Leveraging Knowledge Graphs for Legal Document Analysis: Evaluating GraphRAG with German Legal Data

Sushmita Singh
sushmita.singh@fau.de

Project: GraphRAG Analysis
Degree: M.Sc. Data Science
Matriculation #: 23131103

1 Introduction

The advent of retrieval-augmented generation (RAG) frameworks has shown promise in improving the quality and accuracy of large language models (LLMs). However, applying these techniques to domain-specific datasets, such as German legal texts, remains a challenge. This report explores the use of GraphRAG, a framework for integrating LLMs with knowledge graphs, to enhance the accuracy and relevancy of responses in this domain.

The primary objective of this project is to evaluate the effectiveness of GraphRAG when applied to German legal datasets. We compare the quality of answers generated by a baseline LLM to answers enhanced with domain knowledge by using GraphRAG. Here we use the framework Ragas to evaluate the answers using metrics answer correctness and answer relevancy.

This report is structured as follows:

- Section 2 describes the dataset used for GraphRAG indexing and querying.
- Section 3 reviews related work, including GraphRAG, vLLM, and Ragas metrics.
- Section 4 details the methods and tools used in this project.
- Section 5 discusses the experiments conducted to evaluate the system.
- Section 6 presents the results and findings.
- Section 7 outlines possible future work.
- Section 8 provides the conclusion for this study.

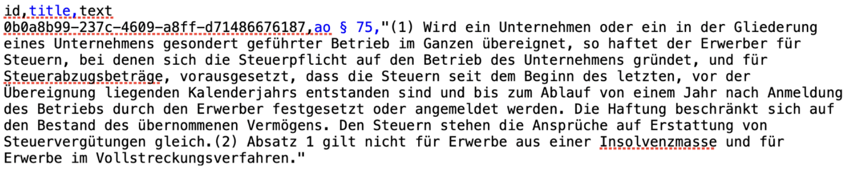
2 Dataset

2.1 Input for GraphRAG Indexing

The input dataset for GraphRAG is in the `csv` file format. GraphRAG supports two types of input files: `csv` and `text`. For this project, the `csv` format was chosen as the most suitable.

The data source used in this project is German law data obtained from the Bundesministerium der Justiz website (2). The input `csv` file requires two columns: `title` and `text`.

- **Title:** Contains the name of the law and its corresponding law number, for example, `ao § 75`, where `ao` stands for *Abgabenordnung*.



```
id,title,text
0b0a8b99-237c-4609-a8ff-d71486676187,ao § 75,"(1) Wird ein Unternehmen oder ein in der Gliederung
eines Unternehmens gesondert geführter Betrieb im Ganzen übereignet, so haftet der Erwerber für
Steuern, bei denen sich die Steuerpflicht auf den Betrieb des Unternehmens gründet, und für
Steuerabzugsbeträge, vorausgesetzt, dass die Steuern seit dem Beginn des letzten, vor der
Übereignung liegenden Kalenderjahrs entstanden sind und bis zum Ablauf von einem Jahr nach Anmeldung
des Betriebs durch den Erwerber festgesetzt oder angemeldet werden. Die Haftung beschränkt sich auf
den Bestand des übernommenen Vermögens. Den Steuern stehen die Ansprüche auf Erstattung von
Steuervergütungen gleich.(2) Absatz 1 gilt nicht für Erwerbe aus einer Insolvenzmasse und für
Erwerbe im Vollstreckungsverfahren."
```

Figure 1: Example of Input for GraphRAG Indexing

- **Text:** Contains the content of the law corresponding to the title.

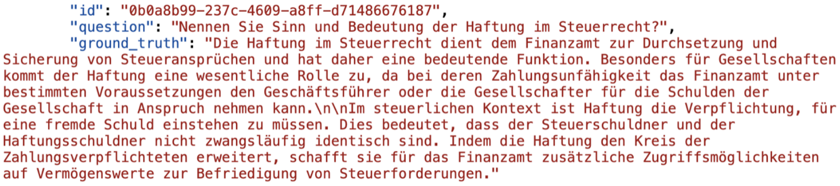
A sample structure of the input file is shown in Figure 1.

