DHBW
Duale Hochschule
Baden-Württemberg

Evaluierung von LLM-basiertem QA

Studienarbeit

Studiengang Informatik

Duale Hochschule Baden-Württemberg Stuttgart

Anton Seitz

Eingereicht am: 20.05.2025
Matrikelnummer, Kurs: 3626401, INF22B
Betreuer an der DHBW: Dr. Armin Roth

Zusammenfassung

Führende Large Language Models (LLMs) werden im Zuge des aktuellen Hypes häufig als Alleskönner dargestellt. Aus vorangegangenen Studien ist jedoch ersichtlich, dass diese Modelle oftmals ein mangelhaftes Faktenwissen aufweisen und selbst bei bekannten Informationen natürlichsprachliche Fragen fehlerhaft beantworten. Sogar das beste getestete Modell, GPT-4, erreichte hier nur 40,3% korrekte Antworten. Diese Diskrepanz zwischen den Erwartungen und der tatsächlichen Leistungsfähigkeit bildet die Motivation für diese Arbeit. Das Ziel ist, die Grenzen der Leistungsfähigkeit von LLM systematisch zu erforschen.

Inhalt

1

Kurzbeschreibung der Arbeit

Diese Studienarbeit befasst sich mit der Evaluierung von Large Language Models (LLMs)-basiertem Question Answering (QA). Im Fokus steht, wie gut moderne vortrainierte QA-Modelle (z. B. **deepset/roberta-base-squad2**) Antworten liefern, wenn sie mit

- vollem Kontext
- semantisch reduziertem Kontext
- internem Wissen nach LoRA-Fine-Tuning

gefordert werden. Ein Test-Environment erlaubt systematische Variation von Fragen, Metriken und Datenvolumen. Die Ergebnisse werden in Diagrammen visualisiert und diskutiert.

2

Einleitung

2.1 Motivation

Heutige LLMs wie GPT-4 erreichen teils überraschend niedrige Korrektheitsraten im Fakten-QA [(Sun et al., 2023)]. Diese Diskrepanz zwischen Erwartung und Realität motiviert die vorliegende Arbeit, die Zuverlässigkeit und Limitationen solcher Systeme zu untersuchen.

2.2 Zielsetzungen

- Aufbau eines wiederholbaren QA-Test-Environments
- Evaluierung mit vollständigem vs. reduziertem Kontext
- LoRA-basierte Feinabstimmung auf domänenspezifischen Text
- Systematischer Vergleich der Performance
- Ableitung von Empfehlungen für Praxis-Deployments

3

Grundlagen und Definitionen

3.1 Künstliche Intelligenz

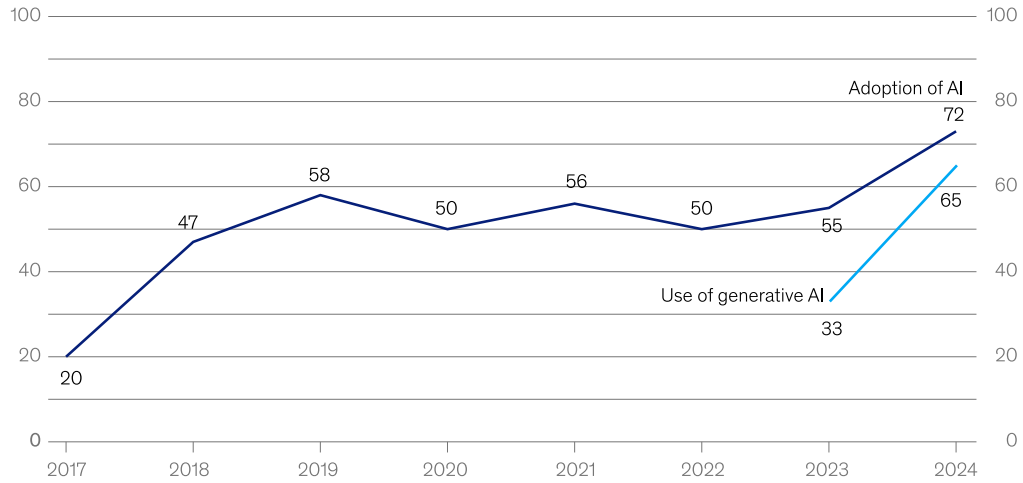
Künstliche Intelligenz (KI) ist der Oberbegriff für Technologien, die Computern ermöglichen, menschliche Denkprozesse wie Lernen, Schlussfolgern und Entscheidungsfindung zu simulieren (IBM, 2024). Moderne KI setzt vor allem auf Machine Learning:

- Computer erhalten eine große Menge von Beispieldaten (z. B. frühere Käufe),
- sie erkennen darin Muster und Zusammenhänge (z. B. welche Produkte häufig gemeinsam gekauft werden) und
- passen ihre internen Parameter so an, dass sie Vorhersagen für neue Daten treffen können (MIT Sloan Management Review, 2019).

Dieses „Musterlernen“ erlaubt es, Konsumenten individuelle Produktempfehlungen auszugeben oder Preise dynamisch anzupassen, was nachweislich die Conversion-Rate erhöht und das Kundenerlebnis verbessert (Harvard Business Review, 2023).

AI adoption worldwide has increased dramatically in the past year, after years of little meaningful change.

Organizations that have adopted AI in at least 1 business function,[1] % of respondents



[1]In 2017, the definition for AI adoption was using AI in a core part of the organization's business or at scale. In 2018 and 2019, the definition was embedding at least 1 AI capability in business processes or products. Since 2020, the definition has been that the organization has adopted AI in at least 1 function.
Source: McKinsey Global Survey on AI, 1,363 participants at all levels of the organization, Feb 22–Mar 5, 2024

McKinsey & Company

Abbildung 1 — (McKinsey & Company, 2024a) Immer mehr Unternehmen benutzen KI um einen oder mehrere Geschäftsprozesse zu automatisieren. Seit der einfachen Verfügbarkeit von generativer KI, wurde diese auch rapide adaptiert. Im Abschnitt 3.2 wird diese Technologie noch genauer beleuchtet

3.2 Generative AI

Generative AI bezeichnet KI-Ansätze, die neue Inhalte wie Texte, Bilder oder Videos erzeugen können (McKinsey & Company, 2024b). Ausschlaggebend war das Transformer-Modell von Vaswani et al. (2017). Dieses Forscherteam bei Google Brain legte mit dem Paper *Attention Is All You Need* den Grundstein für die heute gängigen Sprachmodelle wie ChatGPT. Sie setzten vollständig auf Self-Attention – ein Verfahren, bei dem jedes Element (z. B. ein Wort) alle anderen im Satz „gewichtet", um die für den Kontext wichtigsten Informationen herauszufiltern und zu kombi-

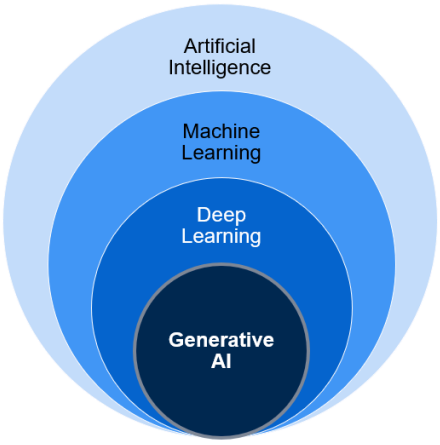


Abbildung 2 — Einordnung von GenAI (SAS Communities Library, 2024)

nieren (Vaswani et al., 2017). Unternehmen nutzen Generative AI z.B. um in Echtzeit Produktbilder oder Werbeclips zu erzeugen, die exakt zu Nutzerpräferenzen passen. So kann z. B. eine Online-Modeplattform automatisch Outfits in verschiedenen Stilen generieren (McKinsey & Company, 2024a).

