Abstraktionen; symbolische Wissensgraphen für verifizierbares Schliessen. GraphMERT + Wissensgraph ist das erste effiziente und skalierbare neurosymbolische Modell, das höchste Benchmark-Genauigkeit und überlegene symbolische Repräsentationen im Vergleich zu Basismodellen erzielt.

Konkret streben wir zuverlässige domänenspezifische Wissensgraphen (KGs) an, die sowohl (1) faktisch korrekt (mit Herkunftsnnachweis) als auch (2) valide (ontologiekonsistente Relationen mit domänenspezifischer Semantik) sind. Wenn ein grosses Sprachmodell (LLM), z. B. Qwen3-32B, domänenspezifische KGs generiert, weist es aufgrund seiner hohen Sensitivität, seiner geringen Domänenexpertise und fehlerhafter Relationen Defizite in der Zuverlässigkeit auf. Anhand von Texten aus PubMed-Artikeln zum Thema Diabetes erzielt unser GraphMERT-Modell mit 80 Millionen Parametern einen KG mit einem FActScore von 69,8 %; ein LLM-Basismodell mit 32 Milliarden Parametern erreicht hingegen nur einen FActScore von 40,2 %. Der GraphMERT-KG erzielt zudem einen höheren ValidityScore von 68,8 % gegenüber 43,0 % beim LLM-Basismodell.

GraphMERT Node Embeddings (t-SNE View)

GraphMERT Node Embeddings (t-SNE View)



GraphMERT Semantic Graph Visualization

GraphMERT Semantic Graph Visualization



Query search on the graphs results Das ist es, was wir wollen, da die Suche im Graphen linear ist und auf verkettetem Wissen basiert, wobei die Knoten Daten über sich selbst enthalten.