2.2 Input for GraphRAG Querying and Evaluation

The input for querying and evaluation is structured with the following keys:

- **Question:** Used by GraphRAG for retrieval.
- **Ground Truth:** Used during evaluation by the Ragas framework.
- **ID:** A unique identifier for each question. This key is used for programming purposes only and does not affect indexing or retrieval.

A sample structure of the input for querying and evaluation is shown in Figure 2.



```
        "id": "0b0a8b99-237c-4609-a8ff-d71486676187",
        "question": "Nennen Sie Sinn und Bedeutung der Haftung im Steuerrecht?",
        "ground_truth": "Die Haftung im Steuerrecht dient dem Finanzamt zur Durchsetzung und
Sicherung von Steueransprüchen und hat daher eine bedeutende Funktion. Besonders für Gesellschaften
kommt der Haftung eine wesentliche Rolle zu, da bei deren Zahlungsunfähigkeit das Finanzamt unter
bestimmten Voraussetzungen den Geschäftsführer oder die Gesellschafter für die Schulden der
Gesellschaft in Anspruch nehmen kann.\n\nIm steuerlichen Kontext ist Haftung die Verpflichtung, für
eine fremde Schuld einstehen zu müssen. Dies bedeutet, dass der Steuerschuldner und der
Haftungsschuldner nicht zwangsläufig identisch sind. Indem die Haftung den Kreis der
Zahlungsverpflichteten erweitert, schafft sie für das Finanzamt zusätzliche Zugriffsmöglichkeiten
auf Vermögenswerte zur Befriedigung von Steuerforderungen."
```

Figure 2: Example of Input for GraphRAG Querying and Evaluation

3 Related Work

3.1 GraphRAG

GraphRAG offers a structured and hierarchical alternative to naive semantic-search methods by utilizing a knowledge graph extracted from raw text. It employs an LLM to derive an entity knowledge graph from the source documents and generate community summaries of related entities (3).

3.2 vLLM

vLLM is a framework developed by Sky Computing Lab at UC Berkeley for LLM inference and serving. It provides state-of-the-art serving throughput. In this project, it is used to serve both the LLM and the embedding model (7).

3.3 Llama-3.1-8B-Instruct

The Llama 3.1 instruction-tuned text-only models (8B, 70B, 405B) are optimized for multilingual dialogue use cases. It is an auto-regressive language model that uses an optimized transformer architecture. As of writing this report, it supports eight languages, including German (6).

3.4 Ragas Metrics

RAGAS (Retrieval Augmented Generation Assessment) is a framework for evaluating Retrieval Augmented Generation (RAG) pipelines. It provides a suite of metrics to assess the retrieval and generation components of RAG systems. In this project, answer relevance and answer correctness are used to evaluate the answers generated by GraphRAG.(4)

3.4.1 Answer Relevancy

Response Relevancy measures how well a response aligns with the user input, based on the average cosine similarity between the input embedding and embeddings of questions generated from the response. Higher scores indicate better alignment, focusing on relevance rather than accuracy. The score typically ranges between -1 and 1. A good response allows the original question to be inferred from the answer (15).

3.4.2 Answer Correctness

The assessment of Answer Correctness involves gauging the accuracy of the generated answer when compared to the ground truth. This evaluation relies on the ground truth and the generated answer, with scores ranging from 0 to 1. A higher score indicates a closer alignment between the generated answer and the ground truth, signifying better correctness. Answer correctness encompasses two critical aspects: semantic similarity between the generated answer and the ground truth, as well as factual similarity. These aspects are combined using a weighted scheme to formulate the answer correctness score (14).

4 Method

4.1 Prompt Tuning

The auto prompt tune feature of GraphRAG creates domain-adapted prompts for generating the knowledge graph. This step is optional but encouraged to yield better results when executing the index run. It processes inputs, splits them into chunks, and runs a series of LLM invocations and template substitutions to generate the final prompts. The modified prompts are stored in the `/prompts` folder. The prompt-tune script has several parameters, one of which is `language`, which, in this project, is set to `German`.(9) Details of the command can be found in Appendix A.

