DHBW Stuttgart

Retrieval Augmented Generation (RAG) - Verschiedene Methoden

Forschungsseminar | 03.06.24 |
Tien Dee Lin

Agenda

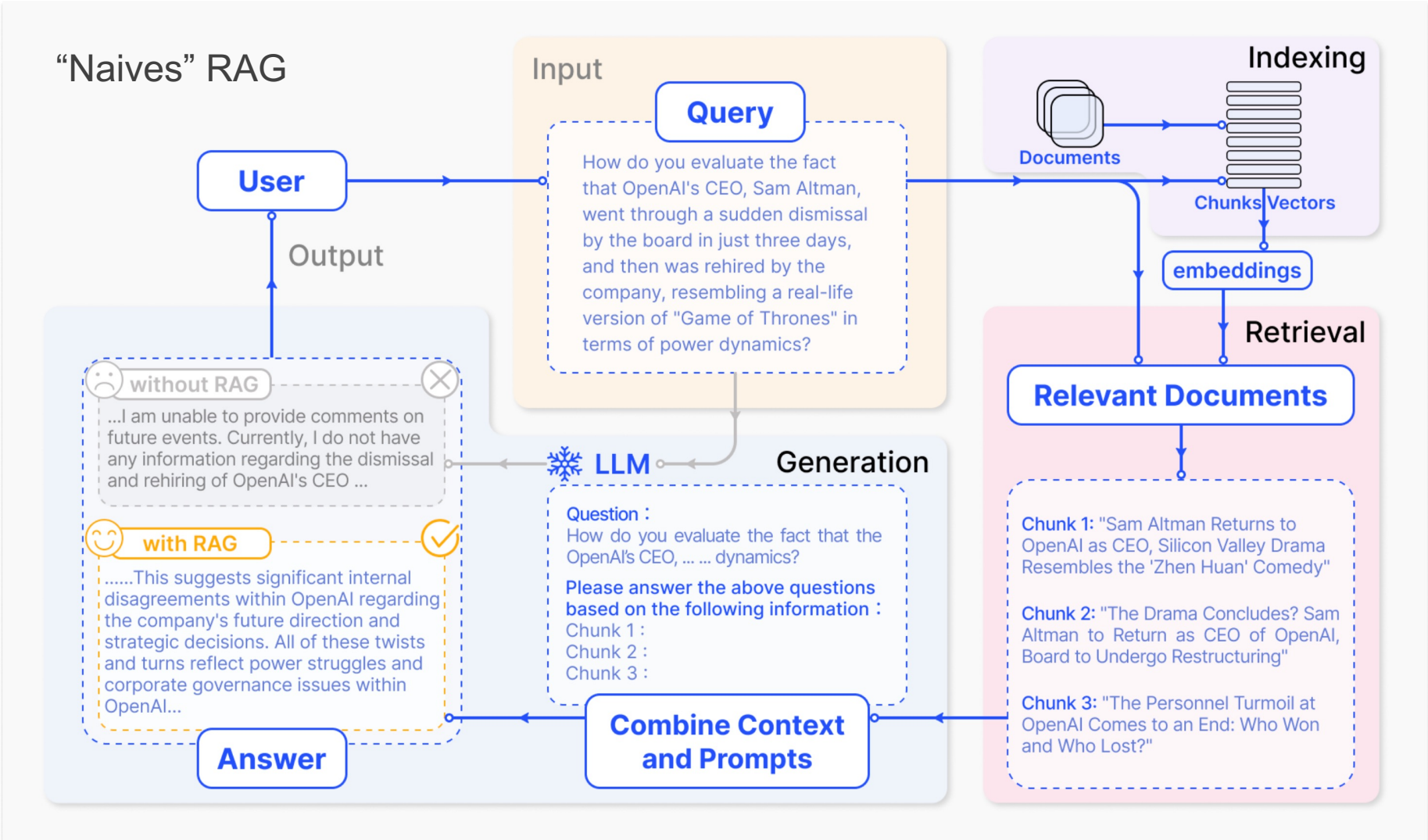
- Was ist RAG?
- Warum wird RAG verwendet?
- Retriever
- Augmentation und Generation
- Implementierungsbeispiel

Was ist RAG?

