



## Retrieval-Augmented Generation mit LLMs in finanzbezogenen Frage-Antwort-Systemen

Oğuzhan-Burak Bozkurt

Suggested citation:

Bozkurt, Oğuzhan-Burak. 2025. "Retrieval-Augmented Generation mit LLMs in finanzbezogenen Frage-Antwort-Systemen." Hannover: Hochschule Hannover. <https://doi.org/10.25968/opus-3767>.

### Abstract

Die natürliche Sprachverarbeitung hat durch Fortschritte im maschinellen Lernen, insbesondere durch den Einsatz neuronaler Netze, einen Wandel von klassischen statistischen Methoden hin zu leistungsfähigen Large Language Models wie GPT, LLaMA oder Mistral erfahren. Diese Modelle demonstrieren eindrucksvolle Fähigkeiten in der Textgenerierung und der Bearbeitung komplexer sprachlicher Aufgaben. Mit Zunahme ihrer Integration in unternehmerische Prozesse und Einsatzmöglichkeit im Finanzsektor, rückt jedoch auch ihre Anfälligkeit für Fehler in den Fokus. Ein zentrales Problem stellt die sogenannte Halluzination dar, bei der plausibel klingende, jedoch faktisch unzutreffende Inhalte generiert werden. In regulierten Domänen, wie etwa dem Finanzwesen, können derartige Fehler zu signifikanten Risiken führen. Ein vielversprechender Ansatz zur Reduktion dieser Problematik ist die Retrieval-Augmented Generation, bei der externe Wissensquellen zur Kontexterweiterung zur Inferenzzeit der Modelle herangezogen werden. Die vorliegende Arbeit hat die Intention, den Einsatz von Retrieval-Augmented Generation in domänenspezifischen Frage-Antwort-Systemen im Finanzbereich zu untersuchen. Zu diesem Zweck wurde ein prototypisches System bestehend aus Retrieval- und Generierungskomponente entwickelt, das Finanzberichte börsennotierter Unternehmen als Wissensquelle nutzt. Die Architektur integriert ein Embedding-Modell, eine Vektordatenbank sowie das Modell Mistral-7B. Die Evaluation erfolgt anhand eines Referenzdatensatzes, der Frage-Antwort-Paare und zugehörige Kontexte umfasst. Zur Bewertung werden sowohl klassische Retrieval- und Generierungsmetriken als auch eine sprachmodellgestützte Bewertung mithilfe von GPT-3.5 eingesetzt. Die Ergebnisse zeigen, dass Retrieval-Augmented Generation im Vergleich zum rein generativen Ansatz zu konsistenten Verbesserungen in Bezug auf Antwortqualität, Kontextbezug und Reduktion von Halluzination führt. Gleichzeitig identifiziert die Analyse Schwächen bei der Verarbeitung quantitativer Angaben und künftige Potenziale zur Optimierung durch Reranking oder Fine-Tuning auf Finanzdaten. Der entwickelte Prototyp ist unter <https://github.com/OxBuro/FinLLM-RAG-Eval> verfügbar.

### Terms of use

CC BY-NC-SA 4.0

This document is made available under these conditions:  
**Creative Commons - CC BY-NC-SA - Namensnennung - Nicht kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International**  
For more information see:  
<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de>



# Retrieval-Augmented Generation mit LLMs in finanzbezogenen Frage-Antwort-Systemen

## Bachelorarbeit

vorgelegt von

Oğuzhan-Burak Bozkurt

Hannover, September 2025

**Erstprüfer:** Prof. Dr. Christian Wartena

Hochschule Hannover

Expo Plaza 12

30539 Hannover

E-Mail: christian.wartena@hs-hannover.de

**Zweitprüfer:** Prof. Dr. Peter Wübbelt

Hochschule Hannover

Expo Plaza 12

30539 Hannover

E-Mail: peter.wuebbelt@hs-hannover.de

**Abstract.** Die natürliche Sprachverarbeitung hat durch Fortschritte im maschinellen Lernen, insbesondere durch den Einsatz neuronaler Netze, einen Wandel von klassischen statistischen Methoden hin zu leistungsfähigen Large Language Models wie GPT, LLaMA oder Mistral erfahren. Diese Modelle demonstrieren eindrucksvolle Fähigkeiten in der Textgenerierung und der Bearbeitung komplexer sprachlicher Aufgaben. Mit Zunahme ihrer Integration in unternehmerische Prozesse und Einsatzmöglichkeit im Finanzsektor, rückt jedoch auch ihre Anfälligkeit für Fehler in den Fokus. Ein zentrales Problem stellt die sogenannte Halluzination dar, bei der plausibel klingende, jedoch faktisch unzutreffende Inhalte generiert werden. In regulierten Domänen, wie etwa dem Finanzwesen, können derartige Fehler zu signifikanten Risiken führen. Ein vielversprechender Ansatz zur Reduktion dieser Problematik ist die Retrieval-Augmented Generation, bei der externe Wissensquellen zur Kontexterweiterung zur Inferenzzeit der Modelle herangezogen werden. Die vorliegende Arbeit hat die Intention, den Einsatz von Retrieval-Augmented Generation in domänenspezifischen Frage-Antwort-Systemen im Finanzbereich zu untersuchen. Zu diesem Zweck wurde ein prototypisches System bestehend aus Retrieval- und Generierungskomponente entwickelt, das Finanzberichte börsennotierter Unternehmen als Wissensquelle nutzt. Die Architektur integriert ein Embedding-Modell, eine Vektordatenbank sowie das Modell Mistral-7B. Die Evaluation erfolgt anhand eines Referenzdatensatzes, der Frage-Antwort-Paare und zugehörige Kontexte umfasst. Zur Bewertung werden sowohl klassische Retrieval- und Generierungsmetriken als auch eine sprachmodellgestützte Bewertung mithilfe von GPT-3.5 eingesetzt. Die Ergebnisse zeigen, dass Retrieval-Augmented Generation im Vergleich zum rein generativen Ansatz zu konsistenten Verbesserungen in Bezug auf Antwortqualität, Kontextbezug und Reduktion von Halluzination führt. Gleichzeitig identifiziert die Analyse Schwächen bei der Verarbeitung quantitativer Angaben und künftige Potenziale zur Optimierung durch Reranking oder Fine-Tuning auf Finanzdaten. Der entwickelte Prototyp ist unter <https://github.com/OxBuro/FinLLM-RAG-Eval> verfügbar.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen und verwandte Arbeiten</b>	<b>3</b>
2.1 Transformer-Architektur . . . . .	4
2.1.1 Self-Attention . . . . .	4
2.1.2 Kontextuelle Repräsentation . . . . .	5
2.2 Large Language Models . . . . .	7
2.2.1 Generative Pre-trained Transformers . . . . .	7
2.2.2 Autoregressive Textgenerierung . . . . .	7
2.2.3 Zero-/Few-Shot Learning . . . . .	8
2.2.4 Halluzination in LLMs . . . . .	8
2.3 Embedding-Modelle und Vektorraum . . . . .	9
2.4 Information Retrieval . . . . .	12
2.4.1 Relevanz in IR-Systemen . . . . .	12
2.4.2 Dense Retrieval . . . . .	13
2.5 Retrieval-Augmented Generation . . . . .	14
2.5.1 RAG-Pipeline . . . . .	15
2.5.2 RAG in Frage-Antwort-Systemen . . . . .	17
2.5.3 Evaluierung von RAG-Systemen . . . . .	19
<b>3 Einsatz von LLMs und RAG im Finanzsektor</b>	<b>24</b>
3.1 GenAI in Unternehmen . . . . .	24
3.1.1 Anwendungen von LLMs im Finanzbereich . . . . .	26
3.1.2 Finanzbasierte LLMs . . . . .	28
3.1.3 Herausforderungen beim Einsatz von LLMs im Finanzbereich . . . . .	29
3.2 Regulatorische Rahmenbedingungen: EU AI Act . . . . .	30

3.3	RAG als Lösungsansatz im Finanzkontext . . . . .	31
3.4	Domänenspezifisches RAG und Finanzdatennutzung . . . . .	32
<b>4</b>	<b>Methodik und Implementierung</b>	<b>34</b>
4.1	Datenaufbereitung und QA-Datensatz . . . . .	34
4.2	Retriever Implementierung . . . . .	35
4.2.1	Vektordatenbank: FAISS . . . . .	36
4.2.2	Embedding-Modell: BAAI General Embedding . . . . .	36
4.3	Modellbereitstellung und Inferenzumgebung . . . . .	36
4.3.1	Mistral-7B . . . . .	37
4.3.2	Ollama . . . . .	38
4.4	RAG-Architektur . . . . .	38
<b>5</b>	<b>Evaluation und Analyse</b>	<b>40</b>
5.1	Testaufbau und Experimentdesign . . . . .	40
5.2	Evaluation der Retrievalkomponente . . . . .	43
5.3	Evaluation der Generationkomponente . . . . .	45
5.4	Post-hoc Analyse . . . . .	48
<b>6</b>	<b>Ergebnisse und Diskussion</b>	<b>51</b>
<b>7</b>	<b>Fazit und Ausblick</b>	<b>53</b>
	<b>Literaturverzeichnis</b>	<b>VI</b>

## Abkürzungsverzeichnis

<b>API</b>	Anwendungsprogrammierschnittstelle
<b>BGE</b>	Beijing Academy of Artificial Intelligence General Embedding
<b>CUDA</b>	Compute Unified Device Architecture
<b>DPR</b>	Dense Passage Retrieval
<b>ESG</b>	Environmental, Social, Governance
<b>ESMA</b>	European Securities and Markets Authority
<b>FAIR</b>	Framework for Responsible Adoption of AI in Financial Services
<b>FAISS</b>	Facebook AI Similarity Search
<b>GenAI</b>	Generative Artificial Intelligence
<b>GPT</b>	Generative Pre-trained Transformer
<b>GQA</b>	Grouped Query Attention
<b>IR</b>	Information Retrieval
<b>KI</b>	Künstliche Intelligenz
<b>LCS</b>	Longest Common Subsequence
<b>LLM</b>	Large Language Model
<b>MRR</b>	Mean Reciprocal Rank
<b>NER</b>	Named Entity Recognition
<b>NLG</b>	Natural Language Generation
<b>NLP</b>	Natural Language Processing
<b>ODQA</b>	Open Domain Question Answering
<b>QA</b>	Question Answering
<b>RAG</b>	Retrieval-Augmented Generation
<b>RAGAS</b>	Retrieval-Augmented Generation Assessment Score
<b>RNN</b>	Rekurrentes neuronales Netz
<b>ROUGE</b>	Recall-Oriented Understudy for Gisting Evaluation
<b>SWA</b>	Sliding Window Attention
<b>TF-IDF</b>	Term Frequency-Inverse Document Frequency

# Abbildungsverzeichnis

Abb 2.1:	Dynamische Attention-Gewichtungen (eigene Darstellung angelehnt an Uszkoreit (2017)) . . . . .	6
Abb 2.2:	Klassischer Vektorraum mit Word-Embeddings (eigene Darstellung) . . . . .	11
Abb 2.3:	Datenvorbereitung (Indexing) in RAG-Pipelines (eigene Darstellung) . . . . .	16
Abb 2.4:	Retriever in RAG-Pipelines (eigene Darstellung) . . . . .	16
Abb 2.5:	Generator in RAG-Pipelines (eigene Darstellung) . . . . .	17
Abb 3.1:	Organisationen, die GenAI in mindestens einer Funktion einsetzen. Quelle: Singla u. a. (2025, S. 15) . . . . .	25
Abb 3.2:	Von Unternehmen adressierte GenAI-bezogene Risiken. Quelle: Singla u. a. (2025, S. 6) . . . . .	26
Abb 4.1:	Sliding Window Attention (SWA) (in Jiang u. a. (2023), Abb. 1) . . . . .	37
Abb 4.2:	RAG-Architektur (eigene Darstellung) . . . . .	39
Abb 5.1:	Precision-Recall-Kurve für verschiedene Top- $k$ Werte . . . . .	44

## Tabellenverzeichnis

Tabelle 2.1	Top 5 MTEB-Modelle nach Gesamtperformance (Borda-Ranking) . . . . .	10
Tabelle 5.1	Retrievalevaluierung für verschiedene Top- $k$ Werte . . .	43
Tabelle 5.2	Standardabweichung der semantischen Ähnlichkeit pro $k$	44
Tabelle 5.3	Evaluation der Generierungsqualität . . . . .	45
Tabelle 5.4	Medianvergleich der Generierungsqualität zwischen LLM und RAG . . . . .	46
Tabelle 5.5	Deskriptive Statistik der Faithfulness . . . . .	47
Tabelle 5.6	Kategoriale Verteilung der Faithfulness-Scores . . . . .	47

# 1 Einleitung

Die jüngsten Entwicklungen im Bereich des maschinellen Lernens und der künstlichen neuronalen Netze haben zu einer grundlegenden Transformation der natürlichen Sprachverarbeitung (Natural Language Processing, NLP) geführt. Einen entscheidenden Meilenstein stellte die Einführung der Transformer-Architektur durch Vaswani u. a. (2017) dar, die auf dem sogenannten *Attention Mechanism* basiert und seither als technologische Grundlage moderner Sprachmodelle gilt. Darauf aufbauend wurden leistungsfähige Large Language Models (LLMs) entwickelt, die auf umfangreichen Textkorpora vortrainiert werden und in einer Vielzahl von sprachverarbeitenden Aufgaben herausragende Ergebnisse erzielen. Sie werden als zentrales Werkzeug der generativen künstlichen Intelligenz (GenAI) bezeichnet (vgl. Jurafsky und Martin 2025, S. 204). Zu den bekanntesten Modellen zählen GPT-3 (vgl. Brown u. a. 2020), LLaMA (vgl. Touvron u. a. 2023) und Mistral-7B (vgl. Jiang u. a. 2023).

Insbesondere im Finanzsektor besteht ein hohes Interesse an der Integration solcher Modelle, da sie das Potenzial bieten, datengetriebene Prozesse effizienter zu gestalten. Gleichzeitig steigen die Anforderungen an Genauigkeit, Nachvollziehbarkeit und regulatorische Konformität. Trotz ihrer Leistungsfähigkeit weisen LLMs jedoch inhärente Schwächen auf. Eine zentrale Problematik ist die sogenannte Halluzination - die Erzeugung plausibel klingender, aber faktisch falscher Inhalte (vgl. Ji u. a. 2023). In regulierten und präzisionskritischen Domänen wie dem Finanzwesen können solche Fehler schwerwiegende Konsequenzen haben und stellen eine wesentliche Hürde für den vertrauenswürdigen Einsatz dar.

Ein vielversprechender Ansatz zur Minderung dieses Problems ist die Retrieval-Augmented Generation (RAG). Sie erweitern die modellbasierte Sprachgenerierung durch die Anbindung externer Wissensquellen, auf die zur Inferenzzeit zugegriffen wird. Dadurch erhalten Sprachmodelle Zugriff auf aktuelle und domänenspezifische Informationen, die über ihr ursprüngliches Trainingswissen hinausgehen (vgl. Lewis u. a. 2020).

Diese Arbeit untersucht die Eignung von RAG zur Verbesserung der Antwortqualität in finanzspezifischen Frage-Antwort-Systemen. Im Zentrum stehen dabei die Aspekte Antwortqualität, Kontexttreue und Halluzinationsvermeidung. Daraus ergibt sich folgende zentrale Forschungsfrage:

*Wie wirkt sich der Einsatz von Retrieval-Augmented Generation auf die Antwortqualität von LLMs in domänenspezifischen Frage-Antwort-Systemen im Finanzbereich aus - insbesondere im Hinblick auf Genauigkeit, Kontextrelevanz und die Reduktion von Halluzinationen?*

Zur Beantwortung dieser Frage wurde ein prototypisches System entwickelt, das Finanzberichte börsennotierter Unternehmen als Wissensquelle nutzt. Es basiert auf einer lokalen Retrieval- und Inferenzumgebung, bestehend aus einem Embedding-Modell, einer Vektordatenbank sowie dem Sprachmodell Mistral-7B. Die Evaluation erfolgt anhand eines Vergleichsdesigns und der Hinzunahme eines *Ground Truth* Datensatzes mit rund 7,000 Frage-Antwort-Paaren sowie zugehörigem Kontext. Zur Bewertung werden sowohl klassische Retrieval- und Generierungsmetriken als auch ein modellbasiertes Bewertungsschema auf Basis von GPT-3.5 eingesetzt.

Der Aufbau der Arbeit gliedert sich wie folgt:

- **Kapitel 2** vermittelt relevante theoretische Grundlagen und stellt verwandte Arbeiten vor. Es behandelt zentrale Konzepte wie die Transformer-Architektur, Large Language Models, Embedding-Modelle, Information Retrieval, Retrieval-Augmented Generation und die Evaluierung von Retrieval- und Generierungssystemen.
- **Kapitel 3** gibt einen Überblick zu Anwendungsszenarien sowie regulatorische und technologische Herausforderungen beim Einsatz von generativer künstlicher Intelligenz in Unternehmen und speziell im Finanzbereich. Hier wird die Retrieval-Augmented Generation als Lösungsansatz für Herausforderungen im Finanzbereich diskutiert.
- **Kapitel 4** beschreibt die Methodik und technische Umsetzung des im Rahmen dieser Arbeit entwickelten Systems: einschließlich Datenvorverarbeitung, Retriever Implementierung, Modellbereitstellung und Systemarchitektur.
- **Kapitel 5** widmet sich dem Testaufbau und der Evaluation mittels der in Kapitel 2 vermittelten Evaluierungsansätze. Die Ergebnisse werden hier mit kurzen Analysen dargestellt.
- **Kapitel 6** diskutiert die Ergebnisse der Evaluation im Hinblick auf die Forschungsfrage.
- **Kapitel 7** fasst die zentralen Erkenntnisse zusammen und gibt einen Ausblick auf weiterführende Forschung und mögliche Optimierungen.

## 2 Grundlagen und verwandte Arbeiten

Dieses Kapitel stellt die theoretischen Grundlagen und relevanten Forschungsansätze vor, die für das Verständnis sowie die Umsetzung von RAG zentral sind. Ausgangspunkt ist die Transformer-Architektur, deren Mechanismen wie Self-Attention und kontextuelle Repräsentationen die Basis moderner Sprachverarbeitungssysteme bilden. Darauf aufbauend wird das Konzept von LLMs einschließlich ihrer Fähigkeiten und typischen Herausforderungen wie der Halluzination erläutert. Im Anschluss folgt eine Einführung in Embedding-Modelle und den ihnen zugrunde liegenden Vektorraum, der für semantische Repräsentationen wesentlich ist. Weitere Abschnitte behandeln zentrale Konzepte des Information Retrieval (IR), insbesondere Ähnlichkeitssuche mit Dense Retrieval und gängige Evaluationsmetriken zur Leistungsbewertung. Den Abschluss bildet eine vertiefte Darstellung des RAG-Ansatzes, einschließlich seiner Systemarchitektur, der Anwendung in Frage-Antwort-Systemen sowie geeigneter Evaluationsverfahren zur Beurteilung der Antwortqualität.

Eine tiefgehende mathematische oder systemnahe Analyse, wie etwa zur Parallelisierung oder zu verteilten Architekturen, liegt außerhalb des Rahmens dieser Bachelorarbeit. Diese Aspekte setzen in der Regel weiterführende Kenntnisse in parallelen Systemarchitekturen und spezialisierte Vertiefung in Machine- und Deep Learning voraus, wie sie typischerweise erst im Masterstudium vermittelt werden. Ein gewisses Maß an Grundlagenwissen in den Bereichen maschinelles Lernen und natürliche Sprachverarbeitung ist jedoch unerlässlich, um die in dieser Arbeit eingesetzten Methoden angemessen nachvollziehen zu können. Wartena (2024) betont, dass ein fundiertes Verständnis von LLMs insbesondere durch Konzepte wie Verteilungssemantik, statistische Sprachmodellierung, Embedding-Verfahren und ein grundlegendes Verständnis der Modellarchitektur gefördert wird. Darüber hinaus sind Kenntnisse über Vektorraummodelle, Repräsentationsformen und Evaluationsmethoden essenziell, um modellbasierte Systeme einordnen und deren Leistungsfähigkeit fundiert bewerten zu können. Diese Perspektive unterstreicht die in dieser Arbeit gewählte abstrahierende Betrachtung zentraler Komponenten moderner NLP-Systeme.

## 2.1 Transformer-Architektur

Die Transformer-Architektur, die von Vaswani u. a. (2017) vorgestellt wurde, bildet die Grundlage moderner LLMs. Im Gegensatz zu früheren rekurrenten neuronalen Netzen (RNNs), die Eingaben sequenziell verarbeiten, erlaubt der Transformer die parallele Verarbeitung ganzer Tokensequenzen. Diese Parallelisierung wird durch den zentralen *Attention Mechanism* ermöglicht, der es erlaubt, während der Verarbeitung - der sogenannten Inferenzphase - für ein gegebenes Token die Relevanz aller anderen Token in einer Sequenz zu bewerten. So kann das Modell kontextuelle Zusammenhänge erfassen, etwa ob sich ein Pronomen auf das Subjekt am Satzanfang oder auf einen späteren Ausdruck bezieht. Im Kern besteht die Transformer-Architektur aus einem Encoder-Decoder-Aufbau:

- **Encoder-Komponenten** verarbeiten den Eingabetext und erzeugen eine kontextualisierte Repräsentation jedes Token. Modelle, die ausschließlich Encoder verwenden, etwa BERT (vgl. Devlin u. a. 2019) oder sogenannte Sentence Transformers, sind auf Aufgaben wie Klassifikation, Embedding-Erzeugung oder semantische Suche spezialisiert.
- **Decoder-Komponenten** generieren autoregressiv Text, wobei jedes neue Token auf Basis der bisherigen Ausgaben vorhergesagt wird. Modelle wie GPT-3 (vgl. Brown u. a. 2020) basieren ausschließlich auf Decodern und eignen sich besonders gut für Aufgaben zur Textgenerierung oder in Frage-Antwort-Systeme, wie etwa Chatbots.

