Conversational AI. Dialogsysteme, Chatbots, Assistenten

Veranstalter: Christoph Ringlstetter

Sitzung 5: RAG

Was machen wir denn heute.

- Orga. Referate, Zulassung, Termine.
- News of the Week
- Referat RAG
- Besprechung RAG Survey Paper
- Besprechung Langchain 4 RAG Pipeline

Survey RAG Gao et al – Überblick Paper

Naive RAG

Advanced RAG

Modular RAG

Evaluation

Retrieval-Augmented Generation for Large Language Models: A Survey

Yunfan Gao[a], Yun Xiong[b], Xinyu Gao[b], Kangxiang Jia[b], Jinliu Pan[b], Yuxi Bi[c], Yi Dai[a], Jiawei Sun[a], Meng Wang[c], and Haofen Wang [a,c]

[a]Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University
[b]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[c]College of Design and Innovation, Tongji University

***Abstract*—Large Language Models (LLMs) showcase impressive capabilities but encounter challenges like hallucination, outdated knowledge, and non-transparent, untraceable reasoning processes. Retrieval-Augmented Generation (RAG) has emerged as a promising solution by incorporating knowledge from external databases. This enhances the accuracy and credibility of the generation, particularly for knowledge-intensive tasks, and allows for continuous knowledge updates and integration of domain-specific information. RAG synergistically merges LLMs' intrinsic knowledge with the vast, dynamic repositories of external databases. This comprehensive review paper offers a detailed examination of the progression of RAG paradigms, encompassing the Naive RAG, the Advanced RAG, and the Modular RAG.**

in Figure 1. The development trajectory of RAG in the era of large models exhibits several distinct stage characteristics. Initially, RAG's inception coincided with the rise of the Transformer architecture, focusing on enhancing language models by incorporating additional knowledge through Pre-Training Models (PTM). This early stage was characterized by foundational work aimed at refining pre-training techniques [3]–[5].The subsequent arrival of ChatGPT [6] marked a pivotal moment, with LLM demonstrating powerful in context learning (ICL) capabilities. RAG research shifted towards providing better information for LLMs to answer more com-

Survey RAG Gao et al – Überblick Paper

Limitation von LLMs domänenspezifisches Wissen, Aufgaben. Fragen die über die Trainingsdaten hinausgehen – inhaltlich, outdating

RAG: Basis sind relevante Chunks von Dokumenten mit LLM externem Wissen

1) Paradigmen von RAGs

2) Retrieval

3) Generation

4) Augmentation

5) query embeddings, index: Optimierung

6) postretrieval Optimierung + Finetuning

7) Evaluation

Survey RAG Gao et al – Überblick RAG

Ausgangsfrage and das Modell: „How do you evaluate the fact that OpenAIs CEO ….. In terms of power dynamics“ – probiere das mit Scholz-Lindner-Habeck

→ Retrieval. Chunk1, Chunk2, Chunk3

LLM RAG

Question: „How do you evaluate…In terms of power dynamics“

Please anser the question based on the following information:

Chunk1, Chunk2, Chunk3

3 Phasen im Research dazu: naive, advanced, modular

Survey RAG Gao et al – Naive RAG

Indexing, Retrieval, Generation. Auch als Retrieve-Read Framework bezeichnet

→ Indexing: Datensäubern, Reformatieren: Pain Text

→ Segmenting: Chunks erzeugen: einfach cut-off value

→ Embedding: Chunk Vektoren erzeugen aus statischem Embedding Store

→ Store: Schreiben der Chunks in eine Vektordatenbank

→ Query: Embedding der Query erzeugen: Achtung gleicher Embedding Store

→ Retrieval: Ähnlichkeitssuche im Vektorindex: TopK

→ Generation: aus Query+Selected Chunks einen kohärenten Prompt erzeugen

Model antwortet gemäß der Taskspezifikation: zieht aus internem Wissen

ev. auf Info aus den Chunks beschränken.