Das Ziel von RAG ist es, das in Parametern gespeicherte Wissen von Sprachmodellen mit extern abgerufenen, **nicht-parametrisierten Informationen** zu erweitern und so die Leistung des Modells zu steigern.

Der resultierende Informationszustand wird aufgrund dieser Kombination als **semi-parametrisiert** bezeichnet.

Vgl. Hu, Lu 2024, S. 8



Gao et al. 2024, S. 4

Warum wird RAG verwendet?

Informationsqualität

- Vermeidung von Halluzinationen in LLMs
- Nutzung faktisch korrekter Informationen zur Generierung von Antworten
- Steigerung der Robustheit und Genauigkeit von LLMs

Spezifisches Wissen

- Verwendung von unternehmens- und domänenspezifischen Informationen
- Anwendung in wissensintensiven NLP-Bereichen
- z.B. juristische Dokumentenanalyse

Effizient aktualisieren

- Aktualisierung des Wissens ohne kostspieliges Re-Training
- Einfache Integration von Informationen in etablierten Formaten (.pdf, .word etc.)
- Stetige Verwaltung der Informationsbasis

Vgl. Hu, Lu 2024, S. 1 ff.

Retriever

Wie werden die korrekten Informationen im Dokumentenkorpus gefunden?

Sparse Retriever: TF-IDF

Lexikalische Relevanzbewertung nach Worthäufigkeit

T	= Abfragebegriff
P	= Passage
$Korpus$	= alle Passagen

$$\text{Term Frequency (TF)} = \frac{Anzahl\ von\ T}{Anzahl\ von\ Wörter\ in\ P}$$

$$\text{Inverse Document Frequency (IDF)} = \log \frac{Gesamtzahl\ der\ P\ im\ Korpus}{Anzahl\ der\ P\ die\ T\ beeinhalten}$$

TF x IDF ⇨ Häufigkeit des Terms in der Passage (TF) **skaliert** nach allgemeiner Häufigkeit des Terms im Korpus (IDF).

Vgl. Christian et al. 2016, S. 289

Sparse Retriever: BM-25

Erweiterte lexikalische Relevanzbewertung mit TF-IDF

$$\text{BM-25} = \sum_i^n IDF(q_i) \times \frac{TF(q_i,P) \times (k1+1)}{TF(q_i,P) + k1 \times (1 - b + b \times \frac{|P|}{avg|P|})}$$

n	= Anz. Abfragebegriffe
q	= Frage / Abfrage
q_i	= i-te Abfragebegriff
P	= Passage
$\|P\|$	= Passagenlänge
$avg\|P\|$	= ∅, Passagenlänge
b	= frei wählbar (0.75)
$k1$	= frei wählbar

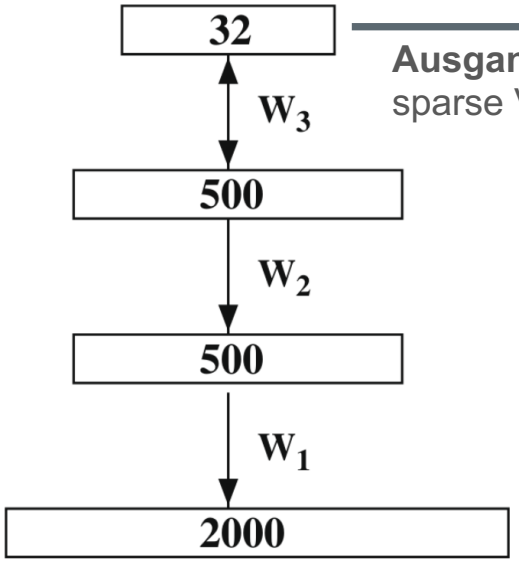
Parameter b beeinflusst die Normalisierung, k1 die Sättigung

BM-25 ⇨ Termhäufigkeit (TF-IDF) modifiziert durch **Sättigung** und **Normalisierung** mit frei wählbaren Parametern k1 und b.

