DHBW
Duale Hochschule
Baden-Württemberg

Evaluierung von LLM-basiertem QA

Studienarbeit

Studiengang Informatik

Duale Hochschule Baden-Württemberg Stuttgart

Anton Seitz

Eingereicht am: 03.06.2025
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

In dieser Studienarbeit wird die Leistungsfähigkeit moderner Large Language Models (LLMs) im Bereich des Question Answering (QA) systematisch untersucht. Ausgangspunkt ist die Erkenntnis, dass selbst hochentwickelte Modelle wie GPT-4 nur rund 40 % der Fragen korrekt beantworten, obwohl ihnen häufig universelle Problemlöserfähigkeiten zugeschrieben werden (Sun et al., 2023). Zunächst wird ein thematisch geeigneter Textkorpus ausgewählt und für die spätere Evaluierung aufbereitet. Darauf aufbauend werden Testfragen formuliert und Referenzantworten erstellt, um eine belastbare Vergleichsbasis zu schaffen.

Anschließend wird ein ausgewähltes LLM in einer speziell eingerichteten Testumgebung eingesetzt. Hierbei werden sowohl quantitative Metriken wie Genauigkeit und Vollständigkeit als auch qualitative Kriterien zur Bewertung herangezogen. Die Experimentierphase umfasst Tests unter variierenden Modellparametern und Anpassungen der Pipeline, um deren Einfluss auf die Antwortqualität zu erfassen.

Im letzten Schritt erfolgt die systematische Auswertung der gewonnenen Daten. Dabei werden Limitationen der Modelle aufgezeigt und mögliche Optimierungsansätze diskutiert. Die Dokumentation fasst sämtliche Ergebnisse zusammen und liefert Handlungsempfehlungen für den praktischen Einsatz von LLM-basierten QA-Systemen, insbesondere in ressourcenbeschränkten Umgebungen.

2

Einleitung

2.1 Motivation

Heutige Large Language Models (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-basiertes Fine-Tuning auf domänenspezifischen Text
- Systematischer Vergleich der Performance
- Ableitung von Empfehlungen für Praxis-Deployments

Die Arbeit gliedert sich in drei Phasen: eine Vorbereitungsphase mit Literaturrecherche, Korpuserstellung und Methodendefinition, eine Experimentierphase mit Implementierung und Testdurchführung sowie eine abschließende Auswertungs- und Dokumentationsphase. Auf diese Weise sollen fundierte Erkenntnisse über die tatsächliche Leistungsfähigkeit von LLMs im Question Answering gewonnen werden.

3

Grundlagen und Definitionen

Zunächst werden die nötigen Grundlagen für das Verständnis der Arbeit geschaffen.

3.1 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 (*Question Answering with BERT*, 2023).

3.1.1 Arten von Wissen

Knowledge lässt sich in verschiedene Kategorien unterteilen, die für QA-Systeme relevant sind. Basierend auf **Types and qualities of knowledge** (De Jong & Ferguson-Hessler, 1996) 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.

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).

3.1.2 Typen von QA-Systemen

Im Folgenden werden die üblichen Typen des QA beschrieben und erläutert, welcher davon sich am besten für den bestehenden Anwendungsfall eignet.

- **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.

(*Question Answering with BERT*, 2023)

- **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).

Für unseren Anwendungsfall haben wir uns für Extractive QA entschieden, da hier die Antworten direkt als Textspans aus einem vorgegebenen Dokument extrahiert werden und somit hohe Präzision und Nachvollziehbarkeit gewährleisten. Anders als bei generativen Modellen, die freie Fließtext-Antworten erzeugen und dabei zu Halluzinationen neigen können (Wolf et al., 2020), sucht das Extractive-System gezielt nach der Start- und Endposition der korrekten Antwort im Kontexttext, wie es beispielsweise im SQuAD-Datensatz üblich ist (Rajpurkar et al., 2016). So lassen sich falsche Vorhersagen einfach analysieren und korrigieren, weil der Modell-Output immer klar auf eine Textstelle zurückzuführen ist. Zudem bedarf es kaum Prompt-Engineering, sondern lediglich einer geeigneten Hugging-Face-Pipeline, die in Jupyter-Notebooks effizient auf verschiedene Dokumente skaliert. Diese Kombination aus Verlässlichkeit, schneller Integrationsfähigkeit und geringem Anpassungsaufwand macht Extractive QA für unsere Evaluierung ideal.