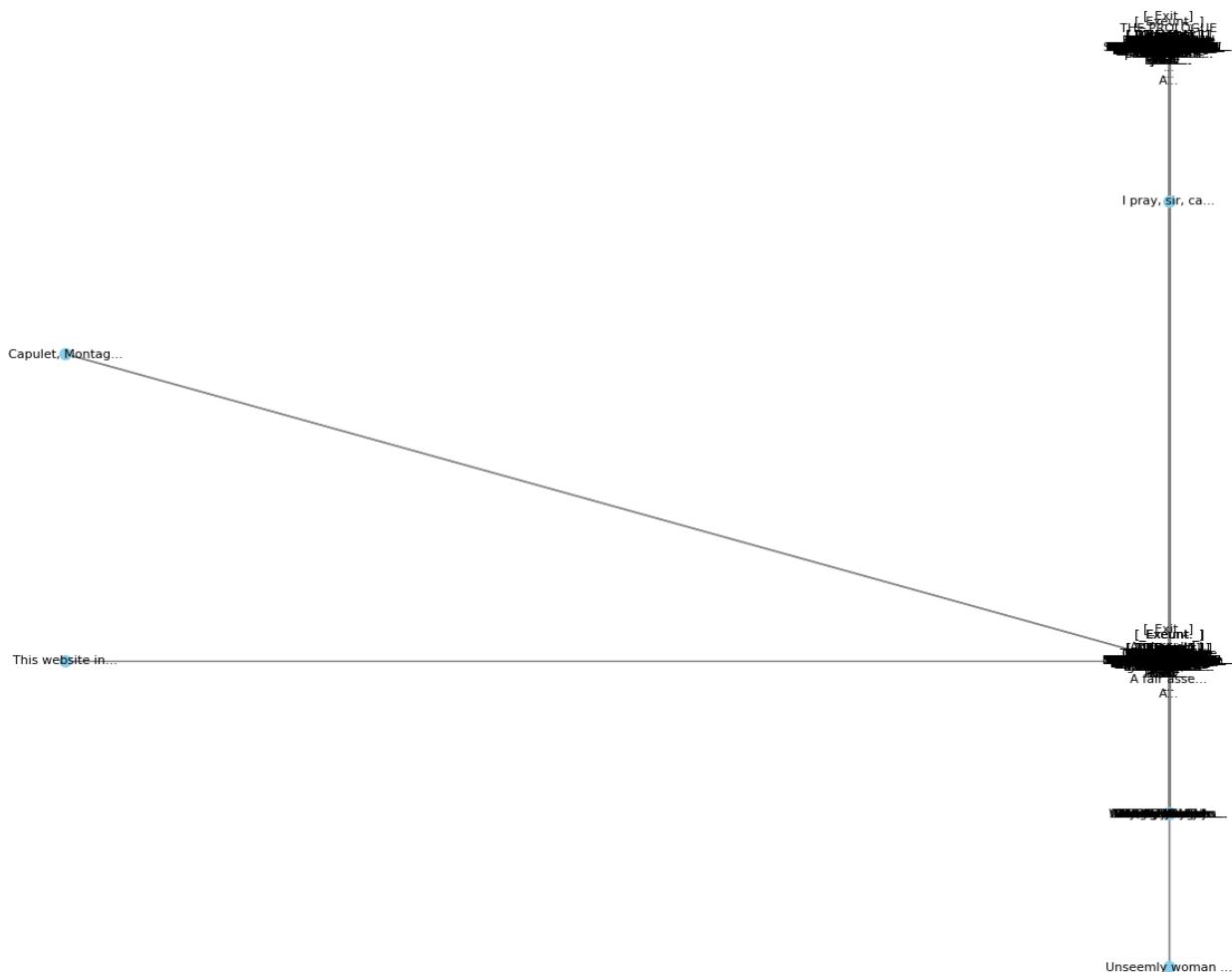
Ein perfektes Resultat

Graph Visualization from GraphMERT Model Output Embeddings



Ein fast perfektes Resultat

Graph Visualization from GraphMERT Model Output Embeddings



- **Extraktion:** Umwandlung von Text in Entitäten und Relationen.
-

Aggregation: Semantische Aggregation zur Reduzierung von Redundanz.