Vgl. Guo et al. 2021, S. 98; Elasticsearch: https://www.elastic.co/de/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables

Sparse Vector Representation

Neuronales Netz

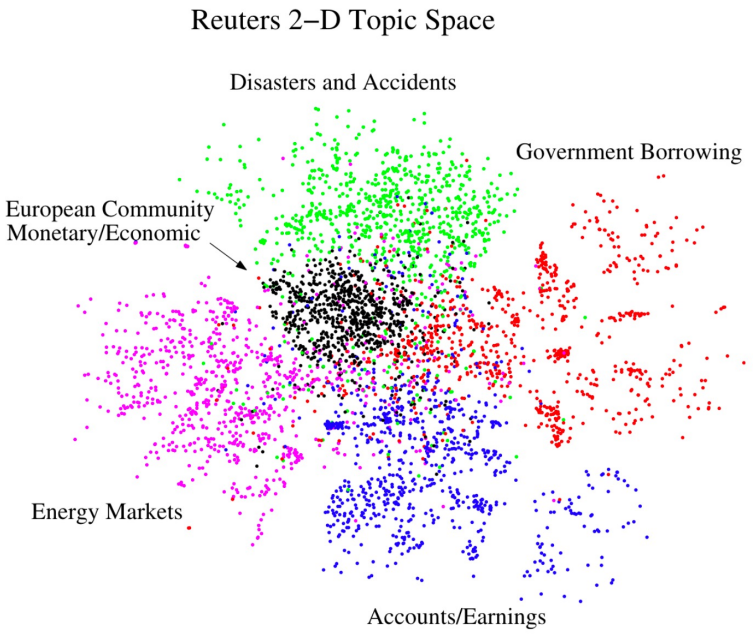


Ausgang: binärer sparse Vektor

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{30} \\ x_{31} \\ x_{32} \end{bmatrix}$$

Eingang: Worthäufigkeit eines Dokuments als Vektor

$x_i \in \{0,1\} \; \forall \; i$



Sparse Vektoren repräsentieren semantische Ähnlichkeit der Dokumente

Vgl. Salakhutdinov, Hinton 2009, S. 970 ff.; Für weiterführende Informationen siehe: Hambarde, Proenca 2023

Dense Retriever: Einführung

Limitationen von Sparse Retrievern:

- Fokussieren sich auf lexikalische Analyse; semantische Zusammenhänge zwischen Textbestandteilen bleiben unberücksichtigt.
  - z.B: “bad guy” vs. “villain”