3.2 Aktuelle LLMs: Architektur und Training

Im Folgenden werden nötige Grundlagen zu den Themen Künstliche Intelligenz (KI) und insbesondere LLM und deren Fine-Tuning geschaffen.

3.3 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).

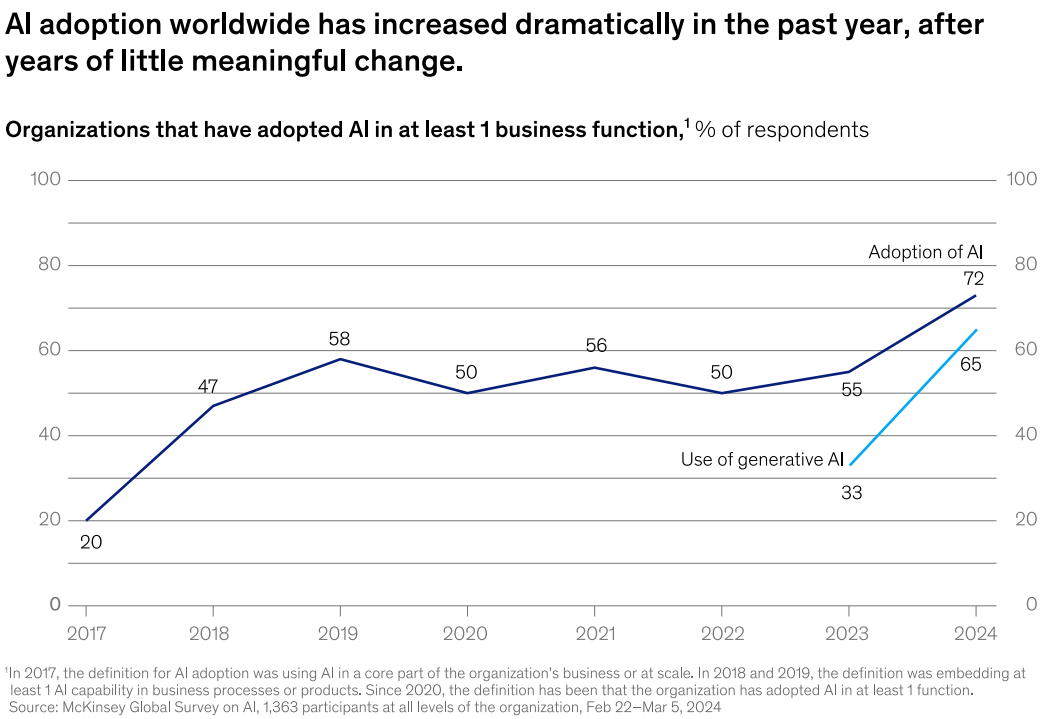

Abbildung 1 — (McKinsey & Company, 2024a) Immer mehr Unternehmen benutzen KI um einen oder mehrere Geschäftsprozesse zu automatisieren. Seit der einfachen Verfügbarkeit von generativer KI, wurde diese auch rapide adaptiert. Im Abschnitt 3.4 wird diese Technologie noch genauer beleuchtet

3.4 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-

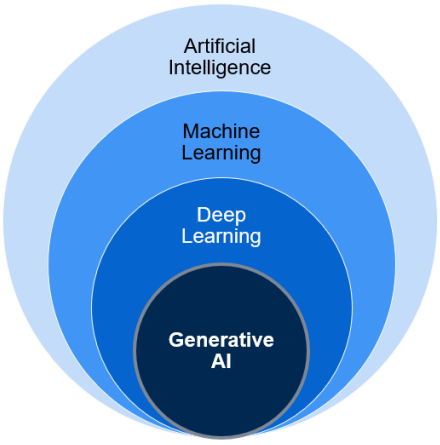

Abbildung 2 — Einordnung von GenAI (SAS Communities Library, 2024)

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).