4.2 Indexing

The indexing script extracts meaningful, structured data from unstructured text using LLMs. The configurable indexing pipeline performs the following tasks:

- Extracting entities, relationships, and claims from raw text.
- Performing community detection among entities.
- Generating community summaries and reports at multiple granularity levels.
- Embedding entities into a graph vector space.
- Embedding text chunks into a textual vector space.

The outputs are stored as Parquet tables (default location: `/output` folder) and embeddings are saved in the configured vector store (default: LanceDB).(10) The exact command is detailed in Appendix A.

4.3 Querying

The Query Engine retrieves information from the knowledge graph using one of four task methods: local search, global search, drift search, and question generation. This project uses the **local search method**, which generates answers by combining structured data from the knowledge graph with unstructured data from input documents. It augments the LLM context with relevant entity information, making it well-suited for answering questions about specific entities.(11) The exact command is provided in Appendix A.

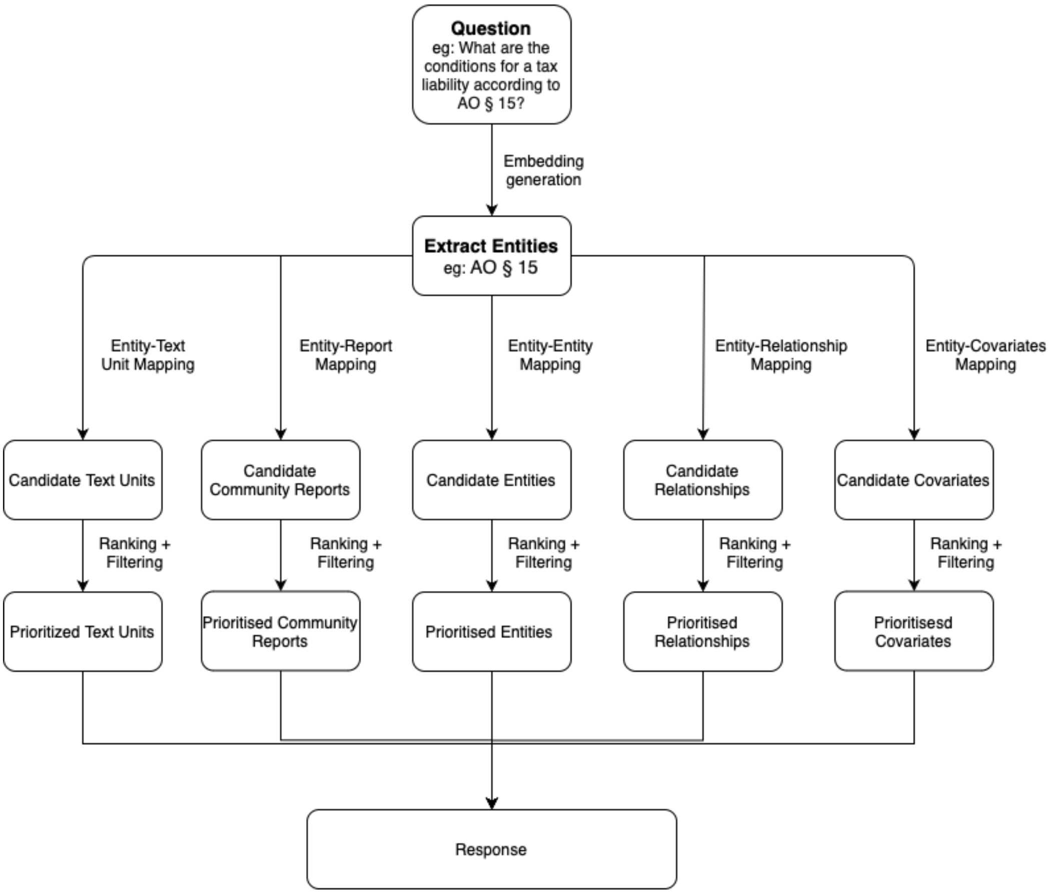


Figure 3: Local Search Dataflow for Querying in GraphRAG

5 Experiments

5.1 Setup and Running the vllm Servers

The vllm servers were set up by activating the appropriate environment, requesting GPU resources (5), and proxy settings were configured to enable internet access and exclude localhost from proxy routing. Versions of the main libraries, LLM models, and hardware specifications used in this project can be found in Appendix B. Detailed shell commands used in this setup are provided in Appendix C.