Klassische Transformer-Modelle nutzen für Aufgaben wie maschinelle Übersetzung beide Komponenten: Der Encoder verarbeitet den Quelltext, während der Decoder Text in der Zielsprache unter Einbeziehung der Encoder-Ausgabe erzeugt (vgl. Vaswani u. a. 2017, 8f.).

### 2.1.1 Self-Attention

Die zentrale Operation aller Attention-Mechanismen im Transformer ist die sogenannte *Scaled Dot-Product Attention*, die insbesondere im Rahmen der *Self-Attention* eingesetzt wird, also immer dann, wenn ein Token im Verhältnis zu anderen Token derselben Eingabesequenz bewertet wird. Formal wird der *Scaled Dot-Product Attention* wie folgt definiert:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (2.1)$$

Dabei sind  $Q$  (Queries),  $K$  (Keys) und  $V$  (Values) Matrizen, die aus den Eingaberepräsentationen berechnet werden. Der skalierte Skalarprodukt-Term  $\frac{QK^\top}{\sqrt{d_k}}$  misst die Ähnlichkeit zwischen Query und Key, bevor er durch die Softmax-Funktion normalisiert wird, um Gewichtungen für die Werte  $V$  zu erzeugen (vgl. Vaswani u. a. 2017, S. 4).

Die Self-Attention erlaubt es jedem Token, kontextuelle Informationen von allen anderen Tokens der Eingabesequenz zu aggregieren. Dies geschieht durch gewichtete Kombination der Values  $V$ , wobei die Gewichte durch Ähnlichkeiten zwischen Queries und Keys bestimmt werden. Dadurch können insbesondere semantische Abhängigkeiten über größere Distanzen hinweg modelliert werden.

### 2.1.2 Kontextuelle Repräsentation

Die Fähigkeit zur dynamischen Kontextbewertung ermöglicht es Transformer-Modellen, kontextuelle Wortrepräsentationen zu erzeugen, bei denen dasselbe Wort je nach Kontext unterschiedliche Vektordarstellungen erhält (vgl. Jurafsky und Martin 2025, S. 186). Die Komplexität dieser kontextuellen Disambiguierung wird besonders bei mehrdeutigen Pronomen deutlich. Ein typisches Beispiel für den Nutzen dieses Kontexts zeigt sich in der folgenden Satzstruktur zweier unterschiedlicher Tokensequenzen:

Die Katze schlenderte nicht durch die Gasse, weil sie zu müde war.

Die Katze schlenderte nicht durch die Gasse, weil sie zu voll war.

Beide Sätze sind strukturell identisch, unterscheiden sich aber im semantischen Bezug des Pronomens „sie“. In der ersten Variante bezieht sich „sie“ auf die Katze, in der zweiten auf die Gasse. Ein LLM, das autoregressiv Text generiert, muss diesen Bezug im laufenden Generationsprozess aus dem vorherigen Kontext erschließen. Der Attention-Mechanismus erlaubt es dem Modell, relevante Token im vorherigen Kontext unterschiedlich stark zu gewichten, also je nachdem, ob es sich bei „sie“ um die Katze oder die Gasse handelt. In der Visualisierung der Attention-Gewichtung in einer Vektordarstellung (Abb. 2.1) wird diese unterschiedliche Aufmerksamkeitsverteilung exemplarisch dargestellt.

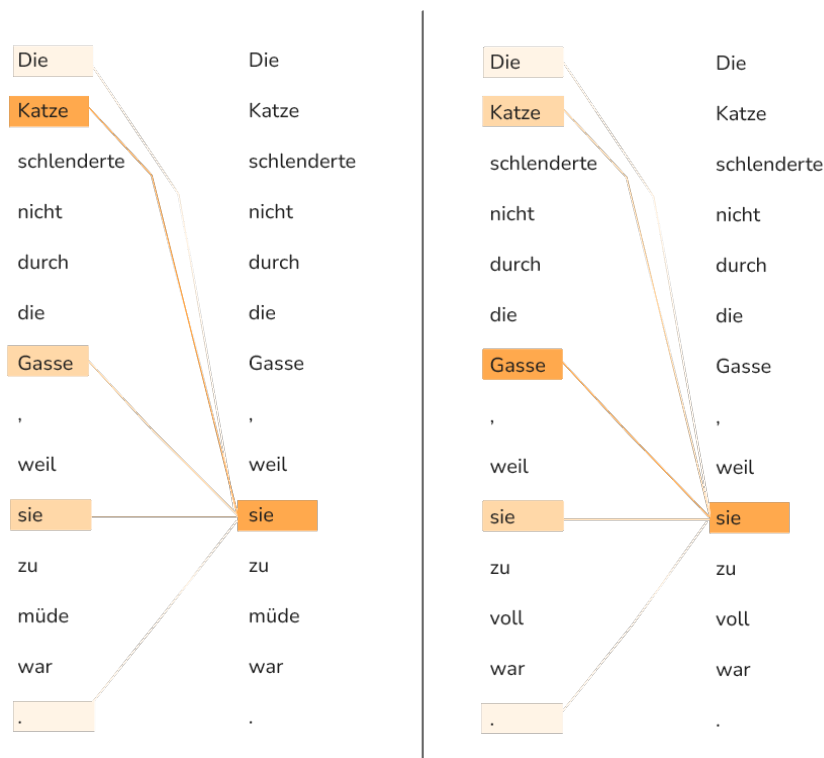


Abb. 2.1: Dynamische Attention-Gewichtungen (eigene Darstellung angelehnt an Uszkoreit (2017))

Dabei ist jedoch auch eine alternative Lesart denkbar, in der sich „voll“ nicht auf räumliche Überfüllung, sondern metaphorisch auf einen Zustand der Katze bezieht, etwa im Sinne von „gesättigt“ oder „überfressen“. Diese Art von lexikalischer Mehrdeutigkeit (Polysemie) stellte eine Herausforderung dar, da die korrekte Disambiguierung nicht nur syntaktische, sondern auch semantische Informationen über den Konkretheitsgrad und typische Verwendungskontexte der beteiligten Wörter voraussetzt. Konkretheit lässt sich als semantische Dimension im Vektorraum von Worteinbettungen identifizieren. Konkrete und abstrakte Wörter nehmen dabei gegenüber einer Konkretheitsrichtung im semantischen Raum unterschiedliche Positionen ein. Gerade bei polysemen Wörtern, die sowohl in konkreten als auch abstrakten Kontexten auftreten (wie „voll“ oder „leer“), ist es für Sprachmodelle schwierig, aus einem statischen Embedding auf die aktuelle Lesart zu schließen (vgl. Wartena 2022). Erst durch kontextuelle Repräsentationen und dynamische Aufmerksamkeit können Sprachmodelle unterschiedliche Interpretationen erzeugen. Der Transformer und seine inhärenten Mechanismen adressieren diese Herausforderungen.

## 2.2 Large Language Models

Aufbauend auf der zuvor erläuterten Transformer-Architektur behandelt dieser Abschnitt sogenannte Large Language Models (LLMs), die auf großen Datenmengen trainiert wurden. LLMs zeichnen sich durch ihre Fähigkeit aus, menschenähnlichen Text zu verstehen und zu generieren, was sie zu vielseitigen Werkzeugen für eine breite Palette von Aufgaben in der natürlichen Sprachverarbeitung macht. Im folgenden werden bekannte Sprachmodelle vorgestellt und ihre wichtigsten Funktionen erläutert. Abschließend wird eine zentrale Herausforderung und die inhärente Limitation von LLMs diskutiert: das Auftreten von Halluzinationen.

### 2.2.1 Generative Pre-trained Transformers

Die Generative Pre-trained Transformers (GPT) von OpenAI stehen exemplarisch für die Leistungsfähigkeit und Entwicklung großer Sprachmodelle. Sie basieren auf der Decoder-Komponente der Transformer-Architektur und sind speziell für die Textgenerierung optimiert (vgl. Radford u. a. 2018, S. 5).

Die Entwicklung begann mit vergleichsweise kleinen Modellen und erreichte mit GPT-3 einen markanten Meilenstein: Mit 175 Milliarden Parametern markierte es einen signifikanten Fortschritt sowohl in Bezug auf Modellgröße als auch auf Ausdrucksfähigkeit. GPT-3 war in der Lage, kohärente und kontextuell stimmige Texte zu erzeugen und Aufgaben von kreativem Schreiben bis hin zur Beantwortung komplexer Fragen zu lösen (vgl. Brown u. a. 2020).

Parallel dazu zeigte die Forschung, dass auch kleinere, effizientere Modelle wie die LLaMA-Reihe von Meta konkurrenzfähige Leistungen erzielen können – vorausgesetzt, sie werden auf umfangreichen und qualitativ hochwertigen Datensätzen trainiert (vgl. Touvron u. a. 2023).

### 2.2.2 Autoregressive Textgenerierung

Die autoregressive Textgenerierung bestimmt die Art und Weise, wie moderne decoder-basierte LLMs wie GPT Texte produzieren. Bei dieser Vorgehensweise wird Text sequenziell von links nach rechts generiert: Das Modell sagt jeweils das nächste Token voraus, basierend auf allen vorherigen Token. Formal wird zu jedem Zeitpunkt  $t$  die Wahrscheinlichkeitsverteilung  $P(w_t | w_1, \dots, w_{t-1})$  über das nächste Wort berechnet und aus dieser Verteilung ein Token generiert (vgl. Jurafsky und Martin 2025, S. 167f.).

Die Kombination aus Self-Attention (Abschnitt 2.1.1) für kontextuelle Bewertung und autoregressiver Generierung für sequenzielle Textproduktion bildet

das Fundament der Funktionsweise moderner LLMs und erklärt deren Fähigkeit, kohärente und kontextbezogene Texte zu generieren.

### 2.2.3 Zero-/Few-Shot Learning

Ein wesentliches Merkmal und ein entscheidender Vorteil moderner LLMs ist ihre Fähigkeit zum Few-Shot Learning. Im Gegensatz zu traditionellen Machine Learning Modellen, die für jede spezifische Aufgabe Tausende von gelabelten Beispielen für das Fine-Tuning benötigen, können LLMs neue Aufgaben mit nur einer Handvoll von Beispielen oder sogar gänzlich ohne explizite Beispiele (Zero-Shot Learning) ausführen. Diese Fähigkeit wird dadurch erreicht, dass die Beispiele oder Anweisungen direkt in den Prompt integriert werden, also den Texteingaben, die an das Modell gesendet werden. Das Modell nutzt sein während des Vortrainings erworbenes Wissen um das Muster der gestellten Aufgabe aus den wenigen gegebenen Demonstrationen abzuleiten und auf neue, ähnliche Situationen zu übertragen (vgl. Brown u. a. 2020, S. 6f.).

### 2.2.4 Halluzination in LLMs

Die Halluzination in LLMs stellt eine der gravierendsten Herausforderungen dar. Im wesentlichen werden Halluzinationen als die Generierung plausibler, aber faktischer inkorrektur Inhalte beschreiben. Dieses Phänomen bringt angesichts der offenen und universellen Natur von LLMs einzigartige Herausforderungen mit sich, die über die Probleme früherer aufgabenspezifischer Modelle hinausgehen (vgl. Huang u. a. 2025).

In einer Übersichtsstudie kategorisieren Ji u. a. (2023) die Ursachen von Halluzinationen in LLMs in verschiedene Bereiche, die sich gegenseitig bedingen und verstärken können:

- **Datenbezogene Ursachen:** Eine wesentliche Ursache liegt in den Trainingsdaten selbst. Oftmals weisen die zur Trainingsphase genutzten Quell- und Referenzdaten Divergenzen auf. Dies kann durch heuristische Datenerfassung oder durch die inhärente Natur bestimmter natürlicher Sprachgenerierungsaufgaben bedingt sein. Wenn Modelle auf solchen inkonsistenten Daten trainiert werden, entwickeln sie eine Tendenz, nicht verifizierbare oder nicht quellgetreue Inhalte zu generieren. Dies tritt beispielsweise auf, wenn der Referenztext Informationen enthält, die nicht direkt aus der bereitgestellten Quelle abgeleitet werden können.
- **Fehlerhaftes parametrisches Wissen:** LLMs speichern während ihres umfangreichen Vortrainings eine enorme Menge an implizitem Wissen

in ihren Parametern. Wenn dieses gespeicherte Wissen ungenau oder veraltet ist, kann dies zu faktischen Halluzinationen führen, selbst wenn der Eingabekontext korrekt ist.

- **Inkorrekter Attention-Mechanismus:** Die Transformer-Architektur basiert maßgeblich auf dem Attention-Mechanismus, der es dem Modell ermöglicht, relevante Teile des Eingabekontextes zu gewichten (siehe Abschnitt 2.1.1). Fehlfunktionen oder Ungenauigkeiten in diesem Mechanismus können dazu führen, dass das Modell irrelevante oder sogar falsche Informationen aus dem Input übermäßig stark berücksichtigt oder wichtige Details ignoriert, was zu halluzinierten Ausgaben führt.
- **Ungeeignete Trainingsstrategie:** Die Art und Weise, wie ein LLM trainiert wird, hat einen erheblichen Einfluss auf seine Neigung zu halluzinieren. Suboptimale Trainingsmethoden, wie beispielsweise unzureichendes Fine-Tuning oder Probleme bei der Formulierung der Trainingsziele, können die Anfälligkeit des Modells für die Generierung unzutreffender Inhalte erhöhen.
- **Inferenz-Exposure-Bias:** Dieser Bias entsteht, wenn es eine Diskrepanz zwischen den Bedingungen während der Trainingsphase und denen während der Inferenzphase gibt. Während des Trainings kann das Modell Zugriff auf die korrekten vorherigen Token haben, wohingegen bei der Inferenz das Modell seine eigenen, potenziell fehlerhaften Vorhersagen als nächsten Input nutzen muss. Diese kumulativen Fehler können zur Generierung von Halluzinationen beitragen.

Eine weitere Perspektive auf die Definition von Halluzinationen, insbesondere im Kontext von LLMs, wird von Y. Zhang u. a. (2023) beleuchtet. Sie definieren Halluzinationen als Inhalte, die vom LLMs generiert werden und vom ursprünglichen Benutzereingabe abweichen, bereits generierten Kontext widersprechen oder nicht mit etabliertem Weltwissen übereinstimmen.

Die Reduktion von Halluzinationen stellt über die große Anzahl möglicher Fehlerursachen hinweg eine der größten Herausforderungen für die breite Akzeptanz und den sicheren Einsatz von LLMs dar, insbesondere in wissensintensiven oder sicherheitskritischen Domänen und Anwendungen.

## 2.3 Embedding-Modelle und Vektorraum

Embedding-Modelle bilden die Grundlage für die Verarbeitung natürlicher Sprache in LLMs, indem sie Wörter, Phrasen oder Dokumente in eine Wort- oder Satzeinbettung, sogenannte Word- oder Sentence-Embeddings, überführen. Diese dichten, hochdimensionalen Vektorrepräsentationen von Segmenten

werden in einem semantischen Vektorraum so angeordnet, dass ihre Abstände und Richtungen syntaktische oder semantische Ähnlichkeiten widerspiegeln (vgl. Jurafsky und Martin 2025, S. 105ff.).

Frühere Modelle wie Word2Vec (vgl. Mikolov u. a. 2013) oder GloVe (vgl. Pennington, Socher und Manning 2014) erzeugen für jedes Wort eine feste Repräsentation auf Basis von Kookkurrenzstatistiken. Moderne, kontextbasierte Varianten wie BERT oder Sentence-BERT (vgl. Reimers und Gurevych 2019) nutzen hingegen Transformer-Architekturen, um kontextabhängige Embeddings zu erzeugen, also Vektoren, die sich je nach Umgebung des Wortes oder der Phrasen verändern. Solche Embeddings eignen sich besonders gut für semantische Suchaufgaben und werden vor allem für das Dense Retrieval eingesetzt (siehe Abschnitt 2.4.2). Die Leistungsfähigkeit von Embedding-Modellen lässt sich durch Benchmarks wie MTEB oder MMTEB systematisch vergleichen. Die nachfolgende Tabelle 2.1 listet die fünf führenden Modelle im MTEB-Leaderboard (Stand: Juli 2025) auf, welche sich durch ihre ausgewogene Performance über verschiedene Aufgabenbereiche auszeichnen (vgl. Enevoldsen u. a. 2025).

<b>Rang</b>	<b>Modell</b>	<b>Param.</b>	<b>Dim.</b>	<b>Retrieval</b>
1	gemini-embedding-001	Unknown	3072	67.71
2	Qwen3-Embedding-8B	7B	4096	70.88
3	Qwen3-Embedding-4B	4B	2560	69.60
4	Qwen3-Embedding-0.6B	595M	1024	64.65
5	Linq-Embed-Mistral	7B	4096	58.69

Tabelle 2.1: Top 5 MTEB-Modelle nach Gesamtperformance (Borda-Ranking)

Im Vektorraum korreliert die Nähe zweier Vektoren mit der semantischen Ähnlichkeit der repräsentierten Texte. Beispielsweise liegen die Vektoren für Katze und Maus näher beieinander als die für Katze und Auto (siehe Abb. 2.2).

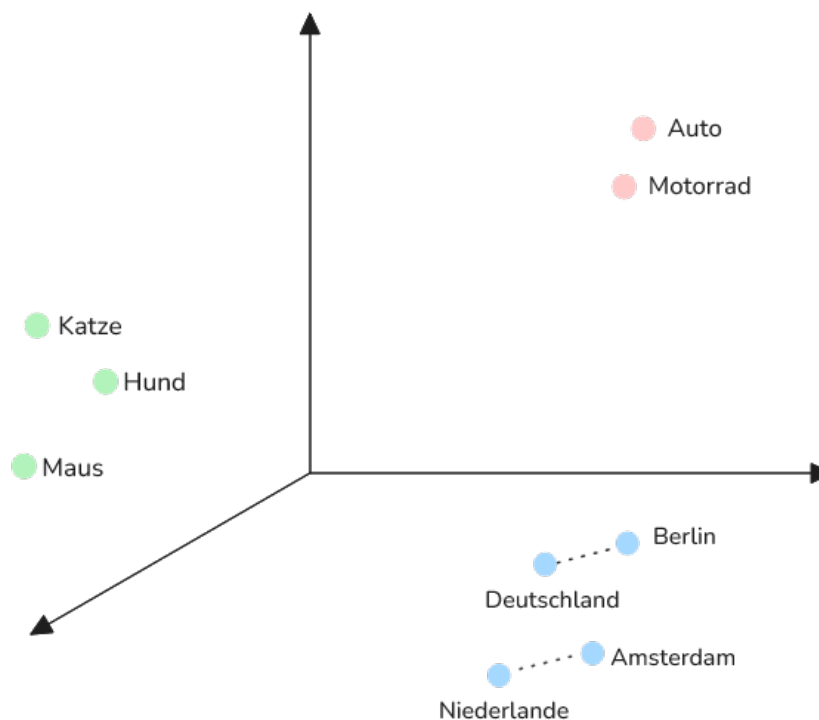


Abb. 2.2: Klassischer Vektorraum mit Word-Embeddings (eigene Darstellung)

In klassischen Word-Vektorrepräsentationen lassen sich außerdem relationale Zusammenhänge wie die Beziehung zwischen Land und Hauptstadt sowie komplexere Vektoroperationen ausdrücken, wie etwa:

$$\vec{\text{Berlin}} - \vec{\text{Deutschland}} + \vec{\text{Niederlande}} \approx \vec{\text{Amsterdam}}$$

Die Ähnlichkeit zwischen Vektoren wird üblicherweise durch die Kosinusähnlichkeit gemessen, definiert als:

$$\text{Cosine Similarity}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|} \quad (2.2)$$

wobei

- $\vec{A} \cdot \vec{B}$  das Skalarprodukt der beiden Vektoren bezeichnet,
- $\|\vec{A}\|$  und  $\|\vec{B}\|$  die euklidischen Normen (Längen) der Vektoren  $\vec{A}$  und  $\vec{B}$  darstellen.

Der Wert der Kosinusähnlichkeit liegt im Bereich  $[-1, 1]$ . Ein Wert von 1 bedeutet, dass die Vektoren in dieselbe Richtung zeigen (maximale Ähnlichkeit), ein Wert von 0 kennzeichnet orthogonale Vektoren (keine Ähnlichkeit) und ein Wert von  $-1$  bedeutet, dass die Vektoren entgegengesetzt ausgerichtet sind. Ein hoher Kosinusähnlichkeitswert (nahe 1) weist auf eine hohe semantische Korrespondenz hin (vgl. Jurafsky und Martin 2025, S. 110).

## 2.4 Information Retrieval

Information Retrieval (IR) beschäftigt sich mit der Suche nach Informationen in großen Sammlungen unstrukturierter Daten, typischerweise in Form von Textdokumenten. Ziel eines IR-Systems ist es, auf eine vom Nutzer formulierte Anfrage (Query) hin diejenigen Dokumente zu finden, die ein zugrunde liegendes Informationsbedürfnis bestmöglich erfüllen. Im Zentrum steht dabei nicht nur das reine Auffinden von Dokumenten, sondern auch deren Bewertung und Sortierung nach Relevanz (Ranking). Ein Dokument gilt als relevant, wenn es Informationen enthält, die der Nutzer in Bezug auf seine Anfrage als nützlich erachtet (vgl. Manning, Raghavan und Schütze 2012, S. 1-5).