Ev. History mit einbeziehen: multiturn

Survey RAG Gao et al – Naive RAG

Unzulänglichkeiten:

Retrieval: precision und recall: irrellevant-fehlt

Generation: Hallunzination, Toxikalität, Bias: ähnlich wie Standard LLM

Augmentation: Information integrieren: inkohärente, unverbundene Ausgaben. Repetitiv. Mit redundanter Information. Stil, Tonalität nicht getroffen oder oszilierend.

→ für komplexe Anfragen – Information Needs: ein einzelner Retrievalschritt nicht genug.

→ Einschränkung auf nur den Retrievalinhalt: Dull. Wenig zufriedenstellend.

Survey RAG Gao et al – Advanced RAG

Spezische Verbesserungen um die offenen Punkte des Naive RAG zu lösen.
Pre – Post Retrieval verbessern/ergänzen

→ Indexing: Ein sliding Window um Satzgrenzen einzuhalten. Feinjustierte, linguistische Segmentierung ; Metadaten, Alignment, Mixed Retrieval
→ query: klarere Frage; auseinanterziehen: Rewriting; Transformation, Expansion siehe Literatur
Postretrieval: Integration der Retrieval Chunks mit der Query. Reranking von Chunks, Kompression. Context an die äußeren Grenzen des Prompts platzieren. Essentielle Teile auswählen: kritische Abschnitte, Kürzen.

Survey RAG Gao et al – Modular RAG

Adaptierbar. Suchmodul für Ähnlichkeitssuche.
Verbessern des Retrievers durch Finetuning auf die Applikation.
Sequentielle Abarbeitung mit Stopkriterien. End2End Training über die Komponenten hinweg. Siehe Literatur S.4.
Memory Funktion mit iterativer Selbstverbesserung.
Routing im RAG System unter Einbezug diverser Datenquellen.
Predict Modul nutzt das LLM um Redundanzen im Kontext zu erkennen.
Task Adapter Modul: RAG System auf Downstream Aufgaben einstellen.
Query generation: Task spezifische Retriever durch Few-Shot Query Generation.

Survey RAG Gao et al – Modular RAG

Patterns für modulare RAGs. Einfacher Modulaustausch sozusagen on the fly.
Z.B. Rewrite, Retrieve, Read: Das LLM kann die Retrieval Queries umschreiben durch ein Rewriting Modul mit einem Feedback Mechanismus.
Hybride Strategien.

RAG vs. Finetuning: S. 5.
Fazit: RAGs sind konsistent besser in der Performanz.
Finetuning: Customisierung des Modellverhaltens, Stil, Wiederholung von Vorgehensweise == Persönlichkeit??

Survey RAG Gao et al – Retrieval

Entscheidend: Relevantes Material. Source, granularity, sturcture type, preprocessing strategy, embedding model

Source: doc type: PDF, XML, KG, Wikipedia: domains diversity, crosslingual

Structural type: semistructuted. Z.B. PDF Tabelleninformation. Kann bei falschem splitting zu Datenzerstörung führen; Komplexität steigt auch bei erfolgreicher Tabelleneinlesung;

-- SQL Queries gegen die Tabellen.

-- Textbasierte Methonden: Flatten der Tabelle

Strukturierte Daten: z.B. Knowledge Graphen. KnowledGPT, G Retriever GNN

Graphartige Anfragen gegen die Knowledge Base

Survey RAG Gao et al – Retrieval

Mit LLMs erzeugter Content. Limitationen im Rag: über interne Situation des LLM „nachdenken“

→ known

→ unknown: dann Retriever einsetzen.

Braucht Alignment mit dem Pretraining.

Survey RAG Gao et al – Retrieval

Retrieval Granularity – Coarse, Fine: For Text: Token, Phrase, Sentence, Proposition, Chunks, Document: DenseX → Propositions – was ist das überhaupt?

Survey RAG Gao et al – Index Optimization

process, segment, embedd, store

=> Konstruktion des Index entscheidend ob der korrekte Inhalt während des Retrieval effizient erreicht werden kann.

1) Chunking Strategie. Fixed Nr of Tokens, 100,256 …. Content vs Noise

Sentence splits, Sliding windows, merging strategies

small2Big: ein Satz wird central retrieved und dann noch zwei Sätze Vor-Nachkontext.