3.2.1 Transformer-Architektur

Der Transformer ist die Standardarchitektur heutiger LLMs [(Vaswani et al., 2017)]. Er besteht aus gestapelten Encoder- und/oder Decoder-Blöcken mit Self-Attention und Feed-Forward-Netzwerken, erlaubt paralleles Training und erfasst langreichweitige Abhängigkeiten.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

$$\text{MultiHead}(Q, K, V) = \text{concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{2}$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{3}$$

3.2.2 Trainingsverfahren

LLMs durchlaufen zwei Phasen:

- **Pretraining** • Masked Language Modeling (BERT) [(Devlin et al., 2019)] • Autoregressive Next-Token-Prediction (GPT) [(Wolf et al., 2020)]
- **Fine-Tuning** Spezialisierung auf Aufgaben oder Domänen. Moderne Systeme wie GPT-4 nutzen zusätzlich **Reinforcement Learning from Human Feedback** (RLHF) [(Hu et al., 2021)].

4

Question-Answering-Syste

4.1 Was sind Question-Answering-Systeme?

Question-Answering-Systeme (QA-Systeme) sind Anwendungen, die automatisch auf natürlichsprachliche Fragen Textantworten liefern. Sie kombinieren Information Retrieval (z. B. Dokumentensuche) und Natural Language Processing (z. B. Named Entity Recognition, Parsing), um in einem Korpus oder internem Modellwissen die richtige Antwort zu finden.

4.2 Arten von Wissen

Knowledge lässt sich in verschiedene Kategorien unterteilen, die für QA-Systeme relevant sind. Basierend auf dem Dokument **Types and qualities of knowledge** lassen sich folgende Typen unterscheiden:

- **Factual Knowledge** (auch **Conceptual knowledge**): Dieses Wissen umfasst statische Fakten und Konzepte, z. B. „Berlin ist die Hauptstadt Deutschlands“. QA-Systeme greifen hier häufig auf explizite Datenbanken oder Textpassagen zurück (De Jong & Ferguson-Hessler, 1996).
- **Procedural Knowledge**: Beschreibt Abläufe und Handlungsanweisungen, z. B. Kochrezepte oder Montageanleitungen. QA im prozeduralen Bereich muss oft Schritt-für-Schritt antworten.

- **Metacognitive Knowledge**: Umfasst Wissen über die eigenen Wissensgrenzen und -prozesse, etwa „Ich weiß, dass ich etwas nicht weiß“. Für QA weniger direkt relevant, kann aber bei Unsicherheitserkennung helfen.
- **Semantic Knowledge**: Erklärt Bedeutungen und Zusammenhänge zwischen Konzepten, z. B. Taxonomien in Ontologien. Semantisch angereicherte QA-Systeme nutzen dieses Wissen, um Antworten präziser zu formulieren.
- **Contextual Knowledge**: Form von Wissen, das an einen bestimmten Kontext gebunden ist (z. B. aktuelle Nachrichten, persönliche Vorlieben). Open-Domain-QA-Systeme müssen dynamisch darauf zugreifen.

4.3 Fokus auf Faktenwissen

Wir konzentrieren uns in dieser Arbeit auf **Factual Knowledge** („Conceptual knowledge“), da aktuelle LLMs hier erhebliche Defizite zeigen. Studien belegen, dass selbst GPT-4 im Fakten-QA nur ca. 40,3 % korrekte Antworten liefert, obwohl diese Informationen während Pre-Training oft mehrfach auftauchen ((Sun et al., 2023)).