Klassische IR-Verfahren wie das boolesche Retrieval oder statistische Ansätze wie TF-IDF (Term Frequency–Inverse Document Frequency) basieren auf lexikalischer Übereinstimmung zwischen Anfrage und Dokumenten. Diese Methoden sind effizient, stoßen jedoch an Grenzen, wenn etwa Synonyme oder kontextuelle Bedeutungen eine Rolle spielen; ein Phänomen, das als Vocabulary Mismatch bezeichnet wird. Moderne Systeme setzen daher zunehmend auf semantische Methoden, etwa durch Einbindung von Vektorraumdarstellungen und Sentence Transformern (vgl. Jurafsky und Martin 2025, S. 298).

### 2.4.1 Relevanz in IR-Systemen

Da eine Suchanfrage oft zahlreiche potenziell relevante Dokumente zurückliefert, ist es entscheidend, diese so zu ranken, dass die relevantesten Treffer an oberster Stelle erscheinen. Die meisten IR-Systeme berechnen hierzu einen Relevanzscore für jedes Dokument, auf dessen Basis eine geordnete Ergebnisliste erzeugt wird (vgl. Manning, Raghavan und Schütze 2012, S. 152ff.).

Die Bewertung der Effektivität von IR-Systemen erfolgt anhand verschiedener Metriken, die messen, wie gut relevante Dokumente identifiziert und korrekt eingeordnet werden. Die Definition von „Relevanz“ basiert dabei in der Regel auf annotierten Datensätzen oder Nutzerfeedback, was eine objektive Bewertung ermöglicht. Zu den wichtigsten Relevanzmetriken gehören:

- **Precision:** Misst den Anteil der abgerufenen Dokumente, die tatsächlich relevant sind. Eine hohe Präzision bedeutet, dass die Ergebnisse, die ein Nutzer sieht, größtenteils nützlich sind (vgl. Manning, Raghavan und Schütze 2012, S. 155).

$$\text{Precision} = \frac{|\text{relevant} \cap \text{retrieved}|}{|\text{retrieved}|} \quad (2.3)$$

wobei  $|\text{relevant} \cap \text{retrieved}|$  die Anzahl der Dokumente bezeichnet, die sowohl relevant als auch vom System abgerufen wurden, und  $|\text{retrieved}|$  die Gesamtzahl der abgerufenen Dokumente.

- **Recall:** Misst den Anteil der relevanten Dokumente im gesamten Korpus, die vom System abgerufen wurden. Eine hohe Recall-Rate bedeutet, dass das System die meisten der existierenden relevanten Informationen gefunden hat (vgl. Manning, Raghavan und Schütze 2012, S. 155).

$$\text{Recall} = \frac{|\text{relevant} \cap \text{retrieved}|}{|\text{retrieved}|} \quad (2.4)$$

wobei  $|\text{relevant}|$  die Gesamtanzahl der in der Datenbasis tatsächlich relevanten Dokumente ausdrückt.

- **Mean Reciprocal Rank (MRR):** Eine Metrik, die besonders bei Frage-Antwort-Systemen oder bei Anfragen mit nur einer korrekten Antwort nützlich ist. Sie bewertet die Platzierung der ersten relevanten Antwort in der Ergebnisliste; je näher diese an Position 1 liegt, desto höher der MRR-Wert (vgl. Mitra und Craswell 2018, S. 18).

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (2.5)$$

wobei  $|Q|$  die Gesamtzahl der Anfragen bezeichnet und  $\text{rank}_i$  die Rangposition der ersten relevanten Antwort für Anfrage  $i$  darstellt.

Diese Metriken ermöglichen eine systematische Evaluation verschiedener IR-Ansätze und dienen häufig als Grundlage für die Optimierung von Retrievalkomponenten innerhalb größerer Systeme, wie etwa in RAG-Anwendungen.

## 2.4.2 Dense Retrieval

Dense Retrieval überwindet die Einschränkungen traditioneller statistischer Verfahren, indem es neuronale Modelle nutzt, um Suchanfragen und Dokumente in dichte Vektorrepräsentationen im semantischen Raum zu überführen. Die Relevanz wird in der Regel über die Kosinus- oder Skalarproduktähnlichkeit zwischen den Embeddings gemessen, wodurch auch semantisch verwandte, aber lexikalisch unterschiedliche Dokumente identifiziert werden können (vgl. W. X. Zhao u. a. 2024, S. 3).

Ein prominentes Beispiel ist das Dense Passage Retrieval (DPR) von Karpukhin u. a. (2020), welches eine Dual-Encoder-Architektur verwendet. Dabei werden Fragen und Passagen jeweils durch separate Encoder (meist BERT-

Modelle) in Vektoren überführt, die so trainiert werden, dass relevante Paare eine hohe Ähnlichkeit aufweisen. DPR zeigte signifikante Leistungsverbesserungen gegenüber statistischen Ansätzen bei der Retrievalgenauigkeit in Open Domain Question Answering (ODQA) Aufgaben. Ein vollständiges IR-System besteht typischerweise aus mehreren Komponenten, die in einer Pipeline angeordnet sind. Die erste Stufe, das sogenannte First-stage Retrieval, reduziert den Suchraum, indem sie eine Menge relevanter Kandidatenpassagen abrufen. Nachgelagerte Komponenten übernehmen das Reranking dieser Kandidaten. Für komplexere Aufgaben wie Question Answering (QA) wird zusätzlich ein Reader-Modul eingesetzt, das auf Basis der abgerufenen Passagen eine konkrete Antwort extrahiert (vgl. W. X. Zhao u. a. 2024, S. 2–3).

## 2.5 Retrieval-Augmented Generation

Die Retrieval-Augmented Generation (RAG) stellt eine hybride Methode dar, die die generativen Fähigkeiten von LLMs mit der Fähigkeit zum gezielten Abrufen von Informationen aus externen Wissensquellen kombiniert. Das RAG-Konzept wurde ursprünglich von Lewis u. a. (2020) eingeführt. Ziel ist es, zentrale Limitationen reiner LLMs zu adressieren, insbesondere ihre begrenzte Fähigkeit, auf aktuelles oder domänenspezifisches Wissen zuzugreifen, sowie ihre Anfälligkeit für Halluzinationen. Durch die Integration eines Retrieval-Schritts können RAG-Systeme Antworten generieren, die nicht nur kohärent, sondern auch faktisch fundiert und überprüfbar sind (vgl. Gao u. a. 2023). Insbesondere im Kontext von Konversationssystemen wurde gezeigt, dass Retrieval-Augmentierung das Problem der Halluzination in modernen Frage-Antwort-Systemen reduzieren kann (vgl. Shuster u. a. 2021).

RAG verschmilzt synergetisch das intrinsische (im Modell vortrainierte) Wissen von LLMs mit großen, dynamischen Speichern externer Datenbanken. Dies führt zu einer Verbesserung der Genauigkeit und Glaubwürdigkeit der generierten Inhalte, insbesondere bei wissensintensiven Aufgaben, und ermöglicht kontinuierliche Wissensaktualisierungen sowie die Integration domänenspezifischer Informationen. Das grundlegende Framework von RAG basiert nach Gao u. a. (2023) auf drei Säulen:

- **Retrieval (Abruf)**: Hierbei werden relevante Informationen aus einer Wissensdatenbank (z.B. Dokumente, Texte) extrahiert, die zur Beantwortung einer bestimmten Anfrage benötigt werden. Dies geschieht typischerweise mithilfe eines neuronalen Retrievers, der semantisch ähnliche Passagen findet.
- **Augmentation (Erweiterung)**: Die abgerufenen Informationen werden

dem LLM als zusätzlicher Kontext zur Verfügung gestellt. Dies erweitert die Wissensbasis des LLM über sein internes, während des Trainings erworbenes Wissen hinaus.

- **Generation (Generierung):** Das LLM nutzt den ursprünglichen Input und den abgerufenen Kontext, um eine kohärente, faktengestützte und präzise Antwort zu formulieren.

Der nachfolgende Abschnitt erläutert die Architektur sowie die zentralen Komponenten einer RAG-Pipeline im Detail.

### 2.5.1 RAG-Pipeline

Die grundlegende Architektur eines RAG-Systems besteht aus mehreren ineinandergreifenden Komponenten, die in einer koordinierten Pipeline zusammenarbeiten. Diese Architektur ermöglicht es das parametrische Wissen von LLMs, welches während der Trainingsphase in den Modellgewichten gespeichert wurde, mit einem nicht-parametrischen Gedächtnis zu erweitern (vgl. Lewis u. a. 2020, S. 1). Der Prozess beginnt mit der Datenvorbereitung und -indexierung (siehe Abb. 2.3). Bevor das RAG-System Anfragen verarbeiten kann, muss die externe Wissensdatenbank aufbereitet werden. Dies beinhaltet das Zerlegen von umfangreichen Dokumenten, wie beispielsweise Artikeln, Büchern oder Webseiten, in kleinere, handhabbare Einheiten (Chunks oder Segmente), die typischerweise 100–300 Token umfassen, sodass sie vollständig innerhalb des Kontextfensters des LLM verarbeitet werden können. Jede dieser Einheiten wird anschließend mithilfe von spezialisierten Embedding-Modellen (siehe Abschnitt 2.3) in hochdimensionale Vektordarstellungen umgewandelt. Diese Embeddings werden wiederum effizient in einer Vektordatenbank indexiert, die für schnelle Ähnlichkeitssuchen ausgelegt ist. Dieser Indexierungsprozess schafft eine durchsuchbare Wissensbibliothek, die das semantische Matching von Benutzeranfragen ermöglicht und somit die Grundlage für effektives Retrieval legt (vgl. Gao u. a. 2023, S. 3f.).

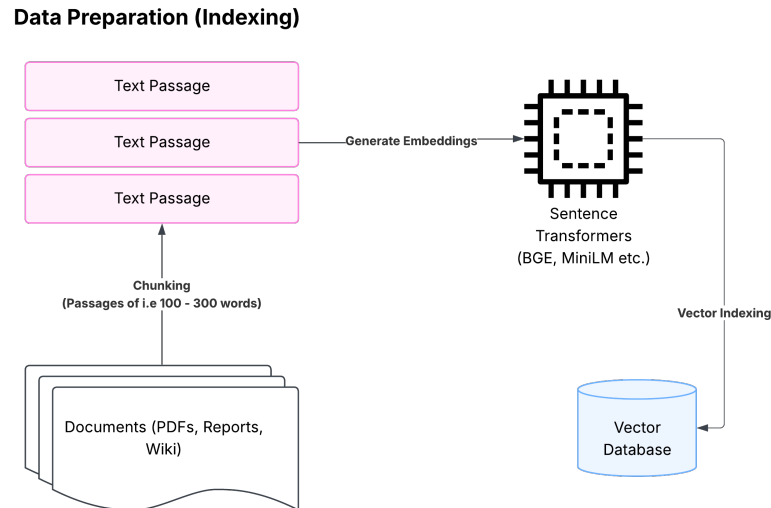


Abb. 2.3: Datenvorbereitung (Indexing) in RAG-Pipelines (eigene Darstellung)

Die Retrievalkomponente (siehe Abb. 2.4), auch als Retriever bezeichnet, ist für die Identifizierung relevanter Informationen zuständig. Wenn das System eine Benutzeranfrage (Query) erhält, wandelt der Retriever diese ebenfalls in ein Embedding um. Anschließend durchsucht er die vorab erstellte Vektordatenbank, um die Passagen zu finden, deren Embeddings die größte Ähnlichkeit zur Anfrage aufweisen. Das Ziel ist es, die *Top-k* der relevantesten Wissenspassagen zu extrahieren, die zur Beantwortung der Anfrage beitragen könnten. Der Einsatz von Dense Retrieval, wie in Abschnitt 2.4.2 beschrieben, ist hierbei von großem Vorteil, da es über lexikalische Übereinstimmungen hinausgeht und auch semantisch verwandte Inhalte finden kann (vgl. Gao u. a. 2023, S. 9).

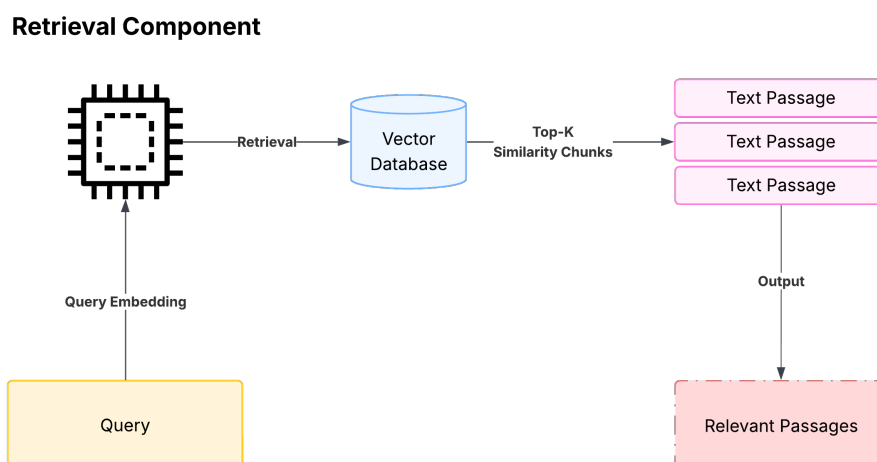


Abb. 2.4: Retriever in RAG-Pipelines (eigene Darstellung)

Die Generierungskomponente (siehe Abb. 2.5), auch als Generator bezeichnet, bildet den dritten Hauptteil des Systems und basiert typischerweise auf

einem vortrainierten Transformer-Modell (vgl. Lewis u. a. 2020, S. 2–3). Die vom Retriever ausgewählten relevanten Passagen werden zusammen mit der ursprünglichen Benutzeranfrage als erweiterter Kontext an den Generator übergeben. Der Generator nutzt diesen angereicherten Kontext, um eine kohärente, genaue und faktenbasierte Antwort zu generieren.

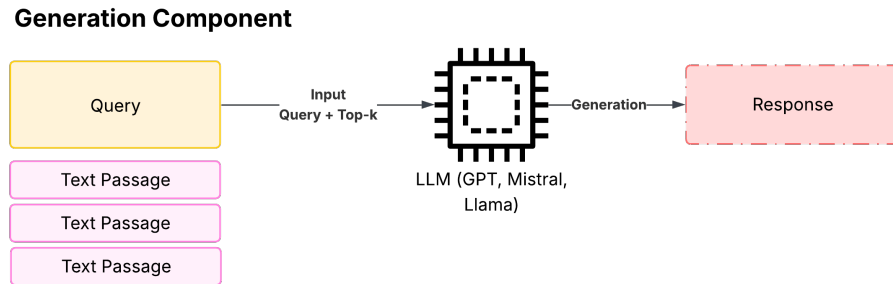


Abb. 2.5: Generator in RAG-Pipelines (eigene Darstellung)

Durch diese Augmentierung wird das LLM angeleitet, seine Antwort primär auf die bereitgestellten abgerufenen Informationen zu stützen. Dies reduziert signifikant die Wahrscheinlichkeit von Halluzinationen und stellt sicher, dass die generierte Ausgabe mit den Fakten aus der Wissensbasis übereinstimmt (vgl. Lewis u. a. 2020, S. 2). Im Gegensatz zu reinen generativen Modellen, die ihr Wissen ausschließlich aus den Trainingsdaten schöpfen, kann ein RAG-System durch den dynamischen Retrieval-Schritt bei Bedarf auf externe, aktuelle oder spezialisierte Informationen zugreifen und diese in Echtzeit für die Generierung nutzen.

### 2.5.2 RAG in Frage-Antwort-Systemen

Frage-Antwort-Systeme, auch als QA-System bezeichnet, sind ein zentraler und sich schnell entwickelnder Forschungsbereich im NLP. Hauptziel von QA-Systemen ist es, menschliche Interaktion mit Maschinen natürlicher zu gestalten, indem sie auf gestellte Fragen eine Antwort in natürlicher Sprache liefern, anstatt lediglich eine Liste von Dokumenten wie traditionelle Suchmaschinen (vgl. Farea u. a. 2022).

Formell kann ein QA-System als Vorhersagemodell  $M$  definiert werden, das eine Eingabe bestehend aus einer Frage  $q \in Q$  und einer Wissensquelle  $KS$  auf eine Antwort  $a \in A$  abbildet:

$$M : (Q; KS) \rightarrow A \quad \text{mit} \quad a_i = M(q_i, KS).$$

Die Wissensquelle  $KS$  kann dabei sehr unterschiedlich gestaltet sein. Sie kann

von strukturierten Wissensbasen bis hin zu unstrukturierten Dokumentensammlungen reichen. Diese Heterogenität macht QA zu einer besonders vielseitigen und komplexen Aufgabenstellung (vgl. Farea u. a. 2022, S. 12f.).

RAG stellt eine besonders effektive Anwendung für QA-Systeme dar. Während reine große Sprachmodelle vielfältige Fragen beantworten können, sind sie oft durch ihr statisches Wissen begrenzt. RAG-basierte Frage-Antwort-Systeme begegnen diesen Herausforderungen, indem sie einen Retriever integrieren, der relevante Passagen zur gestellten Frage liefert, welche anschließend von einem LLM genutzt werden (vgl. Shuster u. a. 2021).

Wie Shuster u. a. (2021) zeigen, führt die Einbindung externer Wissensquellen durch RAG zu mehreren entscheidenden Vorteilen im QA-Kontext:

- **Reduktion von Halluzinationen:** Durch die Verankerung der Generierung an extern abgerufene Fakten wird das Problem der Wissenshalluzination erheblich reduziert. Studien zeigen, dass RAG-Modelle die Anzahl halluzinierter Antworten um über 60% verringern können. Dieser Effekt ist besonders ausgeprägt bei Themen und Testdaten, die nicht in den ursprünglichen Trainingsdaten des Modells enthalten waren, da der Abruf hier intuitiv das im Modell fehlende Wissen ergänzt.
- **Verbesserung durch Retriever:** Die Einführung neuronaler Retriever ist ein wesentlicher Treiber für die Leistungssteigerung in QA-Aufgaben von RAG-Modellen. Es wird kontinuierlich an der Verbesserung dieser Retriever-Architekturen gearbeitet, beispielsweise durch Ansätze wie Poly-Encoder für eine feinere Kontext-Kandidaten-Bewertung von Dokumenten, oder iterative Retrieval-Verfahren, bei denen der Abruf durch Wiederholung verbessert wird. Es wird sogar erforscht, ob die internen Repräsentationen von vortrainierten Modellen direkt für den Abruf relevanter Kontexte genutzt werden können, um die Notwendigkeit eines separaten Retrievers zu eliminieren (*Retrieverless Retrieval*).
- **Generalisierung und Aktualität:** Die Fähigkeit, Wissen bei Bedarf abzurufen, ermöglicht es diesen Systemen, effektiv auf Szenarien zu generalisieren, die nicht in den Trainingsdaten enthalten waren. Dies ist entscheidend für offene Konversationsagenten, die in der Lage sein müssen, mit neuen oder ungesesehenen Themen umzugehen.
- **Anwendung in Dialogsystemen:** Obwohl RAG sich bereits in ODQA bewährt hat, ist seine Anwendung in wissensbasierten Dialogsystemen anspruchsvoller, da hier komplexe, mehrstufige Dialogkontexte für den Abruf und die Generierung kohärenter Antworten berücksichtigt werden müssen. Dennoch zeigen RAG-Modelle in diesen anspruchsvollen Aufga-

ben eine hohe Wissensbasis und Faktentreue, während sie gleichzeitig ihre konversationelle Fähigkeit bewahren.

Somit ermöglichen RAG-Systeme robustere und zuverlässigere QA-Anwendungen, die sich durch ihre Fähigkeit auszeichnen, Faktenfehler zu minimieren und sich dynamisch an neue Informationen anzupassen.

Die Effektivität solcher Systeme lässt sich evaluieren, indem die generierten Antworten mit sogenannten *Goldantworten* (auch als *Referenzen* oder *Ground Truth* bezeichnet) verglichen werden. Diese Referenzen sind dabei häufig innerhalb der verwendeten Wissensquelle enthalten, jedoch nicht direkt zugänglich. Das System muss diese Informationen aktiv extrahieren, um eine korrekte Antwort zu erzeugen (vgl. Farea u. a. 2022, S. 15).

Diese Evaluationsstrategie ermöglicht eine objektive Bewertung der Systemleistung und bildet die Grundlage für gängige Bewertungen. Um die Effektivität von RAG-Systemen gezielt zu analysieren, ist eine differenzierte Evaluierung der Einzelkomponenten und der generierten Ausgaben erforderlich. Der folgende Abschnitt widmet sich daher der gezielten Evaluierung von RAG und seinen Komponenten und stellt die wichtigsten Metriken im QA-Kontext vor.

### 2.5.3 Evaluierung von RAG-Systemen

Die Bewertung von RAG-Systemen stellt aufgrund ihrer hybriden Architektur, die sowohl Retrieval- als auch Generierungskomponenten kombiniert, sowie ihrer Abhängigkeit von dynamischen Wissensquellen einzigartige Herausforderungen dar. Die Gesamtleistung von RAG-Systemen hängt nicht nur von den einzelnen Komponenten ab, sondern auch von deren Interaktionen und ihrer integrierten Funktionalität. Angesichts der weitreichenden Anwendungsbereiche, der Heterogenität der internen Komponenten und der dynamischen Entwicklung in diesem Forschungsfeld ist die Etablierung eines einheitlichen und umfassenden Bewertungsrahmens eine kontinuierliche Forschungsaufgabe (vgl. Gan u. a. 2025).