5.2 GraphRAG Initialization and Configuration

GraphRAG was initialized and configured following the official documentation (8). The initialization process created a workspace with default configuration files (`.env` and `settings.yaml`). Modifications were made to the settings.yaml file to configure it for using the local LLM served by vLLM. Some default parameters of GraphRAG were modified to improve the performance of the LLM. The specific changes to these parameters are detailed below:

- **LLM_MAX_TOKENS:** 1024 (reduced from 4000) – Reduced context size for LLM requests.
- **LLM_TEMPERATURE:** 0.3 (increased from 0) – Introduces response variability.
- **LLM_TOP_P:** 0.8 (reduced from 1) – Limits token diversity in responses.
- **LLM_CONCURRENT_REQUESTS:** 15 (reduced from 25) – Allows fewer concurrent requests.

Details of the initialization command and modifications to the `settings.yaml` file are available in Appendix C.

5.3 GraphRAG Prompt Tune, Index, and Query

- **Prompt Tune:** The prompt tuning engine was run to improve the suitability of default prompts for the dataset.
- **Indexing:** The indexing engine was executed to create the GraphRAG.
- **Query:** The query engine was run using the `local` method, where retrieved chunks from GraphRAG were included in the prompt to generate LLM answers.

Details of the commands used for these operations are provided in Appendix A.

5.4 Ragas Evaluation and Baseline Creation

5.4.1 Calculate Ragas Score

To evaluate GraphRAG, the Ragas framework was used. The evaluation utilized the LLM `gpt-4o-mini`, with metrics including Answer Correctness and Answer Relevancy.

5.4.2 Baseline Creation

To establish a baseline, the model `meta-llama/Llama-3.1-8B-Instruct` was used to generate answers for each dataset question. The Ragas answer correctness and answer relevance metrics were then applied to calculate the mean scores, which served as a baseline for comparison with the answers generated by GraphRAG

6 Findings

Using GraphRAG improves both **Answer Correctness** and **Answer Relevancy** compared to using the LLM alone. Correctness increased from 0.4199 to 0.5018, representing an improvement of **+8.2%**. Similarly, relevancy improved from 0.6818 to 0.7617, an increase of **+8.0%**.

These results, summarized in Table 1, demonstrate that GraphRAG enhances both the factual accuracy and alignment of responses with user queries.

Metric	**LLM Only**	**With GraphRAG**	**Improvement (%)**
Answer Correctness	0.4199	0.5018	+8.2%
Answer Relevancy	0.6818	0.7617	+8.0%

Table 1: Comparison of Ragas Metrics with and without GraphRAG

7 Future Work

When selecting the LLM for GraphRAG, the primary criterion was support for German-language data. The Llama-3.1-8B-Instruct model is multilingual and supports eight languages, including German. Additionally, it is instruction-tuned, making it well-suited for assistant-like chat tasks, which align closely with the structure of our data.

Several other open-source and proprietary LLMs and embedding models could also be explored to improve the quality of the generated answers. Examples include Mistral-NeMo-12B-Instruct (13) and multilingual-e5-large (16).

Furthermore, modifying the default parameters of GraphRAG may enhance results. For instance, setting `LLM_TEMPERATURE=0.0` could produce deterministic outputs, potentially improving consistency.

In addition to these modifications, using DRIFT Search (Dynamic Reasoning and Inference with Flexible Traversal) could enhance the GraphRAG technique by combining global and local search to generate detailed responses. DRIFT Search incorporates community information, broadening the query's scope and enabling the retrieval of a wider variety of facts.

8 Conclusion

The comparison between the baseline LLM responses and those generated by combining the LLM with GraphRAG demonstrates clear improvements in evaluation metrics such as answer correctness and answer relevancy. This indicates that GraphRAG effectively retrieves relevant data from the knowledge graph, contributing to these improvements.