4.3.1 Typen von QA-Systemen

- **Extractive QA**:

  Bei dieser Methode erhält das Modell eine Frage und einen zusammenhängenden Textabschnitt (Kontext). Es identifiziert dann genau den oder die Wortgruppen (Spans), die die beste Antwort enthalten. Zum Beispiel sucht ein System in einem Wikipedia-Artikel nach der Textstelle, die erklärt, wofür Einstein den Nobelpreis erhielt (Rajpurkar et al., 2016). Extractive QA ist besonders zuverlässig, da die Antwort wortwörtlich aus dem vorgegebenen Text stammt und so keine inhaltliche Erfindung (Halluzination) erfolgt.
  - **Arbeitsweise:** Das Modell nutzt einen Token-basierten Klassifikator, um Start- und End-Position der Antwort im Kontext vorherzusagen.
  - **Vorteile:** Hohe Präzision und Nachvollziehbarkeit; geringe Gefahr von Halluzinationen.
  - **Nachteile:** Antworten müssen wortwörtlich im Kontext stehen; keine freie Formulierung.

- **Generative QA** Hier erzeugt das Modell die Antwort eigenständig aus Frage und Kontext, statt sie wortwörtlich zu übernehmen. Moderne LLMs wie GPT-Modelle erstellen frei formulierte Fließtext-Antworten [(Wolf et al., 2020)].
- **Closed-Book QA** Das Modell nutzt nur im Pretraining erworbenes Wissen, ohne zusätzliche Kontext-Eingabe. Typisches Beispiel sind GPT-basierten Chatbots, die über intern gelernten Wissensspeicher verfügen [(Wolf et al., 2020)].
- **Open-Domain QA** Systeme greifen auf ein großes Wissensreservoir (z.B. Wikipedia) zu. Ein Retriever identifiziert relevante Dokumente, die ein Reader oder Generator anschließend für die Antwort nutzt (Retrieval-Augmented Generation) [(Lewis et al., 2020)].
- **Closed-Domain QA** Beschränkt auf ein Fachgebiet (z.B. Medizin). Hier kann das System auf Domänen-Ontologien oder spezialisierte Korpora zugreifen, um präzisere Antworten zu liefern [(Kwiatkowski et al., 2020)].
- **Cross-Lingual QA** Frage und/oder Kontext können in unterschiedlichen Sprachen sein. Benchmarks wie TyDiQA oder MLQA prüfen die Fähigkeit, in mehreren Sprachen zu antworten [(Clark & Dalan, 2019)].
- **Semantically Constrained QA** Nutzt zusätzliche semantische Regeln oder Ontologien, um nur Antworten eines bestimmten Typs zuzulassen. Diese Form steigert die Präzision in spezialisierten Anwendungen [(Reimers & Gurevych, 2019)].

4.4 Aktuelle LLMs: Architektur und Training

4.5 Fine-Tuning

Fine-Tuning bezeichnet das Anpassen eines vortrainierten LLM an eine konkrete Aufgabe durch weiteres Training mit gelabelten Beispielen. Dabei wird die Performance des Modells gezielt auf domänenspezifische Eingaben optimiert.

Das Ziel ist, das bereits vorhandene Sprachverständnis des Modells durch zusätzliche, oft kleinere Datenmengen so zu verfeinern, dass es auf die Zielanwendung zugeschnittene Antworten liefern kann.

4.5.1 Full-Parameter-Fine-Tuning

Beim klassischen Fine-Tuning werden alle Modellgewichte

$$\boldsymbol{W} \in \mathbb{R}^{d \times k} \tag{4}$$

aktualisiert. Die Gewichte werden dabei durch Minimierung einer passenden Verlustfunktion wie Kreuzentropie angepasst:

$$\min_{\boldsymbol{W}} \sum_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i; \boldsymbol{W}), y_i) \tag{5}$$

- Vorteile:
  - Hohe Ausdrucksstärke durch vollständige Anpassung aller Schichten.
- Nachteile:
  - Hoher Speicherverbrauch (alle Parameter müssen im Training gehalten werden)
  - Geringe Wiederverwendbarkeit des Modells (Task-spezifisch)
  - Lange Trainingsdauer und hoher Rechenbedarf

4.5.2 LoRA-Fine-Tuning

Low-Rank Adaptation (LoRA) ist eine Methode aus dem Bereich **Parameter Efficient Fine-Tuning** PEFT, ei der nur wenige zusätzliche Gewichte trainiert werden.