Die umfassende Bewertung von RAG-Systemen kann in zwei unterschiedliche Bereiche unterteilt werden: die interne und die externe Evaluation:

- Die **interne Evaluation** befasst sich mit der Leistung auf Komponentenebene und methodenspezifischen Metriken innerhalb grundlegender RAG-Systeme, wobei der Fokus auf technischen Fortschritten liegt. Sie zerlegt die Bewertung eines gesamten RAG-Systems, um die Wechselwirkungen der internen Komponenten zu untersuchen (vgl. Gan u. a. 2025, S. 4ff.).

- Die **externe Evaluation** hingegen untersucht systemweite Faktoren wie Sicherheit und Effizienz, wobei der Schwerpunkt auf der praktischen Umsetzbarkeit und der Leistung in externen Aufgaben oder spezifischen Bewertungsumgebungen liegt (vgl. Gan u. a. 2025, S. 8f.).

Die folgenden Abschnitte widmen sich zunächst den zentralen Evaluierungsziele für Retrieval und Generierung in RAG-Systemen, bevor im Anschluss die gebräuchlichsten Metriken für beide Teilkomponenten vorgestellt werden. Für beide Bereiche haben sich spezifische Beurteilungsdimensionen etabliert (vgl. Gan u. a. 2025, S. 4f.).

#### Retrieval Beurteilungsdimensionen:

- **Relevanz:** Ob die abgerufenen Passagen tatsächlich relevante Informationen zur Benutzeranfrage enthalten.
- **Comprehensiveness (Vollständigkeit):** Ob die abgerufene Information alle benötigten Fakten oder Argumente zur Beantwortung der Frage abdeckt.

#### Generation Beurteilungsdimensionen:

- **Faithfulness:** Ob die generierte Antwort korrekt auf den abgerufenen Kontext Bezug nimmt, ohne Widersprüche oder erfundene Inhalte.
- **Correctness:** Ob die generierten Inhalte objektiv korrekt und mit bestehendem Weltwissen konsistent sind, auch unabhängig vom Input.

### Evaluationsmetriken in RAG-Systemen

Da RAG ein interdisziplinäres System ist, das auf traditionellen Forschungsfeldern wie IR und Natural Language Generation (NLG) aufbaut, werden eine Reihe von etablierten Metriken zur Bewertung aus diesen beiden Gebieten eingesetzt.

**Metriken für die Retrievalkomponente:** Die klassischen Metriken des IR (siehe Abschnitt 2.4.1) werden hier erweitert um rangbasierte Ansätze:

- **Recall@k:** Misst den Anteil der relevanten Dokumente, die unter den Top- $k$  abgerufenen Dokumenten enthalten sind.

$$\text{Recall@k} = \frac{|\text{RD} \cap \text{Top}_k^d|}{|\text{RD}|} \quad (2.6)$$

wobei  $|\text{RD}|$  die Gesamtmenge der relevanten Dokumente und  $\text{Top}_k^d$  die Menge der ersten  $k$  vom System zurückgegebenen Dokumente bezeichnet.

- **Precision@k**: Misst den Anteil der relevanten Instanzen innerhalb der Top- $k$  Ergebnisse.

$$\text{Precision@k} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.7)$$

wobei TP (True Positives) die Anzahl korrekt als relevant erkannter Dokumente und FP (False Positives) die Anzahl der fälschlicherweise als relevant eingestuften Dokumente bezeichnet.

- **F1-Score**: Bewertet das harmonische Mittel von Precision und Recall und beschreibt somit das Gleichgewicht zwischen beiden Metriken.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.8)$$

wobei Precision den Anteil der korrekt abgerufenen relevanten Dokumente bezeichnet und Recall den Anteil der insgesamt relevanten Dokumente, die vom System gefunden wurden. Der F1-Score nimmt hohe Werte an, wenn sowohl Precision als auch Recall hoch sind, und sinkt deutlich, sobald eine der beiden Größen gering ausfällt.

- **MRR** (Mean Reciprocal Rank): Bewertet die Rangposition des ersten relevanten Dokuments für eine Menge von Anfragen.

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (2.9)$$

wobei  $|Q|$  die Anzahl der Anfragen und  $\text{rank}_i$  die Rangposition des ersten relevanten Ergebnisses für Anfrage  $i$  bezeichnet.

(vgl. Gan u. a. 2025, S. 5)

- **Average Similarity**: Misst die durchschnittliche semantische Ähnlichkeit zwischen den abgerufenen Textpassagen  $c_i$  und dem Ground-Truth-Kontext  $G$ , basierend auf der Kosinusähnlichkeit der jeweiligen Embeddings.

$$\text{Average Similarity} = \frac{1}{n} \sum_{i=1}^n \text{sim}(c_i, G) \quad (2.10)$$

wobei  $n$  die Anzahl der betrachteten Textpassagen und  $\text{sim}(c_i, G)$  die mittels Kosinusähnlichkeit gemessene semantische Nähe zwischen der Passage  $c_i$  und dem Referenzkontext  $G$  bezeichnet. Dieses Verfahren orientiert sich konzeptionell an BERTSCORE (vgl. T. Zhang u. a. 2020).

**Metriken für die Generierungskomponente:** Zur Bewertung der generierten Antworten kommen Metriken aus der NLG zum Einsatz:

- **BLEU** (Bilingual Evaluation Understudy): Misst die lexikalische Über-

lappung zwischen der generierten Antwort und einem oder mehreren Referenztexten anhand von n-Gramm-Übereinstimmungen. Dabei wird ein gewichteter Durchschnitt über verschiedene n-Gramm-Längen gebildet und durch einen sogenannten Brevity Penalty bestraft, wenn die generierte Antwort deutlich kürzer als die Referenz ist.

$$\text{BLEU} = \text{BP} \times \exp \left( \sum_{n=1}^N w_n \log p_n \right) \quad (2.11)$$

wobei  $p_n$  die Präzision der n-Gramm-Übereinstimmungen bezeichnet,  $w_n$  die Gewichte der jeweiligen n-Gramm-Stufen darstellen und BP (Brevity Penalty) einen Strafterm für zu kurze generierte Sequenzen bildet. Der BLEU-Score liegt im Intervall  $[0, 1]$ , wobei ein höherer Wert auf eine größere lexikalische Übereinstimmung hinweist (vgl. Papineni u. a. 2002).

- **ROUGE-L** (Recall-Oriented Understudy for Gisting Evaluation): Misst die längste gemeinsame Teilsequenz (Longest Common Subsequence, LCS) zwischen generierter Antwort  $X$  und Referenzantwort  $Y$ .

$$R_{lcs} = \frac{\text{LCS}(X, Y)}{|Y|} \quad (2.12)$$

$$P_{lcs} = \frac{\text{LCS}(X, Y)}{|X|} \quad (2.13)$$

$$F_{lcs} = \frac{(1 + \beta^2) \cdot R_{lcs} \cdot P_{lcs}}{R_{lcs} + \beta^2 \cdot P_{lcs}} \quad (2.14)$$

wobei  $\text{LCS}(X, Y)$  die Länge der längsten gemeinsamen Teilsequenz der beiden Texte beschreibt,  $|X|$  und  $|Y|$  die jeweilige Textlänge von Generierung und Referenz darstellen und  $\beta$  ein Gewichtungssparameter ist, der festlegt, ob Recall oder Precision stärker gewichtet wird. Üblicherweise wird  $\beta = 1$  gesetzt, sodass:

$$F_{lcs} = \frac{2 \times R_{lcs} \times P_{lcs}}{R_{lcs} + P_{lcs}} \quad (2.15)$$

Das macht ROUGE-L besonders geeignet für QA- oder Zusammenfassungsaufgaben, da semantische Übereinstimmungen auch ohne exakte lexikalische Identität berücksichtigt werden (vgl. Lin 2004).

- **Semantic Similarity**: Zur Messung der semantischen Ähnlichkeit zwischen generierten Antworten und Referenztexten werden typischerweise Satz- oder Dokumenten-Embeddings verwendet, häufig erzeugt durch

Modelle wie Sentence-BERT (vgl. Reimers und Gurevych 2019).

$$\text{Cosine Similarity}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|} \quad (2.16)$$

wobei  $\vec{A}$  und  $\vec{B}$  die Embeddings der generierten Antwort bzw. der Referenz darstellen,  $\vec{A} \cdot \vec{B}$  das Skalarprodukt der Vektoren ist und  $\|\vec{A}\|$  sowie  $\|\vec{B}\|$  deren euklidische Normen bezeichnen. Diese Metrik bewertet somit nicht nur lexikalische, sondern auch semantische Ähnlichkeit und ist daher robust gegenüber Umformulierungen.

Zusätzlich etablieren sich neuere Ansätze wie Retrieval-Augmented Generation Assessment Scores (RAGAS), die LLMs als Evaluatoren (LLM-as-a-judge) einsetzen. Besonders im Fokus steht dabei das Kriterium der **Faithfulness**, also die inhaltliche Korrektheit der generierten Antwort im Verhältnis zu den tatsächlich abgerufenen Quellen. Die Bewertung erfolgt durch Zerlegung der Antwort in Einzelaussagen, die anschließend durch das LLM hinsichtlich ihrer Übereinstimmung mit dem bereitgestellten Kontext überprüft werden. Ein zentrales Ziel dieser Bewertung ist es, Halluzinationen zu erkennen und zu vermeiden. Ein Beispiel für eine mögliche Formalisierung des Faithfulness-Scores ist:

$$\text{Faithfulness} = \frac{|\mathcal{V}|}{|\mathcal{S}|} \quad (2.17)$$

Dabei bezeichnet  $\mathcal{S}$  die Gesamtmenge der in einer Antwort enthaltenen Aussagen und  $\mathcal{V} \subseteq \mathcal{S}$  die Teilmenge der Aussagen, die laut LLM durch die verwendeten Quellen gestützt werden. Der Wert liegt im Intervall  $[0, 1]$ , wobei 1 vollständige inhaltliche Korrektheit anzeigt (vgl. Es u. a. 2024).

## 3 Einsatz von LLMs und RAG im Finanzsektor

Der Einsatz generativer Künstlicher Intelligenz (GenAI) hat in den vergangenen Jahren branchenübergreifend stark zugenommen. Insbesondere LLMs ermöglichen eine qualitativ hochwertige Verarbeitung und Generierung natürlicher Sprache und eröffnen dadurch neue Anwendungsperspektiven. In datenintensiven und regulierten Branchen wie dem Finanzsektor ergeben sich daraus vielfältige Anwendungspotenziale, etwa in der automatisierten Kundenkommunikation, der Analyse von Finanzberichten, der Erkennung von Mustern in großen Datenmengen oder der Erstellung regulatorisch konformer Texte.

Gleichzeitig gehen mit dem Einsatz generativer Modelle auch zentrale Herausforderungen einher. Fehlende Faktentreue, Halluzinationen sowie eine begrenzte Nachvollziehbarkeit der Modellentscheidungen bergen erhebliche Risiken, insbesondere in sicherheitskritischen oder stark regulierten Kontexten. Vor diesem Hintergrund rückt die Kombination von LLMs mit RAG zunehmend in den Fokus. RAG ermöglicht durch die Anbindung an überprüfbare, externe Wissensquellen eine bessere Absicherung der generierten Inhalte und erlaubt die Nutzung domänenspezifischen Wissens, ohne dieses fest im Modell verankern zu müssen.

Dieses Kapitel beleuchtet den Einsatz von GenAI im Unternehmenskontext mit besonderem Fokus auf Anwendungen im Finanzsektor. Es zeigt zentrale Einsatzbereiche, Chancen und Risiken auf, diskutiert die Relevanz domänenspezifischer Modellanpassungen und geht auf regulatorische Rahmenbedingungen ein, die bei der Einführung solcher Systeme berücksichtigt werden müssen. Diese thematische Einbettung schafft die inhaltliche Grundlage für die weitere Auseinandersetzung mit GenAI im Finanzkontext.

### 3.1 GenAI in Unternehmen

Die Nutzung generativer GenAI in Unternehmen nimmt weiterhin deutlich zu. Dabei ist es wichtig, zwischen *künstlicher Intelligenz (KI)* im Allgemeinen und *generativer KI (GenAI)* zu unterscheiden: Während KI als Oberbegriff auch nicht-generative Verfahren wie Entscheidungsbäume, Clustering oder klassische maschinelle Lernverfahren umfasst, bezieht sich GenAI speziell auf Modelle, die eigenständig Inhalte wie Text, Code, Bilder oder Audio erzeugen können.

Laut einer aktuellen Studie von McKinsey & Company verwenden 78 % der befragten Organisationen weltweit KI in mindestens einer Unternehmensfunktion, welches ein signifikanter Anstieg gegenüber 55 % im Vorjahr ist (Stand Juli 2024). Dabei hat sich die Nutzung von GenAI zwischen 2023 und 2024 mehr als verdoppelt, von 33 % auf 71 % (siehe Abbildung 3.1).

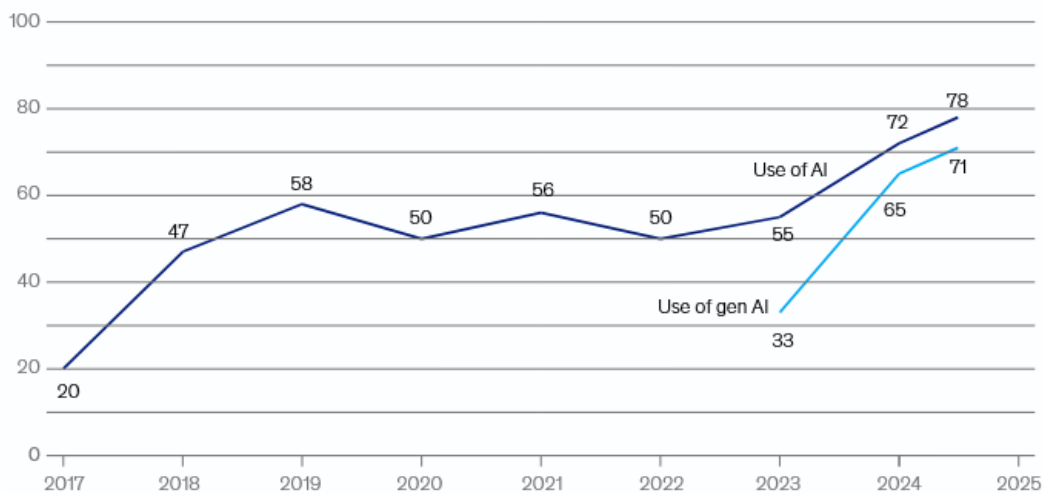


Abb. 3.1: Organisationen, die GenAI in mindestens einer Funktion einsetzen.  
Quelle: Singla u. a. (2025, S. 15)

Die Studie basiert auf einer Online-Befragung von 1.491 Teilnehmenden aus 101 Ländern, die im Juli 2024 durchgeführt wurde. Dabei wurde eine breite Palette von Branchen, Unternehmensgrößen und Regionen abgedeckt. 42 % der Befragten gaben an, für Unternehmen mit einem Jahresumsatz von über 500 Millionen US-Dollar tätig zu sein. Um eine globale Repräsentativität sicherzustellen, wurde die Gewichtung der Antworten entsprechend dem BIP-Anteil der jeweiligen Länder vorgenommen (vgl. Singla u. a. 2025).

Mit dem zunehmenden Einsatz von GenAI in Unternehmen wächst auch das Bewusstsein für damit verbundene Risiken. Laut der McKinsey & Company Studie setzen immer mehr Organisationen gezielte Maßnahmen zur Risikominimierung um. Besonders im Bezug auf Ungenauigkeit, Cybersecurity und Verletzungen geistigen Eigentums (siehe Abbildung 3.2). Diese drei Risiken gehören zugleich zu den häufigsten Ursachen negativer Auswirkungen durch GenAI im Unternehmenskontext (vgl. Singla u. a. 2025, S. 6f.).

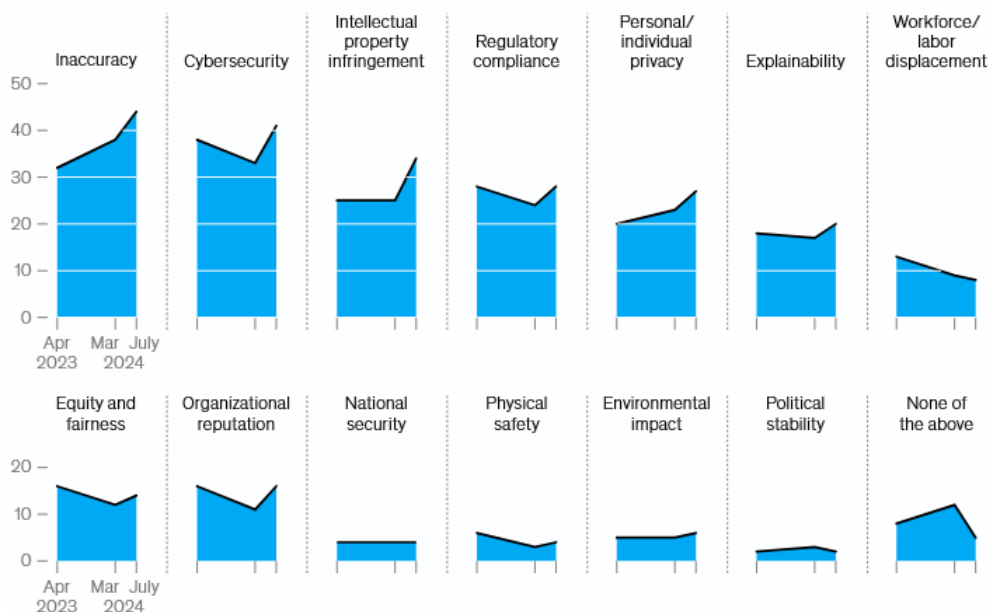


Abb. 3.2: Von Unternehmen adressierte GenAI-bezogene Risiken. Quelle: Singla u. a. (2025, S. 6)

Diese Zahlen verdeutlichen, dass generative KI-Modelle keine isolierte technologische Neuerung mehr darstellen, sondern zunehmend ein integraler Bestandteil moderner Geschäftsprozesse sind. Die Entwicklung verdeutlicht auch die zunehmende Relevanz eines strukturierten GenAI-Risikomanagements. Gerade in sensiblen Sektoren wie dem Finanzwesen sind technische Schutzmaßnahmen sowie die Einführung organisatorischer und rechtlicher Rahmenbedingungen von großer Bedeutung.

### 3.1.1 Anwendungen von LLMs im Finanzbereich

LLMs, die auf umfangreichen Finanzdatensätzen trainiert werden, eröffnen vielfältige Anwendungsmöglichkeiten im Finanzsektor. Ein gemeinsamer Bericht der European Securities and Markets Authority (ESMA), des Institut Louis Bachelier und des Alan Turing Institute identifiziert zentrale Anwendungsfelder, in denen LLMs signifikante Effizienz- und Produktivitätsgewinne erwarten lassen (vgl. Bagattini u. a. 2024).

Zum einen können LLMs die öffentliche Kommunikation und die Interaktion mit Kunden vereinfachen, indem sie komplexe Finanzinformationen in verständlicher Sprache aufbereiten oder in Form von Chatbots für eine personalisierte Kundenbetreuung eingesetzt werden. Ein weiterer zentraler Bereich betrifft die Sicherheit in der Finanzdienstleistung: LLMs unterstützen bei der Erkennung von Betrug, der Markt- und Handelsüberwachung sowie der Risikoanalyse komplexer Finanzprodukte, insbesondere durch die Analyse großer Mengen

an Transaktionsdaten. Darüber hinaus ermöglichen LLMs die Generierung finanzieller Einblicke, etwa zur Marktüberwachung, zur Analyse von Unternehmenskennzahlen, zur Bewertung persönlicher Investmentportfolios oder im Bereich der Environmental-, Social- und Governance (ESG) Bewertung. Nicht zuletzt finden LLMs Anwendung in Finanzierung und Investmentaktivitäten, wie etwa in der Unterstützung von Asset Management und Investment Banking, der Optimierung von Treasury-Prozessen sowie der Entwicklung von Strategien im Bereich Private Equity und Venture Capital.

Ergänzend dazu beleuchten aktuelle Studien von H. Zhao u. a. (2024) und Nie u. a. (2024) weitere zentrale Aspekte und Anwendungsbereiche von LLMs im Finanzsektor, insbesondere im Hinblick auf quantitative Handelsstrategien, Finanzprognosen und risikobezogene Anwendungen.

So werden unter anderem folgende Schwerpunkte identifiziert: Zunächst zeigen sich LLMs als äußerst leistungsfähig bei der Bearbeitung linguistischer Aufgaben. Sie ermöglichen etwa die prägnante Zusammenfassung und Extraktion relevanter Informationen aus komplexen Finanzdokumenten, was insbesondere bei der Analyse umfangreicher Unternehmensberichte oder regulatorischer Texte von Vorteil ist. Darüber hinaus erleichtern sie das Management unterschiedlicher Dokumentstrukturen sowie die Erkennung benannter Entitäten (Named Entity Recognition, NER). Ein besonders prominentes Einsatzfeld stellt die Sentimentanalyse dar. LLMs werden zur Ableitung der Marktstimmung aus Finanznachrichten, Social-Media-Beiträgen oder Unternehmensveröffentlichungen eingesetzt. Dabei erweisen sie sich als hilfreich bei der Interpretation informeller Ausdrucksweisen, Emojis, Memes sowie fachspezifischer Terminologie. Auch in der Analyse finanzieller Zeitreihen zeigen LLMs vielversprechende Ansätze. Sie unterstützen bei der Prognose von Markttrends, der Identifikation von Anomalien und der Klassifikation von Finanzdaten. Ergänzend lassen sie sich für die Datenaugmentation, beispielsweise zur Generierung synthetischer Daten oder zur Imputation fehlender Werte, einsetzen. Ein weiterer Bereich betrifft die finanzielle Argumentation. Hier tragen LLMs dazu bei, große Mengen heterogener Finanzinformationen zu strukturieren, Investitionsempfehlungen zu generieren und Entscheidungsprozesse in Echtzeit zu unterstützen, beispielsweise in der Finanzplanung, der Portfoliooptimierung oder beim Risikomanagement. Zunehmend kommen LLMs auch in der agentenbasierten Modellierung zum Einsatz. Dabei werden wirtschaftliche Entscheidungsprozesse simuliert, um Interaktionen zwischen Agenten, Märkten und Umwelteinflüssen besser zu verstehen. Dies eröffnet Potenziale für den Einsatz in Multi-Agenten-Systemen, der Handelsautomatisierung oder der Simulation makroökonomischer Szenarien (vgl. Nie u. a. 2024, S. 6ff.).