However, despite these advancements, the low accuracy score of the answer correctness metric, even when using GraphRAG, remains a challenge. To address this, the suggestions outlined in the Future Work section can be adopted to further enhance the system's performance and capabilities.

9 Appendix

A GraphRAG Prompt Tune, Index, Query Detailed Commands

Prompt Tuning Script

This command is executed inside the folder
`legalRAG`

```
python -m graphrag prompt-tune --root . --language German
```

Indexing Script

This command is executed from one directory above the
`legalRAG` folder.

```
python -m graphrag index --root ./legalRAG
```

Querying Script for GraphRAG

This command is executed from one directory above the
`legalRAG` folder.

```
graphrag query --root ./legalRAG --method local --query "$question"
```

B Technical Details

Versions of Main Libraries

- **vllm:** Version 0.6.5
- **graphrag:** Version 1.0.0
- **ragas:** Version 0.2.6

LLM Models Used in This Project

- **LLM:** `meta-llama/Llama-3.1-8B-Instruct` (1)

- **Embedding Model:** `intfloat/e5-mistral-7b-instruct` (12)

Hardware Used

- **GPU:** NVIDIA A100 with 80GB RAM

C Detailed Commands and Configurations

Setup and Running the vllm Servers

- **Activate the Base Environment:**

```
conda activate pytorch-2.3.0
```

- **Start the LLM Model Served Using vllm:**

```
python -m vllm.entrypoints.openai.api_server --model /path/
    to/models/meta-llama_Llama-3.1-8B-Instruct --port 8000
    --gpu_memory_utilization=0.7 --chat-template /path/to/
    scripts/tool_chat_template_llama3.1_json.jinja
```

- **Start the Embedding Model Served Using vllm:**

```
python -m vllm.entrypoints.openai.api_server --model /path/
    to/models/intfloat_e5-mistral-7b-instruct --port 8001
```

GraphRAG Initialization and Configuration

- **Initialize the Workspace:**
  The GraphRAG workspace is named `legalRAG` in this project.

```
graphrag init --root ./legalRAG
```

- **Modifications to `settings.yaml`:**

```
llm:
  model: /path/to/models/meta-llama_Llama-3.1-8B-Instruct
  model_supports_json: false
  api_base: http://localhost:8000/v1
embeddings:
  llm:
    model: /path/to/models/intfloat_e5-mistral-7b-instruct
    api_base: http://localhost:8001/v1
input:
  file_type: csv
  file_pattern: ".*\\.csv$"
```

References

[1] Meta AI. Llama 3.1: Instruct-tuned language model, 2024. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct.

[2] Bundesministerium der Justiz. Gesetze im internet. https://www.gesetze-im-internet.de/aktuell.html. Accessed: January 19, 2025.

[3] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2024.

[4] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2023.

[5] NHR FAU. Batch processing with slurm. https://doc.nhr.fau.de/batch-processing/batch_system_slurm/. Accessed: January 19, 2025.

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson,

Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin,

Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[7] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023.

[8] Microsoft. Graphrag documentation. https://microsoft.github.io/graphrag/get_started/. Accessed: January 19, 2025.

[9] Microsoft. Graphrag auto-prompt tuning documentation. Online, 2025. https://microsoft.github.io/graphrag/prompt_tuning/auto_prompt_tuning/.

[10] Microsoft. Graphrag indexing overview documentation. Online, 2025. https://microsoft.github.io/graphrag/index/overview/.

[11] Microsoft. Graphrag local search documentation. Online, 2025. https://microsoft.github.io/graphrag/query/local_search/.

[12] Mistral. E5-mistral 7b-instruct: Embedding model, 2023. https://huggingface.co/intfloat/e5-mistral-7b-instruct.

[13] NVIDIA-Mistral. Mistral-nemo-12b-base. https://huggingface.co/nvidia/Mistral-NeMo-12B-Base. Accessed: January 19, 2025.

[14] Ragas. Ragas: Retrieval-augmented generation assessment suite, 2023. https://docs.ragas.io/en/v0.1.21/concepts/metrics/answer_correctness.html.

[15] Ragas. Ragas: Retrieval-augmented generation assessment suite - answer relevance, 2025. https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/answer_relevance/.

[16] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.