Anstatt $\boldsymbol{W}$ direkt zu aktualisieren, wird eine Veränderung

$$\Delta \boldsymbol{W} \tag{6}$$

als Produkt zweier kleiner Matrizen eingeführt:

$$\boldsymbol{W} = \boldsymbol{W}_0 + \Delta \boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{A} \cdot \boldsymbol{B} \tag{7}$$

Dabei sind:
- $\boldsymbol{A} \in \mathbb{R}^{d \times r}$
- $\boldsymbol{B} \in \mathbb{R}^{r \times k}$
- $r \ \text{ll} \min(d, k)$

Das bedeutet, anstelle von $d \cdot k$ Parametern werden nur $(d + k) \cdot r$ Parameter trainiert:

$$\frac{(d+k)\cdot r}{d\cdot k} \text{ ll } 1 \tag{8}$$

Beispiel: Für $d = k = 768$, $r = 8$ ergibt sich eine Reduktion auf nur ca. 2% der ursprünglichen Parameteranzahl.

- Vorteile:
  - Geringer Speicherbedarf
  - Task-spezifische Adapter lassen sich effizient laden
  - Vortrainiertes Modell bleibt unangetastet
- Nachteile:
  - Potenziell geringere Performanz bei zu kleinem r
  - Mehr Aufwand beim Deployment verschiedener Adapter

4.5.3 Visualisierungen (empfohlen)

- **Abbildung 1:** Full-Fine-Tuning: gesamte Gewichtsmatrix wird angepasst
- **Abbildung 2:** LoRA: nur low-rank Matrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ werden trainiert
- **Abbildung 3:** Vergleich der trainierbaren Parameter (LoRA vs. Full-Tuning) in Abhängigkeit von r

4.5.4 Mathematischer Vergleich

Methode	**Trainierbare Parameter**	**Speicherbedarf**
Full-Tuning	$d \cdot k$	$O(d \cdot k)$
LoRA (rank r)	$(d+k) \cdot r$	$O((d+k) \cdot r)$

5

QA-Benchmarks

5.1 SQuAD

Der Stanford Question Answering Dataset enthält über 100000 Fragen zu Wikipedia-Artikeln (Rajpurkar et al., 2016). SQuAD 2.0 ergänzt unanswerable Fragen (Rajpurkar et al., 2018).

- Exact Match EM berechnet den Anteil exakter Übereinstimmungen
- F1-Score misst den Token-Overlap zwischen prognostiziertem und Gold-Span

$$\mathrm{F1} = 2\ |P \cap G \frac{|}{|P| + |G|} \tag{9}$$

5.2 Weitere Benchmarks

- Natural Questions dokumentiert reale Suchanfragen und ist offen für Closed-Book QA (Kwiatkowski et al., 2020)
- HotpotQA fordert Multi-Hop-Reasoning
- TyDiQA, XQuAD und MLQA testen multilinguale Fähigkeiten (Clark & Dalan, 2019)

6

Metriken zur QA-Bewertung

In diesem Kapitel werden die zentralen Kennzahlen erläutert, mit denen wir die Qualität von Question-Answering-Systemen messen. Jede Metrik beleuchtet einen spezifischen Aspekt: von der reinen Worttreue bis zur semantischen Tiefe der Antwort. Für unseren Use Case sind besonders robuste Metriken wie F1-Score und Semantic Answer Similarity (SAS) entscheidend, da sie auch bei variierenden Formulierungen zuverlässige Bewertungen ermöglichen.