Die Attraktivität dieser Technologien für den Finanzsektor ergibt sich vor allem aus mehreren Schlüsseigenschaften: Dazu zählen das tiefgehende Verständnis finanzspezifischer Sprache, die Fähigkeit zur effizienten Anpassung an neue Aufgaben mittels Transfer Learning, die Skalierbarkeit für Echtzeitanalysen sowie die Möglichkeit zur Integration multimodaler Daten. Hinzu kommen die verbesserte Interpretierbarkeit der Modelloutputs und eine hohe Anpassungsfähigkeit an verschiedene Marktsegmente und Finanzinstrumente (vgl. Nie u. a. 2024, S. 6).

### 3.1.2 Finanzbasierte LLMs

Neben allgemeinen LLMs wie GPT, Mistral oder LLaMA haben sich zunehmend auch domänenspezifische Modelle etabliert, die gezielt auf den Einsatz im Finanzbereich ausgerichtet sind. Diese spezialisierten Modelle sollen finanzbezogene Aufgaben zuverlässiger, präziser und datensensibler erfüllen. Grundsätzlich lassen sich drei Entwicklungsansätze unterscheiden:

- **Feinabgestimmte Modelle (Fine-Tuning):** Allgemeine LLMs werden nachträglich auf finanzspezifischen Datensätzen weitertrainiert. Modelle wie *FinGPT* (vgl. H. Yang, Liu und Wang 2023) erzielen damit oft bessere Ergebnisse bei Klassifikationsaufgaben als nicht-spezialisierte Modelle. Einschränkungen zeigen sich jedoch bei generativen Aufgaben, insbesondere wenn das Fine-Tuning auf begrenzten oder einseitigen Daten beruht.
- **Neu trainierte Modelle:** Einige Modelle wurden vollständig neu entwickelt und ausschließlich für den Finanzbereich trainiert. *BloombergGPT* (vgl. Wu u. a. 2023) oder *Xuan Yuan 2.0* (vgl. X. Zhang, Q. Yang und Xu 2023) kombinieren öffentlich verfügbare mit proprietären Finanzdaten im Pretraining. Trotz begrenztem Anteil exklusiver Daten zeigen sie in Benchmarks eine deutlich erhöhte Leistung bei Finanzanwendungen.
- **Allgemeine Modelle mit Zero- oder Few-Shot-Ansätzen:** Leistungsstarke Basismodelle können auch ohne spezifisches Training bereits viele finanzrelevante Aufgaben bewältigen. Besonders in datenarmen Szenarien, in denen domänenspezifisches Training aus Datenschutz- oder Ressourcengründen nicht möglich ist, stellen sie eine praktikable Alternative dar (siehe Abschnitt 2.2.3).

Je nach Datenschutzerfordernis bietet sich entweder der Einsatz selbstgehosteter Open-Source-Modelle oder externer API-basierter LLMs an. Während letztere eine einfache Integration und hohe Leistung bieten, ermöglichen selbst-

betriebene Lösungen die vollständige Kontrolle über sensible Finanzdaten. Ergänzend können LLMs durch externe Werkzeuge wie Python-Bibliotheken zur Portfoliooptimierung oder Schnittstellen zu Finanzdatenanbietern erweitert werden (vgl. Li u. a. 2023).

### 3.1.3 Herausforderungen beim Einsatz von LLMs im Finanzbereich

Die Studie von Maple und Sabuncuoglu (2024), die im Rahmen des FAIR-Programms (Framework for Responsible Adoption of AI in Financial Services) veröffentlicht wurde, identifiziert eine Vielzahl an Herausforderungen, die mit dem Einsatz von LLMs im Finanzsektor einhergehen. Diese betreffen sowohl technische als auch ethische, regulatorische und organisationale Aspekte und stehen in direktem Zusammenhang mit den Themen, Herausforderungen und Zielen der vorliegenden Arbeit.

Ein zentrales Problem ist die potenzielle Erzeugung von falschen oder irreführenden Inhalten durch LLMs, sogenannten Halluzinationen (siehe Abschnitt 2.2.4). Diese stellen insbesondere in sicherheitskritischen Anwendungsfeldern wie dem Kundenservice oder der Entscheidungsunterstützung im Finanzbereich ein erhebliches Risiko dar. Halluzinationen gefährden nicht nur die Qualität der bereitgestellten Informationen, sondern können auch das Vertrauen der Nutzer in GenAI-Systeme untergraben. Hinzu kommen Risiken im Bereich der Datenintegrität und -genauigkeit, die sich direkt auf die Qualität der generierten Antworten auswirken. Die Verwendung unsauberer, veralteter oder unvollständiger Datenquellen kann dazu führen, dass LLMs unpräzise oder kontextlich unangemessene Ausgaben erzeugen. Dies betrifft insbesondere domänenspezifische Aufgaben, bei denen Fachwissen und aktuelles Kontextverständnis entscheidend sind. Datenschutz und Sicherheit bilden weitere zentrale Herausforderungen. Die Verarbeitung sensibler Finanzdaten durch LLMs erfordert strikte Schutzmechanismen, um die Privatsphäre von Kunden und Mitarbeitenden zu wahren. Sicherheitslücken wie Prompt Injections oder Datenvergiftungen können nicht nur die Systemintegrität kompromittieren, sondern auch zu unbeabsichtigten Datenlecks führen. Auch die inhärente Komplexität von LLMs erschwert ihre Nachvollziehbarkeit und Erklärbarkeit. Dies ist besonders relevant für regulierte Branchen wie den Finanzsektor, in denen Transparenz und Verantwortlichkeit zentrale Anforderungen darstellen. Fehlende Interpretierbarkeit kann zudem die Fehlerdiagnose erschweren, insbesondere wenn LLMs in dynamischen oder stark kontextabhängigen Szenarien eingesetzt werden. Nicht zuletzt zeigt sich, dass LLMs Schwierigkeiten bei der semantischen Einordnung

von Ereignissen und bei der Bewertung grundlegender Phrasen haben, was sich negativ auf die kontextuelle Relevanz ihrer Antworten auswirken kann. Auch der eingeschränkte Zugang zu Echtzeitdaten limitiert ihre Eignung für Aufgaben, die aktuelle Informationen voraussetzen.

Diese Herausforderungen machen deutlich, dass der Einsatz von LLMs im Finanzbereich einer sorgfältigen Bewertung und gezielter Steuerung bedarf. Die Kombination mit RAG verspricht hier ein mögliches Lösungskonzept, indem externe, geprüfte Wissensquellen zur Unterstützung der Antwortgenerierung genutzt werden.

### 3.2 Regulatorische Rahmenbedingungen: EU AI Act

Mit der am 1. August 2024 in Kraft getretenen Verordnung (EU) 2024/1689 des Europäischen Parlaments und des Rates zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz (im Folgenden: EU AI Act)<sup>1</sup> liegt erstmals ein einheitlicher regulatorischer Rahmen für den Einsatz von KI in der Europäischen Union vor. Ziel des EU AI Act ist es, Innovationen zu ermöglichen und gleichzeitig Risiken für Grundrechte, Sicherheit und demokratische Prozesse zu minimieren. Der EU AI Act verfolgt einen risikobasierten Ansatz und klassifiziert hierfür KI-Systeme nach ihrem Risikopotenzial. Er unterscheidet zwischen verbotenen, hochriskanten, begrenzt riskanten und minimal riskanten KI-Systemen. Damit fallen Anwendungen im Finanzbereich, wie etwa Systeme zur Kreditwürdigkeitsprüfung, Risikobewertung oder automatisierten Entscheidungsunterstützung, unter Umständen in die Kategorie der Hochrisiko-KI-Systeme gemäß Art. 6 Abs. 2 lit. a EU AI Act i. V. m. III Nr. 5. Für diese gelten umfangreiche regulatorische Anforderungen, darunter die Einrichtung eines Risikomanagementsystems, Maßnahmen zur Sicherstellung der Datenqualität, technische Dokumentationspflichten sowie Anforderungen an menschliche Kontrolle und Transparenz (Art. 9–15 EU AI Act). Ein zentrales Element des EU AI Act sind KI-Modelle mit allgemeinem Verwendungszweck, sogenannte GPAI-Modelle (aus dem eng. von „general purpose AI model“), also LLMs mit breiter Einsatzfähigkeit wie GPT, LLaMA oder Mistral. Besonders leistungsfähige GPAI-Modelle, die ein sogenanntes systemisches Risiko darstellen (z. B. Modelle mit Trainingsaufwand über  $10^{25}$  FLOPs), unterliegen verschärften Anforderungen, etwa zur Risikoanalyse, Sicherheitsüberprüfung (Adversarial

---

<sup>1</sup>Verordnung (EU) 2024/1689 des Europäischen Parlaments und des Rates vom 13. Juni 2024 zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz und zur Änderung der Verordnungen (EG) Nr. 300/2008, (EU) Nr. 167/2013, (EU) Nr. 168/2013, (EU) 2018/858, (EU) 2018/1139 und (EU) 2019/2144 sowie der Richtlinien 2014/90/EU, (EU) 2016/797 und (EU) 2020/1828 (Verordnung über künstliche Intelligenz)

Testing) und Dokumentation (Art. 50–55 EU AI Act).

Für den Einsatz von LLMs und RAG-Systemen im Finanzbereich bedeutet dies, dass insbesondere domänenspezifische Modelle einer rechtlichen und technischen Prüfung unterzogen werden müssen, sobald sie in regulierte Anwendungsfelder integriert werden oder mit sensiblen, personenbezogenen Daten trainiert bzw. betrieben werden. Der EU AI Act formuliert hierfür Anforderungen an Risikoanalyse, Transparenz und Dokumentation, die durch RAG-Architekturen unterstützt werden können, unter anderem durch die Trennung von Modell und Wissensquelle zur besseren Nachvollziehbarkeit. Ergänzt durch domänenspezifisches Fine-Tuning und eine klar definierte Zweckbestimmung lassen sich so die regulatorisch geforderte Modelltransparenz stärken und potenzielle Risiken frühzeitig adressieren.

### 3.3 RAG als Lösungsansatz im Finanzkontext

Vor dem Hintergrund der zuvor dargestellten Herausforderungen gewinnt die Methode der RAG zunehmend an Bedeutung. RAG erweitert klassische Sprachmodelle, indem externe, domänenspezifische Informationsquellen in den Generierungsprozess einbezogen werden. Dadurch lassen sich zentrale Schwächen von LLMs, wie etwa Halluzinationen, mangelnde Kontexttreue oder begrenzte Aktualität, gezielt adressieren.

Eine aktuelle Studie von Iaroshev u. a. (2024) untersucht den Einsatz von RAG-Systemen im Kontext der Finanzberichterstattung. Ziel war es, die Frage-Antwort-Fähigkeiten von LLMs im Hinblick auf halbjährliche und vierteljährliche Bankberichte zu verbessern, um Privatanlegern fundierte Entscheidungsgrundlagen zu bieten. Im Rahmen eines Design-Science-Ansatzes wurden verschiedene RAG-Konfigurationen evaluiert, wobei insbesondere die Dimensionen Kontextrelevanz, Antworttreue und Antwortrelevanz im Mittelpunkt standen. Die Ergebnisse bestätigen das Potenzial von RAG zur Qualitätssteigerung generativer Systeme. Insbesondere Kombinationen aus leistungsfähigen LLMs und hochwertigen Retrievalkomponenten führten zu konsistenten, nachvollziehbaren und präzisen Antworten. Dabei wurden qualitative Fragestellungen systematisch besser beantwortet als quantitative, was die Stärke von RAG im Umgang mit beschreibenden, textbasierten Informationen unterstreicht. Zugleich zeigt die Studie, dass die Struktur und Kohärenz der zugrunde liegenden Finanzdokumente einen entscheidenden Einfluss auf die Gesamtleistung haben.

Ergänzend dazu stellt Kim u. a. (2025) die bislang wenig beachtete Wechselwirkung zwischen Retrieval- und Generierungskomponente in den Mittelpunkt. Die Autoren betonen, dass die Antwortqualität nicht nur vom Sprachmodell

selbst, sondern maßgeblich von der Qualität des abgerufenen Kontexts abhängt. Besonders bei komplexen und dichten Texten, wie sie im Finanzbereich vorliegen, ist eine präzise Abstimmung beider Komponenten essenziell, um sachlich korrekte und relevante Antworten zu generieren.

Beide Studien verdeutlichen die Relevanz des gewählten Untersuchungsgegenstands und liefern eine konzeptionelle wie empirische Grundlage für die hier vorliegende Arbeit. Gleichzeitig unterscheidet sich der Ansatz der vorliegenden Arbeit in zentralen Punkten: Im Fokus steht eine komponentenbasierte Evaluation, bei der sowohl das Retrieval als auch die Generierung getrennt und systematisch mithilfe etablierter Metriken aus dem IR und NLG bewertet werden. Darüber hinaus wird mit RAGAS ein neuartiger Bewertungsrahmen integriert, der sogenannte Faithfulness-Scores zur Einschätzung der inhaltlichen Korrektheit nutzt. Diese Arbeit versteht sich damit nicht als bloße Wiederholung bestehender Studien, sondern als eigenständiger Beitrag zur vertieften Analyse von RAG-Systemen im Finanzkontext und fokussiert dabei besonders die Reduktion von Halluzinationen, die Verbesserung semantischer Genauigkeit und Kontexttreue sowie die modularisierte Evaluation der einzelnen RAG-Komponenten.

### 3.4 Domänenspezifisches RAG und Finanzdatennutzung

Neben der konzeptionellen Bedeutung von RAG im Finanzkontext wird zunehmend deutlich, dass die Effektivität solcher Systeme maßgeblich von der domänenspezifischen Aufbereitung und Repräsentation der zugrunde liegenden Finanzdaten abhängt.

Ein zentrales Problem bei der Verarbeitung von Finanzdokumenten liegt in der Fragmentierung semantischer Einheiten durch klassische Chunking Methoden. Yepes u. a. (2024) zeigen, dass eine naive Segmentierung auf Basis von Tokenlängen oder Standardgrenzen zu einer signifikanten Verschlechterung der Antwortqualität führen kann. Im Gegensatz dazu schlagen sie ein domänenspezifisches, regelbasiertes Chunking Verfahren vor, das sich an der semantischen Struktur auf Grundlage der von der US-amerikanischen Börsenaufsichtsbehörde SEC standardisierten 10K-Berichten (vgl. SEC o. J.) orientiert. Dazu zählen etwa Abschnittsmarker wie „Item 1A: Risk Factors“, Inhaltsverzeichnisse und typografische Muster. Die Evaluation belegt, dass durch diese Strukturierung sowohl die Abrufqualität als auch die Kohärenz der generierten Antworten messbar verbessert werden.

Ein weiterer zentraler Aspekt für die Effektivität von RAG-Systemen im Finanzkontext betrifft die eingesetzten Embedding-Modelle. Tang und Y. Yang

(2025a) zeigen in einer umfassenden empirischen Untersuchung, dass moderne, auf allgemeine Sprachverarbeitung trainierte Embedding-Modelle signifikante Leistungsabfälle aufweisen, wenn sie auf finanzspezifische Aufgaben angewendet werden. Zur Analyse wurde mit FinMTEB (vgl. Tang und Y. Yang 2025b) ein spezialisiertes Benchmark-Framework entwickelt, der Aufgaben wie Textklassifikation, Clustering und Reranking ausschließlich auf Finanzdaten abbildet. Die Ergebnisse zeigen, dass selbst leistungsfähige Modelle wie E5 oder OpenAI-Embeddings Schwierigkeiten haben, domänenspezifische semantische Strukturen angemessen zu erfassen. Diese Beobachtung bleibt auch bestehen, wenn Unterschiede in der Datenkomplexität kontrolliert werden. Die Studie unterstreicht damit die Notwendigkeit, domänenspezifische Embedding-Modelle einzusetzen und ihre Leistung mit passenden Benchmarks wie FinMTEB zu evaluieren, vor allem für retrieval-basierte Anwendungen im Finanzbereich.

Die genannten Arbeiten verdeutlichen, dass eine effektive RAG-Pipeline im Finanzbereich nicht nur ein leistungsfähiges Sprachmodell, sondern auch eine präzise Steuerung der Retrievalkomponente erfordert. Dabei ist ein systematisches Vorgehen, angefangen bei der Datenvorverarbeitung bis hin zur Auswahl der Embedding-Modelle von großer Bedeutung.

## 4 Methodik und Implementierung

Dieses Kapitel beschreibt die methodische Vorgehensweise und die technische Umsetzung der in dieser Arbeit entwickelten RAG-Lösung im Finanzkontext. Während Kapitel 2 die theoretischen Grundlagen zu LLMs, Herausforderungen wie Halluzinationen sowie RAG-Systeme und Evaluierungsansätze eingeführt hat, beleuchtete Kapitel 3 die spezifischen Anforderungen und Anwendungsfelder im Finanzbereich, insbesondere im Umgang mit domänenspezifischen Daten.

Kapitel 4 stellt nun die praktische Implementierung in den Mittelpunkt. Nach einer Übersicht über die verwendeten Datenquellen und deren Aufbereitung wird die Umsetzung der zentralen Systemkomponenten detailliert erläutert. Dazu zählen die Datenvorverarbeitung mittels Segmentierung und Indexierung, das Retrieval-System bestehend aus einem Embedding-Modell und einer Vektordatenbank, die Modellbereitstellung und lokale Inferenzumgebung mit Ollama und Mistral-7B sowie der Aufbau der RAG-Architektur.

### 4.1 Datenaufbereitung und QA-Datensatz

Für die Domäne der Finanzdaten bilden die offiziellen SEC 10-K Berichte (vgl. SEC o. J.) eine zentrale Wissensquelle. Diese jährlich von börsennotierten US-Unternehmen eingereichten, ausführlichen Geschäftsberichte enthalten strukturierte Informationen zu Geschäftsmodellen, Risiken, Finanzen und weiteren relevanten Themen und stellen somit eine hochwertige und verlässliche Grundlage für das RAG-System dar.

Zur automatisierten Erfassung der aktuellsten Berichte wurde eine eigene Datenpipeline mit einer Anwendungsprogrammierschnittstelle (API) in Python implementiert. Dabei werden gezielt die neuesten Berichte für eine ausgewählte Gruppe von Unternehmen anhand ihrer Börsensymbole (z.B. AAPL, TSLA, MSFT) abgerufen und für jeden Ticker das aktuellste verfügbare 10-K Dokument in Textform heruntergeladen. Die Originalberichte weisen umfangreiche, zum Teil verschachtelte Dokumentenstrukturen auf, die vor der Weiterverarbeitung bereinigt werden müssen. Für die Extraktion und Bereinigung des reinen Textinhalts wird in Python zunächst ein Parser verwendet um alle HTML- und XML-Tags zu entfernen, sodass ein sauberer, fließender Text für die weitere

Verarbeitung entsteht.

Um die Texte für das Retrieval effizient nutzbar zu machen, werden sie in semantisch konsistente Abschnitte unterteilt. Dieser Vorgang wird als Chunking oder Segmentierung bezeichnet. Hierfür kommt die Textsplitter Komponente der Open Source Bibliothek LangChain<sup>1</sup> zum Einsatz, die durch rekursive Zerlegung die Texte in handhabbare Fragmente unterteilt. Im Rahmen der Implementierung wurden die Parameter auf eine Chunkgröße von etwa 1000 Zeichen mit einem Überlapp von 200 Zeichen eingestellt, um den Zusammenhang zwischen benachbarten Textabschnitten zu erhalten und Informationsverluste an den Schnittstellen zu minimieren. Die einzelnen Segmente werden anschließend in einem geordneten Verzeichnis pro Ticker abgelegt, um eine einfache und performante Indexierung und Suche im weiteren Verlauf zu ermöglichen.

Dieser automatisierte Vorverarbeitungsschritt schafft die Voraussetzung für eine präzise und effiziente Suche in großen, heterogenen Finanztexten und bildet die Basis für den Retriever im RAG-System.

Für die Evaluation des Systems wurde ein bereits bestehender Frage-Antwort-Datensatz mit Ground Truth Antworten und Kontextinformationen über die Plattform HuggingFace geladen<sup>2</sup>. Der Datenbestand entsprang ursprünglich ebenfalls 10-K Berichten, allerdings mit einem älteren Stand (Jahr 2023). Der Datensatz enthält Finanzfragen sowie die dazugehörigen Antworten und Kontextdaten sowie Tickersymbole zu den jeweiligen Unternehmen. Er bildet die Grundlage für die Generierung von Eingabefragen sowie die spätere Bewertung der Antwortqualität. Insgesamt umfasst der Datensatz rund 7,000 Frage-Antwort-Paare mit zugehörigen Ground Truth Kontexten und stellt somit eine geeignete Basis für das Testen von Retrieval- und Antwortgenerierungskomponenten im Finanzkontext dar.