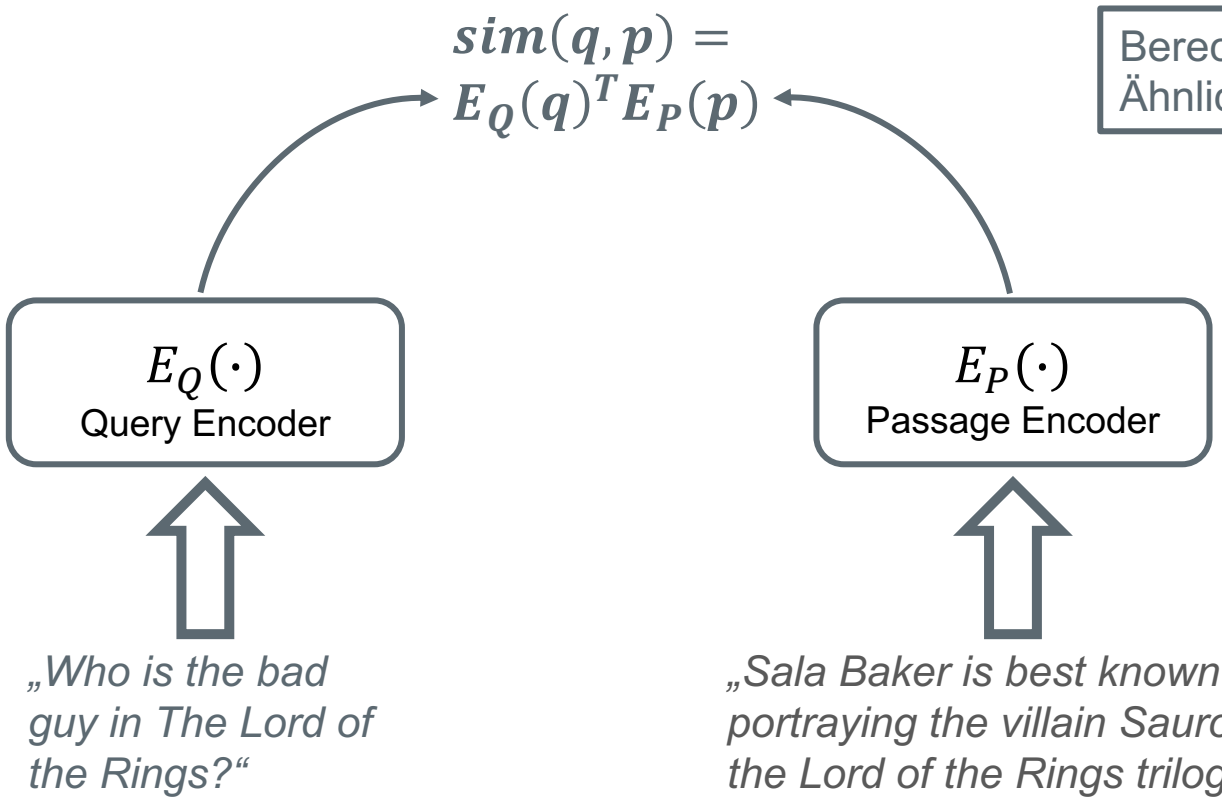
Dense Retriever:

- Verwenden Encoder-Modelle, um kontextualisierte, dichte Embeddings zu erzeugen, die semantische Bedeutungen im d-dimensionalen Raum abbilden
- Ähnliche Fragen und Passagen weisen **größeres** Skalarprodukt auf
- Bieten zusätzliche Flexibilität durch Anpassbarkeit der Encoder-Modelle

→ ”Dense Passage Retriever” im Mittelpunkt der aktuellen Forschung

Vgl. Kharpukhin et al. 2020; Hu, Lu 2024, S. 7; Für weiterführende Informationen siehe: Lewis et al. 2020

Dense Retriever: Dense Passage Retriever (DPR)



Berechnung der semantischen Ähnlichkeit durch das Skalarprodukt

Zwei separate BERT-Encoder zur getrennten Verarbeitung der Frage q und Passage p

Der [CLS]-Token repräsentiert den Output der Encoder

Vgl. Kharpukhin et al. 2020, S. 6770 ff.

Dense Retriever: Dense Passage Retriever (DPR)

Trainingsdaten:

$$D = \left\{\left\langle q_i, p_i^+, p_{i,1}^-, \dots, p_{i,n}^-\right\rangle\right\}_{i=1}^{m}$$

D	= Trainingsdaten
q_i	= i-te Frage
p^+	= p beantwortend
p^-	= p irrelevant
m	= Anzahl Fragen
n	= Anzahl p^-

Loss-Funktion:

$$L\left(q_i, p_i^+, p_{i,1}^-, \dots, p_{i,n}^-\right) = -log\frac{e^{sim\left(q_i, p_i^+\right)}}{e^{sim\left(q_i, p_i^+\right)} + \sum_{j=1}^{n} e^{sim\left(q_i, p_{i,j}^-\right)}}$$

Die Query- und Passagen-Encoder werden **gleichzeitig** trainiert, um das **Skalarprodukt** zwischen Frage und zugehöriger Antwortpassage zu **maximieren** und zwischen Frage und irrelevanten Passagen zu **minimieren**

Vgl. Kharpukhin et al. 2020, S. 6771

Welche Retrieval-Methode sollte eingesetzt werden?