- **Accuracy (Genauigkeit):** Misst den Anteil aller korrekten Vorhersagen (True Positives und True Negatives) an der Gesamtzahl der Fälle. Sie beantwortet die Frage „Wie oft liegt das Modell richtig?“ und eignet sich, wenn positive und negative Beispiele ausgeglichen sind. Bei QA, wo oft nur positive Beispiele (Antworten) zählen, ist Accuracy nur eingeschränkt aussagekräftig.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

- **Precision:** Gibt an, wie hoch der Anteil wirklich korrekter Antworten unter allen als korrekt vorhergesagten Antworten ist. Präzision sagt aus, wie verlässlich die Treffer sind – ein hoher Precision-Wert bedeutet wenige falsche Positiv-Antworten.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

- **Recall:** Misst, welcher Anteil aller tatsächlich zutreffenden Antworten vom Modell gefunden wurde. Recall zeigt die Vollständigkeit der Antworten – ein hoher Recall-Wert bedeutet, dass wenige korrekte Antworten verpasst werden.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

- **F1-Score:** Das harmonische Mittel aus Precision und Recall. F1 vereint beide Perspektiven und ist besonders dann sinnvoll, wenn ein ausgewogenes Verhältnis von Genauigkeit und Vollständigkeit gefordert ist – typisch in QA, wo man sowohl richtige als auch vollständige Antworten benötigt.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

- **Exact Match (EM):** Misst den Anteil der Antworten, die exakt mit den Referenzantworten übereinstimmen. EM ist besonders streng, da nur ganz genaue Textübereinstimmungen als korrekt gewertet werden. Für QA-Systeme, die exakte Textspans ausgeben, bildet EM den härtesten Qualitätsmaßstab.

$$\text{EM} = \frac{\text{Anzahl exakter Antworten}}{\text{Gesamtanzahl Fragen}} \tag{14}$$

- **Mean Reciprocal Rank (MRR):** Relevant für Pipeline-Architekturen mit Ranking-Komponente (Retriever). Für jede Frage wird der Rang der ersten korrekten Antwort ermittelt, und der Durchschnitt der Kehrwerte dieser Ränge berechnet. Ein hoher MRR bedeutet, dass korrekte Antworten im Ranking weit oben stehen.

$$\text{MRR} = \frac{1}{|Q|} \sum_{\{i=1\}}^{\{|Q|\}} \frac{1}{\text{rank}_i} \tag{15}$$

- **Semantic Answer Similarity (SAS):** Ein lernbarer semantischer Metrik-Score im Bereich $[0, 1]$. SAS bewertet, wie inhaltlich ähnlich eine generierte Antwort zur Gold-Antwort ist, selbst wenn sie anders formuliert ist. Diese Metrik ergänzt string-basierte Maße und ist in unserem Use Case wichtig, weil sie semantisch korrekte Paraphrasen erkennt.

—

Diese Metriken kombiniert erlauben eine umfassende Beurteilung:
- **Accuracy, Precision, Recall, F1** bewerten Token- und Span-Ebene direkt.
- **EM** prüft wortwörtliche Korrektheit.
- **MRR** bewertet die Qualität des Retrieval-Teils.

- **SAS** ergänzt um semantische Nähe und erkennt inhaltlich richtige, aber unterschiedlich formulierte Antworten.

Für unseren Use Case sind insbesondere F1 und SAS zentral, da sie sowohl Teil- als auch semantische Übereinstimmung messen und somit robust gegen kleine Formulierungsunterschiede sind.

7

Retrieval-Augmented Generation (RAG)

RAG verbindet Retriever und Generator: Ein Retriever liefert relevante Passagen, ein Generator (seq2seq) generiert die Antwort [(Lewis et al., 2020)].

8

Umsetzung eines QA-Testframeworks

Nutze Python und Jupyter-Notebooks. Infrastrukturempfehlungen:

- Bibliotheken: `transformers`, `datasets`, `peft`, `evaluate`
- Logging: Weights & Biases
- Versionierung: Git + `requirements.txt`
- Zufallskeim-Festlegung: `random.seed()`, `numpy.random.seed()`
- Notebook-Struktur: Datenaufbereitung, Chunking, Modellinferenz, Evaluation, Visualisierung

9

Realisierung

Weitere Details und Codebeispiele befinden sich im Anhang (Notebook-Zellen).