## 4.2 Retriever Implementierung

Das Retrievalsystem stellt eine zentrale Komponente der entwickelten RAG-Lösung dar. Es besteht aus zwei wesentlichen Bausteinen: einem Embedding-Modell, das Textabschnitte in semantische Vektorrepräsentationen überführt, sowie einer Vektordatenbank, die diese Vektoren speichert und durch Indexierung eine effiziente Ähnlichkeitssuchen ermöglicht. Im Folgenden werden diese Komponenten getrennt beschrieben, bevor die Integration und Nutzung im Gesamtsystem erläutert wird.

---

<sup>1</sup><https://github.com/langchain-ai/langchain>, Zugriff am 19.09.2025

<sup>2</sup><https://huggingface.co/datasets/virattt/financial-qa-10K>, Zugriff am 19.09.2025

### 4.2.1 Vektordatenbank: FAISS

Für die Umsetzung des Retrievalsystems wurde ein lokaler Vektorindex mit der Facebook AI Similarity Search (FAISS)<sup>3</sup> realisiert. Vektordatenbanken wie FAISS sind essenziell für RAG-Systeme, da sie große Mengen an Embeddings indexieren und effiziente Ähnlichkeitssuchen ermöglichen. Sie bilden die Grundlage für semantische Suchverfahren, wie sie in modernen IR-Systemen eingesetzt werden (vgl. Le Ma u. a. 2023).

### 4.2.2 Embedding-Modell: BAAI General Embedding

Zur Vektorisierung der Textsegmente kam ein Modell aus der BAAI General Embedding (BGE) Reihe zum Einsatz. Obwohl BGE ursprünglich für den chinesischen Sprachraum entwickelt wurde, lässt es sich durch Feinabstimmung oder auch direkt in anderen Sprachen und Domänen verwenden. Aufgrund seiner breiten Integration in Frameworks wie LangChain und HuggingFace sowie seiner großen Verbreitung eignet es sich gut für den praktischen Einsatz in RAG-Anwendungen (vgl. Xiao u. a. 2024).

Für die vorliegende Implementierung wurde die englischsprachige Variante BAAI/bge-small-en-v1.5<sup>4</sup> gewählt. Dieses Modell bietet ein ausgewogenes Verhältnis zwischen Rechenaufwand und Qualität der Repräsentation. Die erzeugten Dokument-Embeddings wurden mit FAISS indexiert und lokal gespeichert, um eine effiziente Wiederverwendung ohne erneutes Einbetten zu gewährleisten.

Diese Architektur ermöglicht eine performante und präzise Retrieval Komponente im Gesamtsystem. Semantisch relevante Dokumentabschnitte werden mittels DPR identifiziert und für die anschließende Antwortgenerierung bereitgestellt.

## 4.3 Modellbereitstellung und Inferenzumgebung

Die Modellbereitstellung und Inferenzumgebung bilden das Rückgrat der praktischen Umsetzung des RAG-Systems. Während die vorherigen Kapitel die theoretischen Grundlagen sowie die Datenaufbereitung und den Aufbau des Retrievalsystems darstellten, fokussiert dieser Abschnitt auf die konkrete Implementierung der Modellinferenz.

Ziel ist es, ein effizientes und robustes Setup bereitzustellen, das die Nutzung leistungsfähiger LLMs ermöglicht. Dabei spielt sowohl die Auswahl des Modells

---

<sup>3</sup><https://github.com/facebookresearch/faiss>, Zugriff am 19.09.2025

<sup>4</sup><https://huggingface.co/BAAI/bge-small-en-v1.5>, Zugriff am 19.09.2025

als auch die Umgebung, in der das Modell ausgeführt wird, eine zentrale Rolle für die Performance und Skalierbarkeit des Gesamtsystems. In diesem Zusammenhang werden im Folgenden die Modellarchitektur von Mistral-7B sowie die lokale Bereitstellung über die Inferenzplattform Ollama vorgestellt und deren Integration in die RAG-Pipeline erläutert.

### 4.3.1 Mistral-7B

Mistral-7B ist ein aktuelles Beispiel für die jüngste Generation effizienter und leistungsstarker LLMs, die unter einer Open Source Lizenz verfügbar sind. Das Modell zeichnet sich durch den Einsatz fortschrittlicher Techniken wie Grouped Query Attention (GQA) für eine schnellere Inferenz und Sliding Window Attention (SWA) zur effizienten Verarbeitung längerer Tokensequenzen aus. Diese architektonischen Neuerungen ermöglichen es Mistral-7B, eine starke Performance in Bereichen wie Schlussfolgerung, Mathematik und Codegenerierung hinzulegen (vgl. Jiang u. a. 2023).

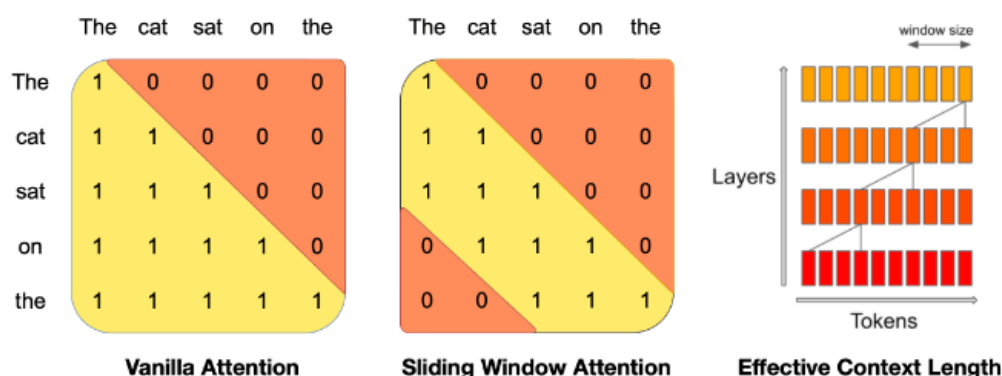


Abb. 4.1: Sliding Window Attention (SWA) (in Jiang u. a. (2023), Abb. 1)

SWA ist eine der Schlüsselinnovationen von Mistral-7B. Bei herkömmlichen Modellen steigt der Rechenaufwand stark an, wenn sehr lange Texte verarbeitet werden sollen. SWA begegnet diesem Problem, indem das Modell jeweils nur einen begrenzten Ausschnitt des Textes betrachtet. Formal ausgedrückt wird die Aufmerksamkeitsspanne auf ein festes Fenster der Größe  $W$  um das aktuelle Token beschränkt. Das bedeutet, das Token an Position  $i$  in Schicht  $k$  ( $h_i^k$ ) beachtet nur die *Hidden States* der vorherigen Schicht im Bereich von  $i-W$  bis  $i$ . Hierauf wird rekursiv auf Tokens über mehrere Schichten und einer fixierten Distanz (*Window Size*) zugegriffen, wie in Abb. 4.1 illustriert. Dies ermöglicht eine erheblich effizientere Verarbeitung langer Sequenzen, ohne die Fähigkeit zur Erfassung weit entfernter Abhängigkeiten stark einzuschränken (ebd.).

Aufgrund seiner ausgewogenen Balance aus Leistungsfähigkeit und Effizienz,

insbesondere bei der Verarbeitung längerer Kontexte durch SWA, wird Mistral-7B im Rahmen dieser Arbeit als LLM für das RAG-System eingesetzt.

### 4.3.2 Ollama

Für die Bereitstellung und Inferenz des Mistral-7B Modells wurde die Plattform Ollama<sup>5</sup> verwendet. Ollama ermöglicht die lokale Ausführung großer Sprachmodelle in einer kontrollierten und reproduzierbaren Umgebung. Durch eine einfache Schnittstelle können verschiedene Modelle effizient geladen und abgefragt werden, was insbesondere für Anwendungen mit Anforderungen an Datenschutz und niedrige Latenz von Vorteil ist.

Da im Rahmen dieser Arbeit kein Grafikkartenprozessor mit einer Compute Unified Device Architecture (CUDA) (vgl. Chakrabarti u. a. 2012) zur Verfügung stand, erfolgte die Inferenz vollständig auf der zentralen Prozesseinheit. Ollama erwies sich hierbei als besonders geeignet, da es eine vergleichsweise ressourcenschonende Ausführung von Modellen wie Mistral-7B auch auf zentralen Prozessoren ermöglicht. Trotz der erhöhten Laufzeiten konnte die Inferenz zuverlässig und stabil für eine geeignete Menge an Strichproben durchgeführt werden.

Die lokale Bereitstellung mit Ollama gewährleistet zudem eine direkte Kontrolle über die Modellversion und die Hardware Ressourcen. Dies minimiert externe Abhängigkeiten und ermöglicht eine skalierbare Inferenz, ohne auf cloudbasierte Dienste oder API-basierte Modelle angewiesen zu sein. In Kombination mit dem Retrieval-System bildet Ollama so die Grundlage für eine konsistente und reproduzierbare Antwortgenerierung im RAG-Prozess.

Diese Kombination aus Mistral-7B und Ollama stellt somit eine praktikable, performante und datenschutzfreundliche Lösung für die Inferenz von LLMs im Kontext dieser Arbeit dar.

## 4.4 RAG-Architektur

Zur Einbettung des entwickelten Systems in den Gesamtkontext erfolgt im Folgenden eine Darstellung (siehe Abb. 4.2) der zugrunde liegenden Systemarchitektur. Diese zeigt, wie zentrale Komponenten, etwa semantisches Retrieval und das textgenerierende Sprachmodelle, orchestriert werden, um eine domänenspezifische Beantwortung von Nutzeranfragen im Finanzkontext zu ermöglichen.

---

<sup>5</sup><https://ollama.com/library/mistral>, Zugriff am 19.09.2025

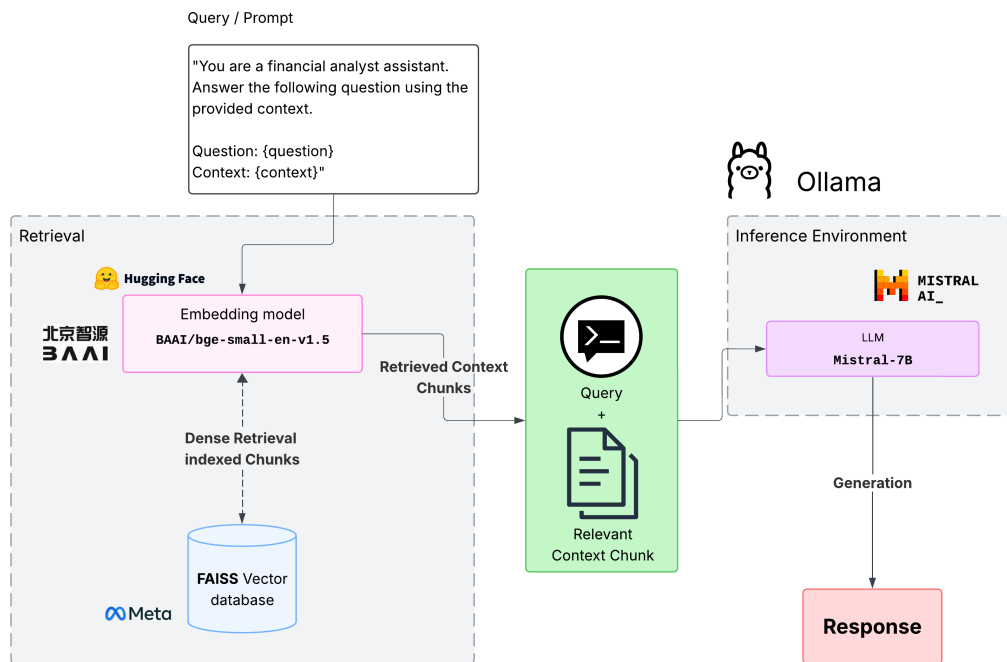


Abb. 4.2: RAG-Architektur (eigene Darstellung)

Der Prozess beginnt mit der Eingabe einer Nutzerfrage (Prompt oder Query), welche mittels eines vordefinierten Prompt Templates in eine Anfrage an das System überführt wird. Zunächst wird das Retrieval Modul aktiviert: Das Embedding-Modell `BAAI/bge-small-en-v1.5` wandelt die Frage in einen Vektor um, der in der Vektordatenbank (FAISS) genutzt wird, um die semantisch relevantesten Dokumentabschnitte zu identifizieren. Diese kontextbezogenen Dokumente werden anschließend zusammen mit der ursprünglichen Frage in einem kombinierten Prompt (Query und Relevante Kontextsegmente) an das LLM (hier Mistral-7B) in der Ollama Inferenzumgebung übergeben. Das LLM generiert daraufhin eine Antwort, die sowohl auf der Frage als auch auf den abgerufenen Dokumenten basiert. Schließlich wird die generierte Antwort zurück an den Nutzer ausgegeben. Dieses Zusammenspiel von Retrieval und Generierung soll im Rahmen eines Vergleichsdesigns im nachfolgenden Kapitel evaluiert werden, um das implementierte RAG-System gegenüber einem rein generativen Ansatz ohne externe Kontextanbindung zu bewerten.

## 5 Evaluation und Analyse

Das methodische Vorgehen dieser Arbeit ist darauf ausgelegt, eine robuste empirische Basis für die Beantwortung der Forschungsfragen zu schaffen und die Effektivität von RAG im Finanzkontext valide und quantifizierbar zu beurteilen. Grundlage der Bewertung bilden die in Abschnitt 2.5.3 beschriebenen Evaluierungsmetriken, die sowohl die Leistung des Retrievals als auch die Qualität der generierten Antworten erfassen.

Im Folgenden wird zunächst der experimentelle Testaufbau beschrieben, mit dem die Effektivität der entwickelten Lösung im Vergleich zur reinen Modellgenerierung evaluiert wird. Anschließend erfolgt eine systematische Anwendung der Evaluationsmetriken mit detaillierter Darstellung und Diskussion der Ergebnisse. Abschließend werden die Erkenntnisse in einer Post-hoc Analyse kritisch reflektiert und methodische Limitationen aufgezeigt.

### 5.1 Testaufbau und Experimentdesign

Für den experimentellen Testaufbau werden die Frage-Antwort-Generierung sowie das Kontextretrieval über das lokal bereitgestellte LLM Mistral-7B in der Ollama Inferenzumgebung und die indexierte FAISS Vektorbank mit dem BGE-Modell realisiert. Die Evaluation erfolgt in einem kontrollierten Testing Framework mit zwei distinkten Betriebsmodi:

- **LLM-only Modus:** Reine Sprachgenerierung ohne externen Kontext als Baseline.
- **RAG Modus:** Generierung mithilfe der Integration relevanter Dokumentabschnitte aus dem Retrievalprozess.

Die nachfolgend verwendeten Prompt Templates folgen einem Zero-Shot-Ansatz (siehe Abschnitt 2.2.3), bei dem das Modell ohne vorherige Beispiele oder Few-Shot-Demonstrationen arbeitet. Diese Designentscheidung gewährleistet eine faire Vergleichsbasis beider Modi und eliminiert potentielle Verzerrungen durch exemplarische Vorgaben.

## LLM-only Prompt Template

Der minimalistische Baseline-Ansatz stellt dem Modell lediglich die Frage mit einer grundlegenden Rollenspezifikation zur Verfügung:

```
You are a financial analyst assistant.  
Answer the following financial question concisely  
and accurately.
```

```
Question: {question}
```

Dieser reduzierte Prompt dient als Kontrollbedingung und ermöglicht die isolierte Bewertung der reinen Generationsfähigkeiten des Modells ohne externe Informationsquellen.

## RAG Prompt

Der RAG-Ansatz erweitert den Zero-Shot-Prompt um einen strukturierten Kontextbereich, der die vom Retrieval-System ermittelten relevanten Textpassagen systematisch einbettet:

```
You are a financial analyst assistant.  
Answer the following question using the provided  
context.
```

```
Question: {question}  
Context: {context}
```

Die explizite Instruktion zur Kontextnutzung soll eine konsistente Anwendung der bereitgestellten Informationen gewährleisten und das Risiko kontextunabhängiger Halluzinationen minimieren.

## Datenverarbeitung und Batchprozess

Die Antworten werden in parallelisierten Batches generiert, wobei je nach experimenteller Bedingung entweder die rein generative oder die retrievalgestützte Variante verwendet wird. Im RAG-Modus erfolgt zunächst der Abruf der  $k$  relevantesten Dokumente über semantische Ähnlichkeitssuche, die anschließend als strukturierter Kontext in das Prompt eingebunden werden. Zur Verwaltung der Ein- und Ausgabedaten kommt eine eigens definierte `pydantic` Klasse in Python zum Einsatz, die alle Attribute der Datenstruktur umfasst. Dies erleichtert die Nachverfolgung, systematische Auswertung und Reproduzierbarkeit der Experimente.

Für Robustheit und Nachvollziehbarkeit werden die Ergebnisse kontinuierlich in JSON-Dateien persistiert, während ein umfassendes Logging-Framework den gesamten Evaluationsprozess überwacht, Fehler protokolliert und Fortschritte dokumentiert. Ein exemplarischer Ausschnitt aus der JSON-Datei mit allen relevanten Informationen pro Ticker und Frage-Antwort-Paar ist nachfolgend dargestellt:

```
{
  "ticker": "TSLA",
  "question": "What new production locations and vehicle models were
  ↪ active in 2023?",
  "gt_answer": "In 2023, the active production locations... ",
  "gt_ctx": "The following is a summary of the status of
  ↪ production... ",
  "rag_response": "Based on the provided context, in 2023, new
  ↪ production locations for electric vehicles... ",
  "no_rag_response": "I don't have real-time data... ",
  "retrieved_ctx": [ "As we work to transition our entire
  ↪ pickup-and-delivery fleet...", "..." ]
}
```

Eine separate JSON-Struktur wurde entwickelt, die den exakten Vorgaben des RAGAS-Frameworks zur Bewertung des Faithfulness-Scores entspricht. Diese Struktur ermöglicht eine standardisierte Evaluation mit etablierten LLMs wie GPT-3.5 und Bewertungsverfahren auf Basis von RAGAS Metriken wie die Faithfulness. Nachfolgendes Beispiel zeigt die Struktur mit erweiterten Schlüsselwertpaaren und einem perfekten Faithfulness-Score von 1:

```
{
  "user_input": "In which markets does NIKE operate its Converse
  ↪ brand, and what products are included?",
  "retrieved_contexts": [
    "Our wholly-owned subsidiary brand, Converse, headquartered in
    ↪ Boston, Massachusetts, designs, distributes and..."
  ],
  "response": "The Converse brand operates its products in markets
  ↪ worldwide. According to the provided context, ...",
  "reference": "In the markets where it operates, the Converse brand
  ↪ by NIKE designs, ...",
  "faithfulness": 1.0
}
```

Diese standardisierten JSON-Formate dienen als methodische Grundlage für die nachfolgende quantitative Bewertung und qualitative Analyse des RAG-Systems sowie seiner Teilkomponenten aus Retrieval und Generation.

## 5.2 Evaluation der Retrievalkomponente

Für die Bewertung der Retrievalkomponente wurden 3,400 Stichproben aus den Ergebnissen der Testaufbaustruktur über verschiedene  $k$ -Werte hinweg analysiert, wobei  $k \in \{3, 5, 10, 15, 20, 30\}$  die Anzahl der zurückgegebenen Dokumente pro Abfrage definiert. Die zentrale Zielgröße ist dabei das Wiederauffinden relevanter Kontexte zur Beantwortung der jeweiligen Frage.

Neben klassischen IR Metriken wie  $Recall@k$ ,  $Precision@k$ ,  $F1@k$  und  $MRR$  wurde zusätzlich die durchschnittliche semantische Ähnlichkeit ( $Avg. Sim$ ) der zurückgegebenen Kontexte zur Ground Truth referenziert. Die Evaluation basiert auf einem Ähnlichkeitsschwellenwert von 0,72, um ein Dokument als relevant zu klassifizieren.

<b>k</b>	<b>Recall@k</b>	<b>Precision@k</b>	<b>F1@k</b>	<b>MRR</b>	<b>Avg. Sim</b>
3	0,810	<b>0,671</b>	<b>0,713</b>	0,767	<b>0,756</b>
5	0,850	0,642	0,696	0,775	0,749
10	0,887	0,605	0,667	0,781	0,739
15	0,897	0,578	0,643	0,781	0,733
20	<b>0,903</b>	0,559	0,626	<b>0,782</b>	0,729
30	0,903	0,526	0,595	0,782	0,723

Tabelle 5.1: Retrievalevaluierung für verschiedene Top- $k$  Werte

Die Ergebnisse (siehe Tabelle 5.1) zeigen eine erwartbare, aber dennoch aufschlussreiche Tendenz: Mit steigender Anzahl zurückgegebener Dokumente steigt der  $Recall@k$  monoton an, während die  $Precision@k$  erwartungsgemäß abnimmt. Diese inverse Beziehung reflektiert den klassischen Trade-off zwischen Precision und Recall in IR-Systemen (vgl. Manning, Raghavan und Schütze 2012, S. 158) (siehe Abbildung 5.1). Den besten F1-Wert erreicht der Wert  $k = 3$ , was auf eine insgesamt ausgewogene Rückgabequalität hinweist. Gleichzeitig liefern höhere  $k$ -Werte wie  $k = 20$  oder  $k = 30$  einen maximalen  $Recall@k$  von 0,903, jedoch auf Kosten der Genauigkeit und durchschnittlichen Kosinushnlichkeit.