Survey RAG Gao et al – Index Optimization

2) Metadata Attachments: Filterung der Chunks

→ timeaware mit Zeitstempeln

→ Metadaten anreichern: Summaries

→ hyphotetical: synthetische LLM queries vergleichen mit Laufzeitqueries

3) Sturkturierter Index. Hierarchische Struktur der Dokumente. Eltern-Kind Knoten mit Summaries die das Retrieval steuern

KG Index: mit Konzepten und Entities logische Zusammenhänge zwischen Inhalt und Struktur herstellen s. Literatur

KGP: Methode um einen Index zu bilden der mehrere Dokumente mit einem KG verbindet: Nodes = paragrafs, tables und edges = Lex Ähnlichkeiten oder Beziehungen in der Dokumentenstruktur

Survey RAG Gao et al – Query Optimization

Hauptproblem des naiven Retrieval. Unverändertes Verwenden der User Query für das Retrieval:

komplexe Anfragen/Informationneeds oft schlecht organisiert durch den menschlichen Nutzer.

Ambiguität, Domänensprache, Abkürzungen, implizite Voraussetzungen

1) Expansion der Query: Per LLM aus einer Frage mehrere machen. Mehr Kontext, Multiquery.

SubQuery: → notwendige Folgefragen z.B. least-to-most Prompting Lit [92]

Chain of Verification (CoVe)

Validierte Expansion mit LLM Filter

Survey RAG Gao et al – Query Optimization

2) Query Transformation: statt der Originalfrage ein Transformierte
Query Rewrite: LLM wird für die Umformung benutzt, auch spezielle LLMs RRR(rewrite, retrieve, read)
Taobao: BEQUE Rewrite für longtail queries. Stepback Methode: Hypothetische Antworten werden per Ähnlichkeit gepoolt und dann mit einem „Stepback“ Prompting aus der Query und den Hybrid Documents eine neue „abstraktere“ Query erzeugt.
3) Query Routing: disjunkte RAG Pipelines für diverse Szenarien anpassbar.
Metadata Router, Semantic Router.

Survey RAG Gao et al – Embedding Optimization

Naives RAG Retrieval: Cos zwischen den Embeddings. Query <-> Doc Chunks.

Je nach Encoders: sparse, dense.

AngIE, BGE, Voyage Embeddings. Mit multi-task instruction tuning.

MTEB Leader Bord für 8 Tasks, 58 Datensets

1) Mixed/Hybrid Retrieval. Sparse and Dense

<-> verschiedene relevante Features komplementäre Ergebnisse.

Sparse: Robustheit für seltene Entities

Survey RAG Gao et al – Embedding Optimization

2) Finetuning des Embedding Models:

Für Szenarien wo der Kontext sehr stark vom Pre-Training Korpus abweicht.:

Spezial Domänen: Health, Legal. Finetuning auf einem speziellen Datenset könnte den Unterschied in der Performanz beheben.

Auch Fine-Tuning für das Retrieval selbt könnte ein Grund für Finetuning sein – im Sinne Instruction Tuning.

Z.B. LSr: LM Supervised Retriever. Literatur für Vorgehensweisen im Paper.

Survey RAG Gao et al – Adapter

Adapter: External Adapter um Funktionaliät in das Retrievalmodell einzubringen.

Uprise: Prompt von einem Prebuilt Pool für erkannte Zeroshot Tasks holen

AAR: Augmentation Adapter Retriever. Adapter um das Retrieval an verschiedene Downstreamtasks anzupassen

BGM: Brückenmodell zw. Retriever und LLM Seq2Seq Lit 75

Survey RAG Gao et al – Generation

Nach dem Retrieval nicht einfach alle eingeholte Information in das LLM ziehen um die Query zu beantworten:

→ den eingeholten Kontext anpassen

→ das LLM anpassen

Context Curation: Redundante Information kannmit dem final gewünschten Generation Ergebnis in Widerspruch stehen. Überlange Kontexte: Lost in the Middle Problem Lit. 98

1) Reranking: highlight die vielversprechensten Resultate zuerst Lit 70 Enhancer und Filter

Survey RAG Gao et al – Generation

Regelbasiert: Diversity Relevanced.
Spezialisierte Modelle. Kohärenz Rerank.