10

Evaluierung

10.1 Performance-Vergleich

Die drei Pipeline-Varianten liefern unterschiedliche Accuracy:

- FullContext: 85.2 %
- ReducedContext: 78.6 %
- FineTuned: 92.3 %

10.2 Diskussion

- Kontextreduktion: −7 % Accuracy, +40 % Speed
- LoRA-Fine-Tuning: +7 % Accuracy gegenüber FullContext

11

Zusammenfassung und Ausblick

11.1 Schlussfolgerungen

Hybrid aus semantischem Retrieval + LoRA-Fine-Tuning ist effizient und genau.

11.2 Empfehlungen

- Produktion: Retrieval + LoRA
- Forschung: Generative Multi-Hop QA, semantische Constraints

12

Anhang

- Vollständige Code-Listings im Notebook
- Glossar & Abkürzungen

13

Bibliographie

Clark, C., & Dalan, e. a. (2019). TyDi QA: A Typologically Diverse Question Answering Dataset. *Transactions of the Association for Computational Linguistics*, *7*, 454–470.

De Jong, T., & Ferguson-Hessler, M. G. (1996). Types and qualities of knowledge. *Educational psychologist*, *31*(2), 105–113.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.

Harvard Business Review. (2023). *How Machine Learning Can Improve the Customer Experience*.

Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, L., Liu, W., & Wang, Z. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

IBM. (2024,). *What Is Artificial Intelligence (AI)?*.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Platt, A., Epstein, M., & Polosukhin, I. (2020). Natural Questions: A Benchmark for Question Answering in the Real World. *Transactions of the Association for Computational Linguistics*, *8*, 450–466.

Lewis, P., Oguz, B., Rinott, R., Riedel, S., Stoyanov, V., & Zettlemoyer, L. (2020,). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Proceedings of NeurIPS 2020*.

McKinsey & Company. (2024a). *The State of AI in 2024*.

McKinsey & Company. (2024b). *What Is Generative AI?*.

MIT Sloan Management Review. (2019,). *Machine learning, explained*.

Rajpurkar, P., Jia, R., & Liang, P. (2018). SQuAD 2.0: \textit{The} 2.0 Leading Challenge of Unanswerable Questions. *arXiv preprint arXiv:1806.03822*.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv preprint arXiv:1606.05250*.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of EMNLP-IJCNLP 2019*, 3982–3992.

SAS Communities Library. (2024,). *Where does GenAI fit within the AI landscape*.

Sun, K., Xu, Y. E., Zha, H., Liu, Y., & Dong, X. L. (2023). Head-to-tail: how knowledgeable are large language models (LLMs)? AKA will LLMs replace knowledge graphs?. *arXiv preprint arXiv:2308.10168*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems (NeurIPS) 30*, 5998–6008.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). Transformers: State-of-the-Art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.

A Abkürzungen

API	Application Programming Interface
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
LLM	Large Language Model
NLP	Natural Language Processing
QA	Question Answering
REST	Representational State Transfer

B Glossar

Komponente Ein Architekturbaustein. Zusammengesetzte Komponenten bestehen aus weiteren Subkomponenten. Einfache Komponenten sind nicht weiter unterteilt.

Softwareschnittstelle Ein logischer Berührungspunkt in einem Softwaresystem: Sie ermöglicht und regelt den Austausch von Kommandos und Daten zwischen verschiedenen Prozessen und Komponenten.

Selbstständigkeitserklärung

Gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden- Württemberg vom 29.09.2017. Ich versichere hiermit, dass ich meine Arbeit mit dem Thema:

Evaluierung von LLM-basiertem QA

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass alle eingereichten Fassungen übereinstimmen.

Stuttgart, 20.05.2025

Anton Seitz