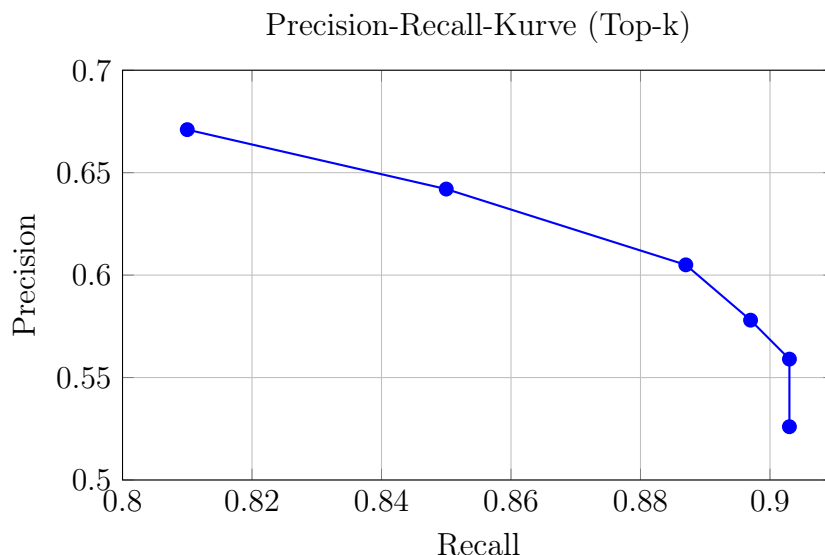


Abb. 5.1: Precision-Recall-Kurve für verschiedene Top- $k$  Werte

Die Standardabweichung der Ähnlichkeitswerte bleibt über alle  $k$ -Werte hinweg konstant, was auf eine stabile und konsistente Leistung des BGE Embedding-Modells in Bezug auf semantische Kohärenz hinweist (siehe Tabelle 5.2). Diese geringe Variabilität deutet darauf hin, dass das Retrieval konsistente Ergebnisse liefert und nicht stark von einzelnen Ausreißern beeinflusst wird.

$k$	Std(Avg. Sim)
3	0,077
5	0,074
10	0,074
15	0,074
20	0,073
30	0,073

Tabelle 5.2: Standardabweichung der semantischen Ähnlichkeit pro  $k$

### Analyse der Retrievalleistung

Aus den empirischen Ergebnissen lassen sich folgende strategische Erkenntnisse ableiten:

- **Für maximale Effizienz ( $k=3$ ):** Die Konfiguration  $k=3$  erreicht die höchste Precision (0,671) und den besten F1-Score (0,713) bei gleichzeitig maximaler semantischer Ähnlichkeit (0,756). Diese Konfiguration eignet sich optimal für Anwendungen, bei denen Genauigkeit Priorität über Vollständigkeit hat und Rechenressourcen begrenzt sind.
- **Für ausgewogene Performance ( $k=5$ ):** Mit  $k=5$  werden bereits

hohe Recall Werte bei deutlich geringerer Kontextmenge erreicht, was diese Konfiguration als praktikablen Kompromiss für produktive RAG-Anwendungen nahelegt.

- Höhere  $k$ -Werte wie  $k = 20$  oder  $k = 30$  liefern zwar maximale Abdeckung, sind jedoch mit einem signifikanten Rückgang der Precision verbunden.

### 5.3 Evaluation der Generationkomponente

Für die systematische Bewertung der Generierungskomponente wurden verschiedene etablierte NLG Metriken angewendet, um sowohl die lexikalische Übereinstimmung mit der Ground Truth als auch die semantische Qualität der generierten Antworten multidimensional zu erfassen.

Die Evaluation erfolgte auf Basis der im Testaufbau generierten Daten, wobei vier verschiedene Vergleichskonfigurationen für eine umfassende Analyse der Antwortqualität implementiert wurden:

1. **RAG vs. Ground Truth Antwort:** Direkte Bewertung der RAG-generierten Antworten gegen Referenzantworten im QA-Datensatz.
2. **LLM-only vs. Ground Truth Antwort:** Baseline Vergleich für reine Modellgenerierung.
3. **RAG vs. Ground Truth Kontext:** Bewertung der Kontextnutzung.
4. **RAG vs. LLM-only:** Direkter Systemvergleich zur Quantifizierung des RAG-Mehrwerts.

#### Quantitative Ergebnisse der NLG Metriken

Vergleich	ROUGE-L	BLEU	F1	Sem. Sim
RAG vs. GT Antwort	0,180	0,067	0,259	0,163
LLM-only vs. GT Antwort	0,137	0,036	0,200	0,120
RAG vs. GT Kontext	<b>0,235</b>	<b>0,091</b>	<b>0,355</b>	<b>0,226</b>
RAG vs. LLM-only	0,216	0,084	0,354	0,219

Tabelle 5.3: Evaluation der Generierungsqualität

#### Analyse der Generierungsleistung

Der direkte Systemvergleich zur Antwortqualität zwischen RAG- und LLM-only-Ansatz gegenüber der Referenzantwort (RAG vs. Ground Truth Antwort, LLM-only vs. Ground Truth Antwort) zeigt konsistente und signifikante Verbesserungen durch den RAG-Ansatz über alle evaluierten Metriken hinweg (siehe

Tabelle 5.3):

- **F1-Score:** +29,5% Verbesserung (0,200 → 0,259)
- **ROUGE-L:** +31,4% Verbesserung (0,137 → 0,180)
- **BLEU:** +86,1% Verbesserung (0,036 → 0,067)
- **Semantische Ähnlichkeit:** +35,8% Verbesserung (0,120 → 0,163)

Die Ergebnisse zeigen signifikante Leistungsunterschiede zwischen den getesteten Vergleichskonfigurationen. Besonders hervorzuheben ist die Konstellation RAG vs. Ground Truth Kontext, die in allen Evaluierungsmetriken die höchsten Werte erzielt. Dies deutet darauf hin, dass das RAG-System in der Lage ist, relevante kontextuelle Informationen präzise zu extrahieren, sinnvoll zu verarbeiten und kohärent in die Antwortgenerierung einzubinden. Daraus lässt sich schließen, dass die generierten Antworten in hohem Maße mit dem zugrunde liegenden Kontext übereinstimmen und somit nur ein geringes Maß an Halluzinationen aufweisen.

Metrik	LLM Median	RAG Median	Verbesserung	High Perf. RAG [%]
ROUGE-L	0,132	0,154	+16,7 %	0,0
BLEU	0,000	0,034	+∞	3,2
F1	0,212	0,248	+17,0 %	10,8
Sem. Sim	0,118	0,142	+20,3 %	0,4

Tabelle 5.4: Medianvergleich der Generierungsqualität zwischen LLM und RAG

Die in Tabelle 5.4 präsentierten Medianwerte verdeutlichen die systematischen Leistungsgewinne durch den Einsatz von RAG im Vergleich zum LLM-only-System. In sämtlichen Metriken zeigt RAG verbesserte Ergebnisse, wobei die größten relativen Zuwächse bei der semantischen Ähnlichkeit (plus 20,3 %) und beim F1-Score (plus 17,0 %) auftreten. Dies spricht für inhaltlich relevantere und besser abgeglichene Antworten. Auch beim BLEU-Score, der beim Basismodell keinerlei nennenswerte Übereinstimmungen aufweist, erreicht RAG erstmals verwertbare Werte. Die nach wie vor niedrigen BLEU-Werte deuten allerdings auf eine insgesamt geringe n-Gramm-Übereinstimmung hin, was bei freier Textgenerierung wenig überraschend ist und die Aussagekraft dieser Metrik in diesem Kontext einschränkt. Der Anteil sogenannter High-Performance-Samples bleibt insgesamt niedrig, zeigt jedoch beim F1-Score mit 10,8 % eine auffällige Konzentration besonders guter Ausgaben. Insgesamt zeigt sich, dass RAG die inhaltliche Qualität der Antworten konsistent verbessert, insbesondere im Hinblick auf semantische Kohärenz. Gleichzeitig besteht weiterhin Optimierungspotenzial, etwa bei der sprachlichen Oberflächenstruktur und der Erhöhung der Zahl sehr starker Einzelantworten.

## Faithfulness Evaluation mit RAGAS

Zur Bewertung der inhaltlichen Treue der generierten Antworten zum jeweiligen Kontext wurde das etablierte RAGAS-Framework im Rahmen einer LLM-as-a-Judge Evaluation mit GPT-3.5 eingesetzt. Ziel dieser Analyse war es, systematisch zu bewerten, inwieweit das RAG-System ausschließlich auf den übergebenen Kontext referenziert und keine halluzinierten oder faktisch inkorrekte Inhalte generiert. Die Auswertung basiert auf etwa 650 stichprobenartig generierten Antworten, wobei jeder Eintrag im Hinblick auf seine Faithfulness mit einem Skalenwert zwischen 0 (keine Kontexttreue) und 1 (vollständige Kontexttreue) bewertet wurde.

Kennzahl	Wert
Durchschnittlicher Score	0,696
Median	0,800
Standardabweichung	0,311
Minimum	0,000
Maximum	1,000

Tabelle 5.5: Deskriptive Statistik der Faithfulness

Faithfulness Kategorie	Anteil
vollständige Kontexttreue (1,0)	31,3 %
partielle Kontexttreue (0,5)	11,3 %
keine Kontexttreue (0,0)	7,8 %
<b>Problematische Samples (&lt; 0,5)</b>	<b>20,1 %</b>

Tabelle 5.6: Kategoriale Verteilung der Faithfulness-Scores

Ein Drittel der generierten Antworten (31,3 %) erreichte den Maximalwert von 1,0 und gilt damit als vollständig kontexttreu. Weitere 11,3 % der Antworten erreichten zumindest einen mittleren Score von exakt 0,5. Kritisch zu bewerten sind 20,1 % der Samples, die Scores unterhalb von 0,5 erreichten, was auf eine partielle oder vollständige Kontextabweichung hinweist. Der Anteil komplett untreuer Antworten (Score = 0,0) beträgt 7,8 %.

Die Auswertungen mit GPT-3.5 als LLM-as-a-judge zeigen eine insgesamt moderate bis gute Faithfulness Leistung. Insbesondere die hohe Mediantreue

von 0,800 spricht für eine überwiegend korrekte Verarbeitung der übergebenen Kontexte.

## 5.4 Post-hoc Analyse

Im Rahmen einer umfassenden qualitativen Nachanalyse der generierten Antworten konnten wiederkehrende Fehlerarten identifiziert werden, die in den automatisierten Evaluationsmetriken nur unzureichend oder verzerrt abgebildet werden. Diese systematischen Probleme betreffen insbesondere Fragestellungen mit numerischen, temporalen oder faktisch differenzierten Komponenten, welche in der Finanzdomäne eine zentrale Rolle spielen.

### Kategorisierung der Fehlertypen

- Semantische Verwechslungen bei temporalen Kontexten: Das Modell erkennt zeitlich benachbarte, aber sachlich distinkte Informationen nicht zuverlässig als unzutreffend. Dies manifestiert sich besonders bei Jahresvergleichen, Quartalsdaten verschiedener Perioden und historischen gegenüber aktuellen Kennzahlen.
- Numerische Präzisionsfehler: Quantitative Angaben werden trotz korrekter Berechnung manchmal inkorrekt kontextualisiert, z.B durch falsche Zuordnung von Prozentwerten zu Segmenten oder Verwechslung ähnlicher Kennzahlen.
- Bewertungsverzerrungen durch Referenzqualität: Korrekte Modellantworten werden aufgrund ungenauer Referenzdaten fälschlich als problematisch klassifiziert. Die geschieht bei unvollständigen Ground Truth Antworten, kontextuellen Mehrdeutigkeiten in den Referenzdaten oder systematischen Unterschieden in der Antwortformulierung.

Während einige Fehlertypen auf tatsächliche Limitationen des Modells hinweisen, reflektiert die Bewertungsverzerrung durch Referenzqualität vor allem eine Schwäche der automatisierten Bewertungsverfahren, die faktische Korrektheit nicht immer angemessen erfassen. Im Folgenden werden exemplarisch zwei konkrete Fälle dargestellt, die die beschriebenen Fehlerkategorien verdeutlichen.

### Fallstudie 1: Temporale Verwechslung

Das folgende Beispiel zeigt eine Verwechslung zeitlich benachbarter Kontexte aufgrund semantischer Ähnlichkeit. Hierbei wird deutlich, wie das Modell sachlich unzutreffende, aber formal passende Textstellen in die Antwort integriert.

```
{
  "ticker": "NVDA",
  "question": "What percentage of the global workforce was female at
  ↪ the end of fiscal year 2023?",
  "rag_response": "Based on the provided context, it states that as
  ↪ of December 31, 2024, ..."
}
```

Obwohl der relevante Kontext zur Frage im Retrieval korrekt enthalten war, bezieht sich die Antwort fälschlich auf einen anderen Zeitpunkt (2024 statt 2023). Solche temporalen Fehler wiegen im Finanzkontext besonders schwer, da sich Kennzahlen oft jährlich stark verändern. Gleichzeitig zeigen sie die Schwächen rein semantischer Bewertungen auf, die auf solche feingranularen Unterschiede nicht ausreichend reagieren.

## Fallstudie 2: Bewertungsverzerrung trotz korrekter Antwort

Das nächste Beispiel zeigt, wie eine sachlich korrekte Antwort durch die Bewertungsmetrik fälschlich als untreu eingestuft wird, weil die korrekt erscheinende Modellantwort zu stark vom tatsächlich abgerufenen Kontext abweicht.

```
{
  "user_input": "Which reportable segment experienced a net sales
  ↪ increase in 2023 and by what percentage did this increase?",
  "retrieved_contexts": [
    "...Net sales ... 2023: $949,2 million ... 2022: $853,4 million
    ↪ ... 11.2% increase."
  ],
  "response": "Based on the provided context, [...] net sales
  ↪ increase International Segment [...] The percentage increase
  ↪ [...] can be calculated as:  $(949.2 - 853.4) / 853.4 = 11.2\%$ .",
  "reference": "The Rest of Asia Pacific segment experienced a net
  ↪ sales increase of 1% in 2023.",
  "faithfulness":0.5
}
```

In diesem Fall hat das Modell eine sachlich korrekte Antwort generiert: Es berechnet aus den gegebenen Kontextzahlen korrekt eine Umsatzsteigerung von 11,2% gegenüber dem Vorjahr. Dennoch wurde die Antwort im automatisierten Evaluationsprozess mit einem Faithfulness-Score von lediglich 0,5 bewertet.

Grund hierfür ist die Funktionsweise der Faithfulness-Metrik: Sie bewertet, inwieweit die Antwort direkt aus den abgerufenen Kontexten ableitbar ist und erwartet eine hohe Übereinstimmung mit der Referenz oder den exakten Kontextformulierungen.

Dieses Beispiel verdeutlicht eine zentrale Einschränkung der automatisierten Faithfulness-Metrik: Selbst inhaltlich richtige und nachvollziehbare Antworten können als teilweise untreu eingestuft werden, wenn sie nicht wortwörtlich oder exakt in der Formulierung den Kontext widerspiegeln. Die qualitative Nachanalyse zeigt somit die systematische Herausforderung: Semantische Korrektheit und numerische Genauigkeit werden von der Metrik nicht immer vollständig erfasst, was die Notwendigkeit einer kritischen und reflektierten Evaluationspraxis unterstreicht.

## 6 Ergebnisse und Diskussion

Die durchgeführte Evaluation des RAG-Systems im Finanzkontext liefert mehrere zentrale Erkenntnisse zur Effektivität retrieval-augmentierter Textgenerierung. Die Ergebnisse zeigen konsistente Verbesserungen gegenüber der reinen Sprachgenerierung bei gleichzeitiger Identifikation spezifischer Herausforderungen in der Finanzdomäne.

Die Gegenüberstellung von RAG- und LLM-basierten Antworten im Rahmen der quantitativen Evaluation zeigt eine signifikante Verbesserung der Antwortqualität durch die Integration des Retrievalschritts. Insbesondere die Zunahme des F1-Scores verdeutlicht, dass die Kombination von Retrieval und Generierung inhaltlich präzisere und vollständigere Antworten ermöglicht. Der direkte Systemvergleich bestätigt die Hypothese einer verbesserten Antwortqualität durch RAG. Über alle NLG Metriken hinweg zeigen sich konsistente Verbesserungen.

Die Evaluation der Retrievalkomponente über verschiedene k-Werte hinweg zeigt eine erwartete Recall-Precision-Trade-off Charakteristik. Mit  $k=3$  wird das optimale Gleichgewicht zwischen Präzision (0,671) und F1-Score (0,713) erreicht, während höhere k-Werte zwar maximalen Recall (0,903 bei  $k=20/30$ ) liefern, aber deutliche Präzisionsverluste zur Folge haben. Die konstante Standardabweichung der semantischen Ähnlichkeit (0,074) über alle k-Werte hinweg deutet auf eine stabile Retrievalqualität hin.

Ein wesentliches Ziel der Integration des Retrievalmoduls war die Erhöhung der Treue zu den Fakten der angebundenen Wissensquelle. Die RAGAS-basierte Faithfulness Evaluation mit GPT-3.5 ergab einen durchschnittlichen Score von 0,696 bei einem Median von 0,800, was auf eine robustere Leistung trotz Ausreißern hindeutet. Während 31,3% der Antworten vollständige Kontexttreue (Score 1,0) erreichten, zeigten 20,1% der Samples problematische Abweichungen (Score  $< 0,5$ ). Dies deutet auf eine insgesamt moderate bis gute, aber nicht durchgängig zuverlässige Kontextverarbeitung hin. Dies unterstreicht das Potenzial externer Kontexte zur Reduktion von Halluzinationen, verdeutlichen jedoch zugleich, dass ohne gezielte Optimierungen im Retrieval und Modellverhalten keine durchgängig verlässliche Kontexttreue gewährleistet werden kann.

Die Post-hoc-Analyse offenbart weitere charakteristische Schwächen des Systems bei finanzspezifischen Aufgabenstellungen. Temporale Verwechslungen, unvollständige quantitative Analysen und die fehlerhafte Verarbeitung ähnli-

cher Kontexte stellen domänenspezifische Herausforderungen dar, die durch rein semantische Retrievalverfahren nur unzureichend adressiert werden. Die identifizierten Diskrepanzen zwischen automatisierten Faithfulness-Scores und tatsächlicher inhaltlicher Korrektheit (siehe Fallstudie 2 in Abschnitt 5.4) verdeutlichen fundamentale Limitationen LLM-basierter Evaluationsverfahren. Auch die Bevorzugung syntaktischer Übereinstimmung gegenüber semantischer Richtigkeit kann zu systematischen Bewertungsverzerrungen führen, insbesondere in hochsensiblen Domänen wie dem Finanzbereich.

Die Evaluation basiert auf einem spezifischen Datensatz von Finanzberichten in einem spezifischen und bekannten Format (SEC 10K Filings) sowie einer begrenzten Anzahl von Testfragen. Die Generalisierbarkeit der Ergebnisse auf andere Finanzdomänen oder Dokumenttypen bleibt damit zu prüfen. Insbesondere die Übertragbarkeit und Eignung auf deutsche Finanzberichte oder alternative Berichtsformate ist ungeklärt.

Methodisch lässt sich reflektieren, dass die verwendeten Modelle (Mistral-7B, BGE) primär aufgrund ihrer Effizienz und Kompatibilität mit lokaler Prozessorinferenz gewählt wurden. Größere oder domänenspezifische Sprachmodelle (siehe Abschnitt 3.1.2) und Embedding-Modelle aus FinMTEB (siehe Abschnitt 3.4), die auf die Finanzdomäne spezialisiert sind, hätten potenziell eine höhere Performance erzielt, waren aber aus Ressourcen Gründen nicht einsetzbar. Dies stellt eine Einschränkung hinsichtlich der Generalisierbarkeit der Ergebnisse dar.

Die Ergebnisse belegen, dass RAG-Systeme bei der Verarbeitung und Beantwortung domänenspezifischer Nutzeranfragen einen substantiellen Mehrwert bieten können, insbesondere durch verbesserte Kontexttreue und präzisere Antwortgenerierung. Für den Einsatz in der Finanzbranche spricht vieles dafür, dass LLM-basierte Assistenten künftig verlässlichere Auskünfte auf komplexe Fragen geben können, sofern die RAG-Komponenten gezielt auf die Domäne abgestimmt werden. Zugleich unterstreichen die Befunde die Notwendigkeit menschlicher Supervision oder der Entwicklung domänenspezifischer Evaluationsframeworks, die die Besonderheiten fachlicher Kontexte angemessen berücksichtigen.

## 7 Fazit und Ausblick

Diese Arbeit untersuchte den Einfluss von RAG auf die Antwortqualität von LLMs in domänenspezifischen Frage-Antwort-Systemen im Finanzbereich. Im Mittelpunkt stand dabei die Frage, inwiefern sich RAG auf die Genauigkeit, Kontextrelevanz und die Reduktion von Halluzinationen in den generierten Antworten auswirkt. Die Ergebnisse zeigen deutlich, dass der gezielte Einsatz von Retrieval Komponenten zu einer signifikanten Qualitätssteigerung führt: Durch kontextgestützte Antwortgenerierung lassen sich sachlich präzisere Aussagen erzielen, der gezielte Dokumentenabruf erhöht die Relevanz der Antworten, und die Anbindung an externe Wissensquellen reduziert nachweislich die Neigung zu Halluzinationen.

Gleichzeitig wurde deutlich, dass der Nutzen von RAG in der Finanzdomäne nicht ohne domänenspezifische Herausforderungen realisierbar ist. Finanzdaten sind stark zeitbezogen, was den Einsatz temporaler Retrievalstrategien erfordert. Auch quantitative Anfragen stellen besondere Anforderungen an das System: Hier sind hybride Architekturen notwendig, die numerische Datenanalyse mit der Fähigkeit zur textuellen Erklärung verbinden. Darüber hinaus verlangen viele Finanzberichte die Verarbeitung multimodaler Inhalte wie Tabellen, Diagramme oder strukturierte Berichtsdaten, ein Aspekt, der in heutigen RAG-Systemen noch unzureichend abgebildet wird.