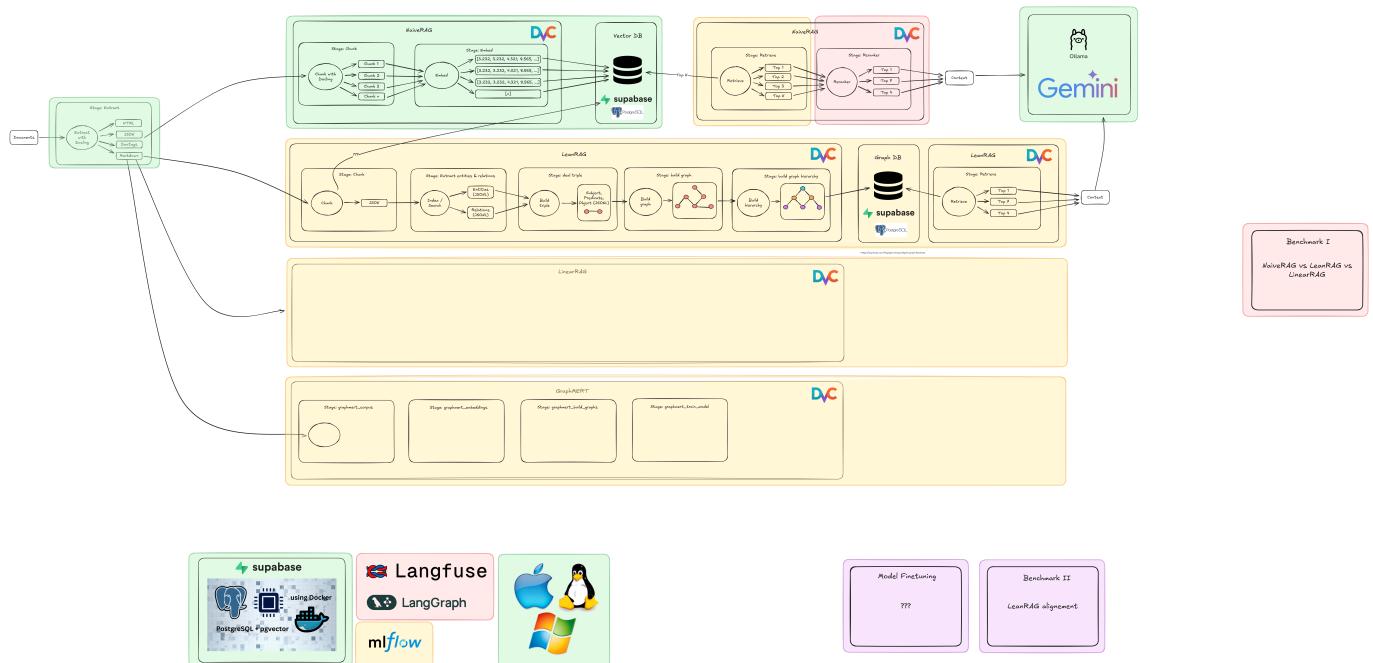
4.2 Fine-tuning Strategie

- Verwendung des **Unsloth Frameworks** für ressourceneffizientes Training.
- Integration von Ansätzen wie **GraphRAFT** oder **GraphMERT** zur Distillation von Wissen in kleine, domänenspezifische Modelle.

5. Implementierung

5.1 Systemarchitektur

Beschreibung der Pipeline von der PDF-Eingabe bis zur Antwortgenerierung.



5.2 Verwendete Hardware

Dokumentation der genutzten Ressourcen (z.B. 1x 4090 Desktop, M3 Pro 24GB) . 1 HP EliteBook X G11 => Massenextraktion mit Docing Prozessor Intel 5U

1 Lenovo Notbook Legion 9 16IRX8 Prozessor 13th Gen Intel(R) Core(TM) i9-13980HX (2.20 GHz) Installierter RAM 32.0 GB (31.7 GB verwendbar) GPU Nvidia RTX4090 Mobile mit 16GB VRAM

6. Evaluation und Benchmarking

6.1 Benchmark-Design

-

Ansatz 1: Generierung eines Testdatensatzes mittels Synthetic Data Generation (SDG) und Evaluierung durch ein "LLM als Judge".

-

Ansatz 2: Nutzung publizierter Benchmarks wie dem Open RAG Benchmark.

6.2 Ergebnisse

Vergleich der Performance: Standard RAG vs. IMARA GraphRAG vs. Fine-tuned Model.

7. Diskussion der Ergebnisse

- Qualität der generierten Graphen.
- Effektivität des Fine-tunings im Vergleich zu GPT-basierten Modellen.
- Ressourcenverbrauch und Skalierbarkeit.

8. Risikomanagement und Lessons Learned

Reflektion über die im Antrag identifizierten Risiken:

- Datenqualität und Graph-Dichte.
- Rechenintensität des Fine-tunings.
- Teamkoordination.

9. Fazit und Ausblick

Zusammenfassung, ob ein 80M domänen spezifisches Modell tatsächlich größere Modelle übertreffen konnte, und mögliche nächste Schritte.

10. Referenzen

- [1] Docling: An Efficient Open-Source Toolkit.
- [2] LeanRAG: Knowledge-Graph-Based Generation.
- [3] GraphMERT: Efficient Distillation of Reliable KGs.
- ... (Weitere Quellen gemäß Antrag).

11. Glossar

xxx

Tipps für die Ausarbeitung:

- **Visualisierungen:** Nutzt die Grafiken aus eurem Zwischenbericht (LeanRAG/Docling Architektur), um die technischen Sektionen (Kapitel 3 & 4) zu füllen.
- **Code-Beispiele:** Fügt kurze Snippets eurer Automatisierungslösung oder der Unslot-Konfiguration in Kapitel 5 ein.
- **Metriken:** In Kapitel 6 solltet ihr Tabellen mit Latenzzeiten und Genauigkeitswerten (Accuracy/F1) eurer Benchmarks zeigen.

Soll ich dir beim Ausformulieren eines spezifischen Kapitels (z.B. der Methodik oder der Evaluation) behilflich sein?