Sparse Retriever

+ Direkte lexikalische Analyse
+ Einfachheit & hohe Effizienz
+ Geeignet für einheitliche Sprachformate (z.B. juristische Texte)
- Keine semantische Betrachtung
- Geringere Genauigkeit

Dense Retriever

+ Abstrakte semantische Analyse
+ Hohe Genauigkeit & Aktualität
+ Flexible Anpassung der Modelle
+ Geeignet für alle Sprachformate
- Keine lexikalische Betrachtung
- Geringere Nachvollziehbarkeit
- Viele Trainingsdaten benötigt

Eine Interpolation von **Dense Retrieval mit lexikalischen Verfahren** verbessert die Retrieval-Genauigkeit signifikant und liefert eine **effektive Kombination der Vorteile** beider Methoden (Wang et al. 2021)

Vgl. Hu, Lu 2024, S. 4 ff.; Vgl. Wang et al. 2021, S. 317 ff.; Vgl. Rosa et al. 2021, S. 1 ff.

Hybrid Retriever

Gewichtete Kombination von DPR und BM-25

$$sim(q,p)^{hyb} = sim(q,p)^{DPR} + \lambda \times BM25(q,p)$$

- Optimierung des Hyperparameters λ **mit annotierten Fragen und Passagen**
- Flexible Kombination von semantischer und lexikalischer Analyse steigert die Retrieval-Genauigkeit

Stufenweises Retrieval-Verfahren ebenfalls möglich:

1. Bewertung aller Passagen mittels BM-25
2. Einsatz von Dense Retrieval Methoden für Top-K Passagen

→ Erhöhte Systemeffizienz

Vgl. Ma et al. 2021 S. 1079

Internet Retriever

Einsatz kommerzieller Suchmaschinen als Retriever

- Übermittlung der Nutzeranfrage von RAG an die Suchmaschine
- Erhalt einer Liste von URLs als Suchergebnis
- Inhalt der Webseiten → Kontext zur Antwortgenerierung

Variationen:

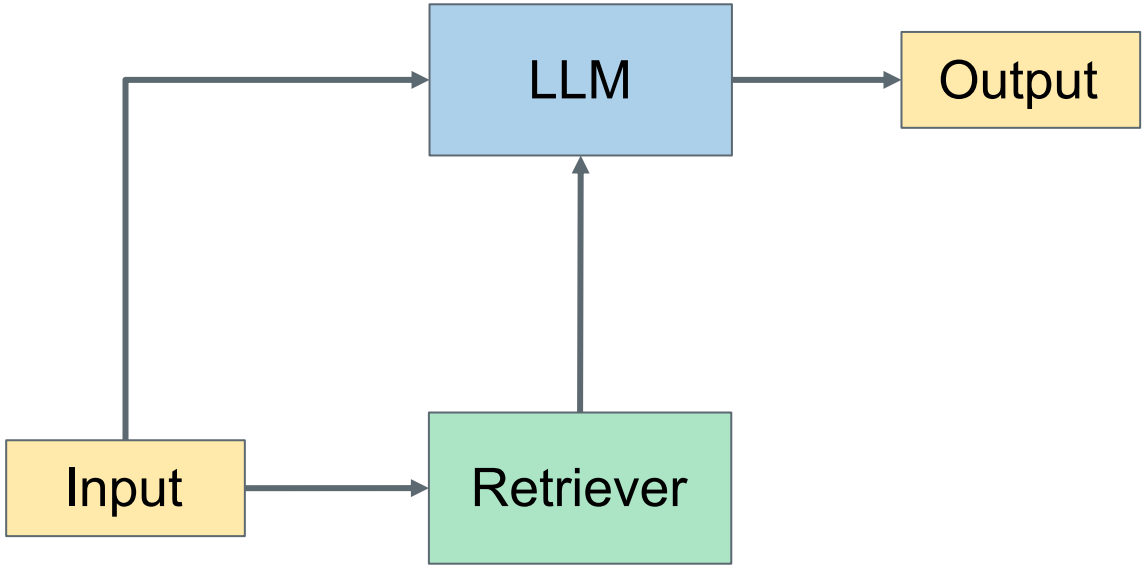
- Verwendung von mehreren Suchmaschinen
- Bevorzugte Gewichtung von URLs aus Wikipedia
- Einsatz von TF-IDF zur Durchführung der Relevanzbewertung
- Kombination von BM-25 mit einem Übersetzungsmodell
- Anwendung von Dense Retrieval Methoden