Ein zentrales Ergebnis betrifft zudem die Qualitätssicherung: Herkömmliche Evaluationsmetriken stoßen in hochsensiblen, regulierten Domänen wie dem Finanzsektor an ihre Grenzen. Es bedarf daher domänenspezifischer Metriken, die über reine Textkohärenz hinaus auch sachliche Korrektheit und numerische Konsistenz bewerten können. Bis robuste automatische Verifikationssysteme existieren, bleibt menschliche Kontrolle ein integraler Bestandteil der Qualitätssicherung.

Für die Weiterentwicklung von RAG-Systemen im Finanzbereich ergeben sich mehrere zentrale Perspektiven. Dazu zählt insbesondere die Integration adaptiver Lernmechanismen, die Rückmeldungen aus der Anwendung systematisch zur Optimierung nutzen. Domänenspezifisches Fine Tuning in Kombination mit präzisiertem, semantisch sensibilisiertem Retrieval etwa durch Cross Encoder basierte Reranking Methoden verspricht weiteres Verbesserungspotenzial. Langfristig könnten solche Systeme die Grundlage intelligenter Finanzassistenten

bilden, die komplexe Analyseaufgaben unterstützen und gleichzeitig regulatorischen Anforderungen gerecht werden. Die nahtlose Einbindung in Compliance Prozesse und die Sicherstellung von Nachvollziehbarkeit und Transparenz sind dabei unerlässliche Voraussetzungen.

Insgesamt zeigt die Arbeit, dass RAG ein vielversprechender Ansatz ist, um die Antwortqualität von LLMs in datenintensiven Domänen wie dem Finanzwesen signifikant zu verbessern. Voraussetzung dafür ist jedoch, dass technische, inhaltliche und regulatorische Herausforderungen gezielt adressiert werden. Die Finanzindustrie steht somit vor der Aufgabe, dieses Potenzial verantwortungsvoll und unter Wahrung höchster Anforderungen an Genauigkeit, Transparenz und Risikomanagement zu erschließen.

## Literaturverzeichnis

- Bagattini, Giulio, Brière, Marie, Guagliano, Claudia, Maple, Carsten und Sabuncuoglu, Alpay (2024). *Leveraging Large Language Models in Finance: Pathways to Responsible Adoption*. Techn. Ber. European Securities and Markets Authority (ESMA); Institut Louis Bachelier; The Alan Turing Institute. URL: [https://cdn.prod.website-files.com/672cea0ae7889396005b1e87/6835e29175ba44c55e9b4def\\_LLMs\\_in\\_finance\\_-\\_ILB\\_ESMA\\_Turing-Report-19\\_05-GB-FB-GB-FB.pdf](https://cdn.prod.website-files.com/672cea0ae7889396005b1e87/6835e29175ba44c55e9b4def_LLMs_in_finance_-_ILB_ESMA_Turing-Report-19_05-GB-FB-GB-FB.pdf) (besucht am 19.09.2025).
- Brown, Tom B., Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared, Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, Agarwal, Sandhini, Herbert-Voss, Ariel, Krueger, Gretchen, Henighan, Tom, Child, Rewon, Ramesh, Aditya, Ziegler, Daniel M., Wu, Jeffrey, Winter, Clemens, Hesse, Christopher, Chen, Mark, Sigler, Eric, Litwin, Mateusz, Gray, Scott, Chess, Benjamin, Clark, Jack, Berner, Christopher, McCandlish, Sam, Radford, Alec, Sutskever, Ilya und Amodei, Dario (2020). „Language Models are Few-Shot Learners“. In: *Advances in Neural Information Processing Systems* 33, S. 1877–1901. URL: <https://arxiv.org/abs/2005.14165>.
- Chakrabarti, Gautam, Grover, Vinod, Aarts, Bastiaan, Kong, Xiangyun, Kudlur, Manjunath, Lin, Yuan, Marathe, Jaydeep, Murphy, Mike und Wang, Jian-Zhong (2012). „CUDA: Compiling and optimizing for a GPU platform“. In: *Procedia Computer Science* 9. Proceedings of the International Conference on Computational Science, ICCS 2012, S. 1910–1919. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2012.04.209>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050912003304>.
- Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton und Toutanova, Kristina (2019). „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, Minnesota: Association for Computational Linguistics, S. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://aclanthology.org/N19-1423>.
- Enevoldsen, Kenneth, Chung, Isaac, Kerboua, Imene, Kardos, Márton, Mathur, Ashwin, Stap, David, Gala, Jay, Siblini, Wissam, Krzemiński, Dominik,

- Winata, Genta Indra, Sturua, Saba, Utpala, Saiteja, Ciancone, Mathieu, Schaeffer, Marion, Misra, Diganta, Dhakal, Shreeya, Rystrom, Jonathan, Solomatin, Roman, Çağatan, Ömer Veysel, Kundu, Akash, Bernstorff, Martin, Xiao, Shitao, Sukhlecha, Akshita, Pahwa, Bhavish, Poświata, Rafał, GV, Kranthi Kiran, Ashraf, Shawon, Auras, Daniel, Plüster, Björn, Harries, Jan Philipp, Magne, Loïc, Mohr, Isabelle, Zhu, Dawei, Gisserot-Boukhlef, Hippolyte, Aarsen, Tom, Kostkan, Jan, Wojtasik, Konrad, Lee, Taemin, Suppa, Marek, Zhang, Crystina, Rocca, Roberta, Hamdy, Mohammed, Michail, Andrianos, Yang, John, Faysse, Manuel, Vatolin, Aleksei, Thakur, Nandan, Dey, Manan, Vasani, Dipam, Chitale, Pranjal A, Tedeschi, Simone, Tai, Nguyen, Snegirev, Artem, Hendriksen, Mariya, Günther, Michael, Xia, Mengzhou, Shi, Weijia, Lù, Xing Han, Clive, Jordan, K, Gayatri, Anna, Maksimova, Wehrli, Silvan, Tikhonova, Maria, Panchal, Henil Shalin, Abramov, Aleksandr, Ostendorff, Malte, Liu, Zheng, Clematide, Simon, Miranda, Lester James Validad, Fenogenova, Alena, Song, Guangyu, Safi, Ruqiyah Bin, Li, Wen-Ding, Borghini, Alessia, Cassano, Federico, Hansen, Lasse, Hooker, Sara, Xiao, Chenghao, Adlakha, Vaibhav, Weller, Orion, Reddy, Siva und Muennighoff, Niklas (2025). „MMTEB: Massive Multilingual Text Embedding Benchmark“. In: *The Thirteenth International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=z13pfz4VCV> (besucht am 19.09.2025).
- Es, Shahul, James, Jithin, Espinosa Anke, Luis und Schockaert, Steven (2024). „RAGAs: Automated Evaluation of Retrieval Augmented Generation“. In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Hrsg. von Aletras, Nikolaos und De Clercq, Orphee. St. Julians, Malta: Association for Computational Linguistics, S. 150–158. DOI: 10.18653/v1/2024.eacl-demo.16. URL: <https://aclanthology.org/2024.eacl-demo.16/>.
- Farea, Amer, Yang, Zhen, Duong, Kien, Perera, Nadeesha und Emmert-Streib, Frank (2022). „Evaluation of Question Answering Systems: Complexity of judging a natural language“. In: *arXiv preprint arXiv:2209.12617*. URL: <https://arxiv.org/abs/2209.12617>.
- Gan, Aoran, Yu, Hao, Zhang, Kai, Liu, Qi, Yan, Wenyu, Huang, Zhenya, Tong, Shiwei und Hu, Guoping (2025). „Retrieval Augmented Generation Evaluation in the Era of Large Language Models: A Comprehensive Survey“. In: *arXiv preprint arXiv:2504.14891*. URL: <https://arxiv.org/abs/2504.14891>.
- Gao, Yunfan, Xiong, Yun, Gao, Xinyu, Jia, Kangxiang, Pan, Jinliu, Bi, Yuxi, Dai, Yi, Sun, Jiawei, Wang, Meng und Wang, Haofen (2023). „Retrieval-

- Augmented Generation for Large Language Models: A Survey“. In: *arXiv preprint arXiv:2312.10997*. URL: <https://arxiv.org/abs/2312.10997>.
- Huang, Lei, Yu, Weijiang, Ma, Weitao, Zhong, Weihong, Feng, Zhangyin, Wang, Haotian, Chen, Qianglong, Peng, Weihua, Feng, Xiaocheng, Qin, Bing und Liu, Ting (2025). „A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions“. In: *ACM Transactions on Information Systems* 43.2, S. 1–55. ISSN: 1046-8188. DOI: 10.1145/3703155.
- Iaroshev, Ivan, Pillai, Ramalingam, Vaglietti, Leandro und Hanne, Thomas (2024). „Evaluating Retrieval-Augmented Generation Models for Financial Report Question and Answering“. In: *Applied Sciences* 14.20. ISSN: 2076-3417. DOI: 10.3390/app14209318. URL: <https://www.mdpi.com/2076-3417/14/20/9318>.
- Ji, Ziwei, Lee, Nayeon, Frieske, Rita, Yu, Tiezheng, Su, Dan, Xu, Yan, Ishii, Etsuko, Bang, Ye Jin, Madotto, Andrea und Fung, Pascale (2023). „Survey of Hallucination in Natural Language Generation“. In: *ACM Computing Surveys* 55.12, S. 1–38. ISSN: 0360-0300. DOI: 10.1145/3571730.
- Jiang, Albert Q., Sablayrolles, Alexandre, Mensch, Arthur, Bamford, Chris, Chaplot, Devendra Singh, Las Casas, Diego de, Bressand, Florian, Lengyel, Gianna, Lample, Guillaume, Saulnier, Lucile, Lavaud, L elio Renard, Lachaux, Marie-Anne, Stock, Pierre, Scao, Teven Le, Lavril, Thibaut, Wang, Thomas, Lacroix, Timoth ee und Sayed, William El (2023). „Mistral 7B“. In: *CoRR* abs/2310.06825. DOI: 10.48550/ARXIV.2310.06825. ARXIV: 2310.06825. URL: <https://doi.org/10.48550/arXiv.2310.06825>.
- Jurafsky, Daniel und Martin, James H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3. Aufl. Online-Manuskript ver offentlicht am 12.01.2025. URL: <https://web.stanford.edu/~jurafsky/slp3/> (besucht am 19.09.2025).
- Karpukhin, Vladimir, O uz, Barlas, Min, Sewon, Lewis, Patrick, Wu, Le-dell, Edunov, Sergey, Chen, Danqi und Yih, Wen-tau (2020). „Dense Passage Retrieval for Open-Domain Question Answering“. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Online: Association for Computational Linguistics, S. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: <https://aclanthology.org/2020.emnlp-main.550>.
- Kim, Sejong, Song, Hyunseo, Seo, Hyunwoo und Kim, Hyunjun (2025). „Optimizing Retrieval Strategies for Financial Question Answering Docu-

- ments in Retrieval-Augmented Generation Systems“. In: *arXiv preprint arXiv:2503.15191*. URL: <https://arxiv.org/abs/2503.15191>.
- Le Ma, Zhang, Ran, Han, Yikun, Yu, Shirui, Wang, Zaitian, Ning, Zhiyuan, Zhang, Jinghan, Xu, Ping, Li, Pengjiang, Ju, Wei, Chen, Chong, Wang, Dongjie, Liu, Kunpeng, Wang, Pengyang, Wang, Pengfei, Fu, Yanjie, Liu, Chunjiang, Zhou, Yuanchun und Lu, Chang-Tien (2023). „A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge“. In: *arXiv preprint arXiv:2310.11703*. URL: <https://arxiv.org/abs/2310.11703>.
- Lewis, Patrick, Perez, Ethan, Piktus, Aleksandra, Petroni, Fabio, Karpukhin, Vladimir, Goyal, Naman, Küttler, Heinrich, Lewis, Mike, Yih, Wen-tau, Rocktäschel, Tim, Riedel, Sebastian und Kiela, Douwe (2020). „Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks“. In: *Advances in Neural Information Processing Systems*. Hrsg. von Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F. und Lin, H. Bd. 33. Curran Associates, Inc., S. 9459–9474. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf) (besucht am 19.09.2025).
- Li, Yinheng, Wang, Shaofei, Ding, Han und Chen, Hang (2023). „Large Language Models in Finance: A Survey“. In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. ICAIF '23. Brooklyn, NY, USA: Association for Computing Machinery, S. 374–382. ISBN: 9798400702402. DOI: 10.1145/3604237.3626869. URL: <https://doi.org/10.1145/3604237.3626869>.
- Lin, Chin-Yew (2004). „ROUGE: A Package for Automatic Evaluation of Summaries“. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, S. 74–81. URL: <https://aclanthology.org/W04-1013> (besucht am 19.09.2025).
- Manning, Christopher, Raghavan, Prabhakar und Schütze, Hinrich (2012). *Introduction to Information Retrieval*. Cambridge University Press. ISBN: 978-0521865715. DOI: 10.1017/CBO9780511809071.
- Maple, Carsten und Sabuncuoglu, Alpay (2024). *The Impact of Large Language Models in Finance: Towards Trustworthy Adoption*. Techn. Ber. The Alan Turing Institute. URL: [https://www.turing.ac.uk/sites/default/files/2024-06/the\\_impact\\_of\\_large\\_language\\_models\\_in\\_finance\\_-\\_towards\\_trustworthy\\_adoption\\_1.pdf](https://www.turing.ac.uk/sites/default/files/2024-06/the_impact_of_large_language_models_in_finance_-_towards_trustworthy_adoption_1.pdf) (besucht am 19.09.2025).
- Mikolov, Tomas, Chen, Kai, Corrado, Greg und Dean, Jeffrey (2013). „Efficient Estimation of Word Representations in Vector Space“. In: *arXiv preprint arXiv:1301.3781*. URL: <https://arxiv.org/abs/1301.3781>.

- Mitra, Bhaskar und Craswell, Nick (2018). „An Introduction to Neural Information Retrieval“. In: *Foundations and Trends in Information Retrieval* 13.1, S. 1–126. ISSN: 1554-0669. DOI: 10.1561/15000000061.
- Nie, Yuqi, Kong, Yaxuan, Dong, Xiaowen, Mulvey, John M., Poor, H. Vincent, Wen, Qingsong und Zohren, Stefan (2024). „A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges“. In: *arXiv preprint arXiv:2406.11903*. DOI: 10.48550/arXiv.2406.11903. URL: <https://arxiv.org/abs/2406.11903>.
- Papineni, Kishore, Roukos, Salim, Ward, Todd und Zhu, Wei-Jing (2002). „BLEU: a Method for Automatic Evaluation of Machine Translation“. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, S. 311–318. DOI: 10.3115/1073083.1073135. URL: <https://aclanthology.org/P02-1040>.
- Pennington, Jeffrey, Socher, Richard und Manning, Christopher (2014). „GloVe: Global Vectors for Word Representation“. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics, S. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: <https://aclanthology.org/D14-1162>.
- Radford, Alec, Narasimhan, Karthik, Salimans, Tim und Sutskever, Ilya (2018). *Improving Language Understanding by Generative Pre-Training*. Techn. Ber. OpenAI. URL: [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf) (besucht am 19.09.2025).
- Reimers, Nils und Gurevych, Iryna (2019). „Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks“. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Hong Kong, China: Association for Computational Linguistics, S. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: <https://aclanthology.org/D19-1410>.
- Shuster, Kurt, Poff, Spencer, Chen, Moya, Kiela, Douwe und Weston, Jason (2021). „Retrieval Augmentation Reduces Hallucination in Conversation“. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, S. 3784–3803. DOI: 10.18653/v1/2021.findings-emnlp.320. URL: <https://aclanthology.org/2021.findings-emnlp.320>.
- Singla, Alex, Sukharevsky, Alexander, Yee, Lareina, Chui, Michael und Hall, Bryce (2025). *The State of AI: How Organizations Are Rewiring to Capture Value*. Techn. Ber. McKinsey & Company. URL: <https://www.mckinsey.com>.

- com/~ /media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai/2025/the-state-of-ai-how-organizations-are-rewiring-to-capture-value\_final.pdf (besucht am 19.09.2025).
- Tang, Yixuan und Yang, Yi (2025a). „Do We Need Domain-Specific Embedding Models? An Empirical Investigation“. In: *arXiv preprint arXiv:2409.18511*. URL: <https://arxiv.org/abs/2409.18511>.
- Tang, Yixuan und Yang, Yi (2025b). „FinMTEB: Finance Massive Text Embedding Benchmark“. In: *arXiv preprint arXiv:2502.10990*. URL: <https://arxiv.org/abs/2502.10990>.
- Touvron, Hugo, Lavril, Thibaut, Izacard, Gautier, Martinet, Xavier, Lachaux, Marie-Anne, Lacroix, Timothée, Rozière, Baptiste, Goyal, Naman, Hambro, Eric, Azhar, Faisal, Rodriguez, Aurelien, Joulin, Armand, Grave, Edouard und Lample, Guillaume (2023). „LLaMA: Open and Efficient Foundation Language Models“. In: *arXiv preprint arXiv:2302.13971*. URL: <https://arxiv.org/abs/2302.13971>.
- United States Securities and Exchange Commission (o. J.). *Form 10-K*. URL: <https://www.sec.gov/files/form10-k.pdf> (besucht am 19.09.2025).
- Uszkoreit, Jakob (2017). *Transformer: A Novel Neural Network Architecture for Language Understanding*. Google Research Blog. URL: <https://research.googleblog.com/2017/08/transformer-novel-neural-network.html> (besucht am 19.09.2025).
- Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz und Polosukhin, Illia (2017). „Attention Is All You Need“. In: *arXiv preprint arXiv:1706.03762*. ARXIV: 1706.03762.
- Wartena, Christian (2022). „On the Geometry of Concreteness“. In: *Proceedings of the 7th Workshop on Representation Learning for NLP*, S. 204–212. DOI: 10.25968/opus-2299.
- Wartena, Christian (2024). „Generative AI is not Magic“. In: *KI-Forum 2024: AI 4 Students - AI in Research - AI Showroom*, S. 10–13. ISBN: 978-3-69018-002-3. DOI: 10.25968/opus-3450.
- Wu, Shijie, Irsoy, Ozan, Lu, Steven, Dabrovolski, Vadim, Dredze, Mark, Gehrmann, Sebastian, Kambadur, Prabhanjan, Rosenberg, David und Mann, Gideon (2023). „BloombergGPT: A Large Language Model for Finance“. In: *arXiv preprint arXiv:2303.17564*. URL: <https://arxiv.org/abs/2303.17564>.
- Xiao, Shitao, Liu, Zheng, Zhang, Peitian, Muennighoff, Niklas, Lian, Defu und Nie, Jian-Yun (2024). „C-Pack: Packed Resources For General Chinese Embeddings“. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '24.

- Washington DC, USA: Association for Computing Machinery, S. 641–649. ISBN: 9798400704314. DOI: 10.1145/3626772.3657878. URL: <https://doi.org/10.1145/3626772.3657878>.
- Yang, Hongyang, Liu, Xiao-Yang und Wang, Christina Dan (2023). „FinGPT: Open-Source Financial Large Language Models“. In: *arXiv preprint arXiv:2306.06031*. URL: <https://arxiv.org/abs/2306.06031>.
- Yepes, Antonio Jimeno, You, Yao, Milczek, Jan, Laverde, Sebastian und Li, Renyu (2024). „Financial Report Chunking for Effective Retrieval Augmented Generation“. In: *arXiv preprint arXiv:2402.05131*. DOI: 10.48550/arXiv.2402.05131. URL: <https://arxiv.org/abs/2402.05131>.
- Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q. und Artzi, Yoav (2020). „BERTScore: Evaluating Text Generation with BERT“. In: *International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=SkeHuCVFDr> (besucht am 19.09.2025).
- Zhang, Xuanyu, Yang, Qing und Xu, Dongliang (2023). „XuanYuan 2.0: A Large Chinese Financial Chat Model with Hundreds of Billions Parameters“. In: *arXiv preprint arXiv:2305.12002*. URL: <https://arxiv.org/abs/2305.12002>.
- Zhang, Yue, Li, Yafu, Cui, Leyang, Cai, Deng, Liu, Lemao, Fu, Tingchen, Huang, Xinting, Zhao, Enbo, Zhang, Yu, Chen, Yulong, Wang, Longyue, Luu, Anh Tuan, Bi, Wei, Shi, Freda und Shi, Shuming (2023). „Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models“. In: *arXiv preprint arXiv:2309.01219*. URL: <https://arxiv.org/abs/2309.01219>.
- Zhao, Huaqin, Liu, Zhengliang, Wu, Zihao, Li, Yiwei, Yang, Tianze, Shu, Peng, Xu, Shaochen, Dai, Haixing, Zhao, Lin, Jiang, Hanqi, Pan, Yi, Chen, Junhao, Zhou, Yifan, Mai, Gengchen, Liu, Ninghao und Liu, Tianming (2024). „Revolutionizing Finance with LLMs: An Overview of Applications and Insights“. In: *arXiv preprint arXiv:2401.11641*. URL: <https://arxiv.org/abs/2401.11641>.
- Zhao, Wayne Xin, Liu, Jing, Ren, Ruiyang und Wen, Ji-Rong (2024). „Dense Text Retrieval Based on Pretrained Language Models: A Survey“. In: *ACM Transactions on Information Systems* 42.4, S. 1–60. ISSN: 1046-8188. DOI: 10.1145/3637870.

## Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die eingereichte Bachelorarbeit selbständig und ohne fremde Hilfe verfasst, keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. KI-Anwendungen, die ich als Hilfsmittel verwendet habe, habe ich im Anhang meiner Arbeit dokumentiert. Daraus geht hervor, welche KI-Anwendung ich für die dort angegebenen Zwecke in den genannten Teilen der Arbeit eingesetzt habe.

---

Ort, Datum

---

Oğuzhan-Burak Bozkurt