2) Kontextauswall, Kompression.
→ Missverständnis → Retrieval möglichst vieler Dokumente und Konkatenation dieser. NOISE.
→ LLM Lingua benutzt kleine Modell um unwichtige Token aus dem Kontext zu entfernen.
→ Kondensierte Form: Information Extraction Filter: SLMs als Filter und LLM als Reordering Agents.

Survey RAG Gao et al – Generation

LLM evaluiert den Kontext (Retrieval Content) vor der Generation == LLM Critique

B: Finetuning, Gezieltes Finetuning auf dem Szenario: Dokumentenarten, Datenformate, Stil. Siehe Literatur.

Survey RAG Gao et al – Augmentation Prozess im RAG

Standard: oft singulärer Retrieval Step. Multistep bei komplizierten Problemen erforderlich.

A: Iterative Retrieval: basierend auf der initialen Query und dem generierten Text bislang.

Retrieval Enchanced Generation + Generation Enchanced Retrieval

B: Recursive Retrieval: Refine search query basierend auf dem was schon gefunden wurde. Konvergenz durch Feedback. Rekursiv z.T. auf Basis von Chain of Thought guidance.

Strukturiert: hierachisch je nach Index Gestalt. Lit. 106

Survey RAG Gao et al – Augmentation Prozess im RAG

C: Adaptive Retrieval: optimieren der Zeitpunkte des Retrievals (Sequentiell in der Generation) und des einzubringenden Inhalts.
=> mit Adapter: Flare, Self-RAG

Trend: LLM Installationen benutzen für ihre operativen Entscheidungen/Vorgehen
aktive Wertungen: Auto-GPT Lit. 106-109
WEB-GPT: GPT Instruction tuned RL für optimierte/autonome Suchmaschinenbenutzung durch das LLM
Flare: automatisiert das Timing über die Konfidenz im Generation Prozess: Unterschreiten einer Threshold: Retrievalschritt

Survey RAG Gao et al – Augmentation Prozess im RAG

SELF-RAG: Reflection Token steuern die Inspektion des Outputs:

- **retrieve**
- **critic**

Survey RAG Gao et al – Tasks and their Evaluation

Hauptaufgabe LLM-RAG: QA simple+multihop + downstream tasks: ID, Dialoggeneration, Code, Search, NLP

Traditionelle Evaluation: Assessment über die Downstream Tasks und verwendung der etablierten Metriken: ACC, Entropy LLM, F1, Bleu, Rouge.

Jetzt:

Retrieval Quality: um die Effektivität des Kontexts anzuschauen.

Hit Rate, MRR, NDCG Standardmetriken von Suchmaschinen, IR, Recommenders

Survey RAG Gao et al – Tasks and their Evaluation

Generation Quality: synthetisieren von kohärenten und relevanten Antworten aus dem zurückerhaltenen Kontext: Bewertung des generierten Content.

→ ungelabelt: Faithfulness, Relevanz (?how), Harmfulness

→ labeled: Acc manuell/automatisch Lit 161

Evaluation Aspects: 3 Qualitätsscores, 4 Fähigkeiten Lit 164 -166

Quality Scores:

- Kontext Relevanz: Präzision, Spezifität des Kontexts für Antwort
- Antwort Faithfulness: generierte Antworten sind wahr, konsistent in Bezug auf den Kontext: keine Widersprüche
- Antwort Relevanz: effektiv den Informationneed erfüllen.

Survey RAG Gao et al – Tasks and their Evaluation

Erforderliche Fähgikeiten:

- Noise Robustness: ausschließen von nutzlosen Dokumenten mit Querybezug
- Negative Rejection: zurückhalten der Antwort wenn Dokumente nicht genug Information enthalten.
- Information Integration: Aus mehreren Dokumenten Informationen zusammenfügen um komplexe Fragen zu beantworten.
- Counterfactual Robustness: Erkennen und zurückweisen bekannter fehlerhafter Generierung

Datensets: S. 13, Measures für Downstream: S.14. Table III