Vgl. Hu, Lu 2024, S. 8

Augmentation & Generation

Wie interagiert der Retriever mit dem Sprachmodell?

Sequential Single Interaction



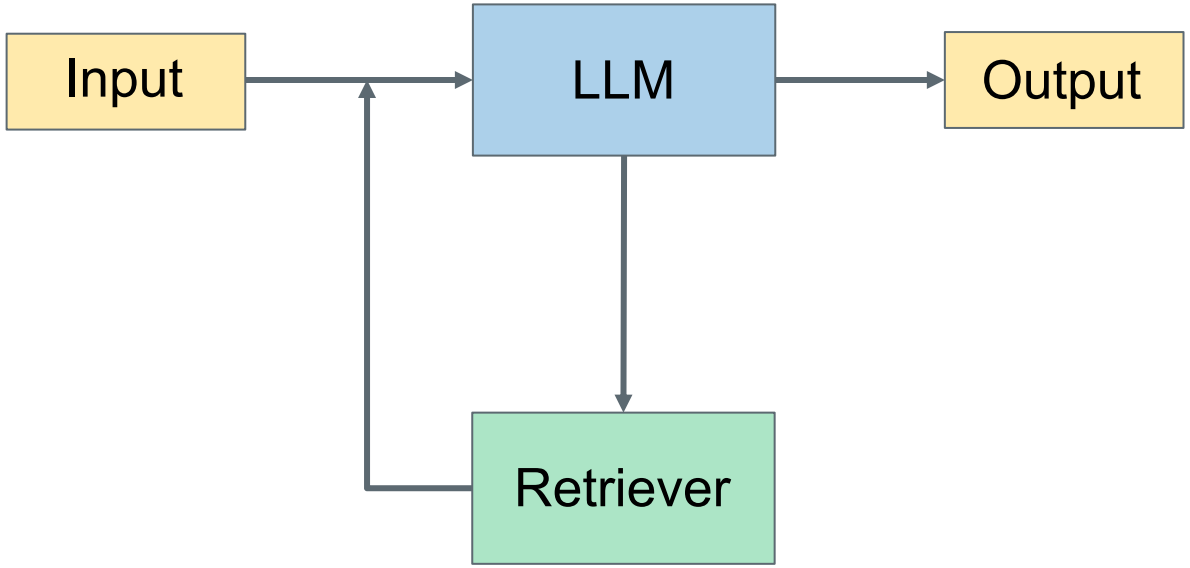
- Einmaliges Retrieval basierend auf Nutzer-Query
- Das LLM erhält die Top-K extrahierten Passagen als Kontext zur Generierung der Antwort

Beispiele:

- „Naives“ RAG
- REALM
- Retro

Vgl. Hu, Lu 2024, S. 3 f.

Sequential Multiple Interactions



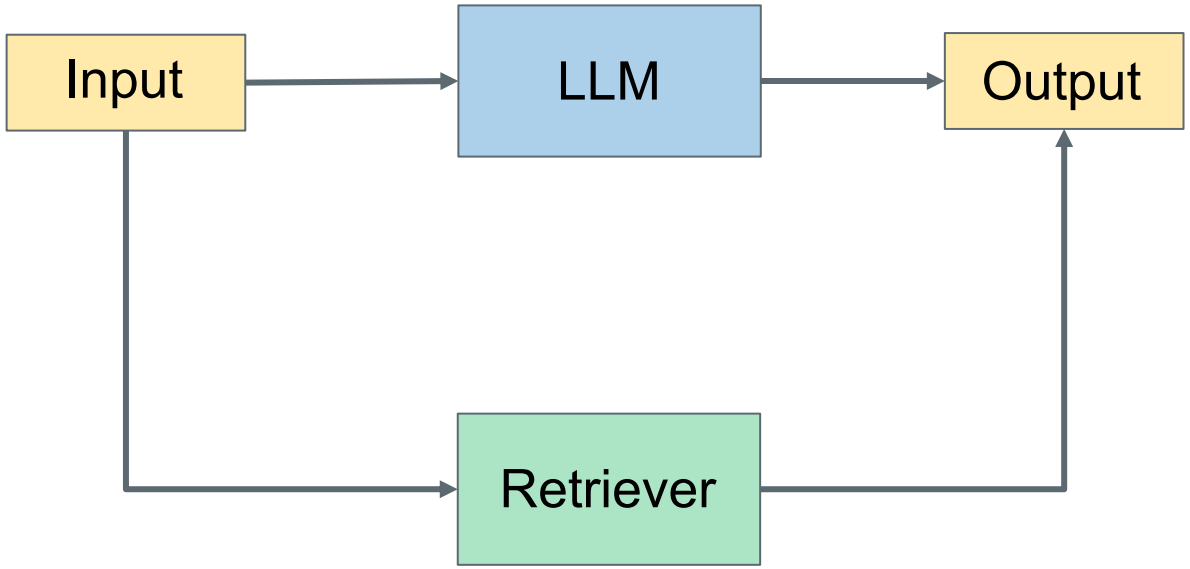
- Für komplexere Antworten
- LLM generiert potenzielle Fortsetzungssätze als Retriever-Query für weiteren Kontext

Beispiele:

- FLARE
- RR
- REFEED

Vgl. Hu, Lu 2024, S. 3 f.; Vgl. Jiang et al. 2023 S.1 f.

Parallel Interaction



- Token-Level Interpolation von Kontext & Antwort
- Retrievte Tokens beeinflussen die Wahrscheinlichkeits verteilung bei der Generierung von Antworttokens

Beispiele:

- KNN-LM
- RETOMATON

Vgl. Hu, Lu 2024, S. 4; Vgl. Khandewal et al. 2019 S.1 f.

Implementierungs-Beispiel

Wie könnte der optimale Interpolationsparameter für das Hybrid Retrieval gefunden werden?

Quellen

- Hu, Lu 2024, “RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing”
- Gao et al. 2024, “Retrieval-Augmented Generation for Large Language Models: A Survey”
- Elasticsearch: “Practical BM25 - Part 2: The BM25 Algorithm and its Variables” (https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables/, Zugriff vom 21.05.2024)
- Christian et al. 2016, “SINGLE DOCUMENT AUTOMATIC TEXT SUMMARIZATION USING TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)”
- Salakhutdinov, Hinton 2009, “Semantic hashing”
- Hambarde, Proenca 2023, “Information Retrieval: Recent Advances and Beyond”
- Kharpukhin et al. 2020, “Dense Passage Retrieval for Open-Domain Question Answering”
- Lewis et al. 2020, “Retrieval-augmented generation for knowledge-intensive NLP tasks”
- Wang et al. 2021, “BERT-based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval”
- Rosa et al. 2021, “Yes, BM25 is a Strong Baseline for Legal Case Retrieval”
- Jiang et al. 2023, “Active Retrieval Augmented Generation”
- Khandelwal et al. 2019, “Generalization through Memorization: Nearest Neighbor Language Models”

Fragen & Diskussion