3.4.1 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.

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

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

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

3.4.2 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).

3.5 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.

3.5.1 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

3.5.2 LoRA-Fine-Tuning

Low-Rank Adaptation (LoRA) ist eine Methode aus dem Bereich **Parameter Efficient Fine-Tuning** PEFT, bei 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:
- $\boldsymbol{A} \in \mathbb{R}^{d \times r}$
- $\boldsymbol{B} \in \mathbb{R}^{r \times k}$
- $r \ \text{ll} \min(d, k)$

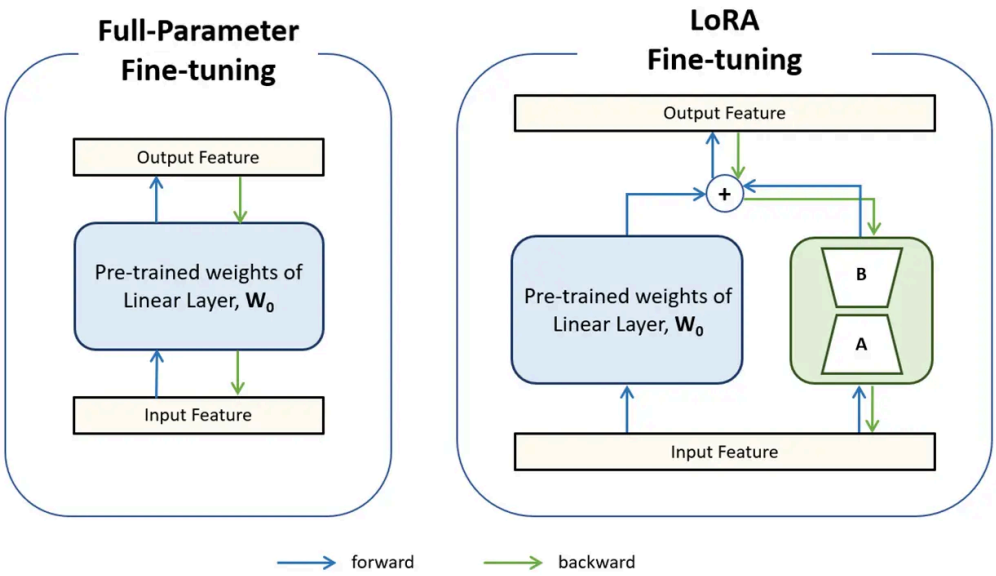

Abbildung 3 — Full-Parameter-Fine-Tuning vs Low-Rank Adaption (LoRA) (Intel Corporation, 2024)

(Intel Corporation, 2024) zeigt, dass beim Full-Parameter-Tuning alle Gewichte (inklusive Bias) eines vortrainierten Layers direkt angepasst werden, während LoRA die ursprünglichen Parameter einfriert und ausschließlich zwei low-rank, bzw. Matrizen A und B trainiert, deren skaliertes Produkt als Residual zum ursprünglichen Layer-Output addiert wird. Dadurch reduziert LoRA den Speicher- und Rechenaufwand beim Fine-Tuning erheblich, da nur ein Bruchteil der Parameter trainiert werden.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

3.5.3 Mathematischer Vergleich

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

Da r typischerweise deutlich kleiner ist als d und k, fällt der Parameter- und Speicheraufwand bei LoRA im Vergleich zum Full-Tuning erheblich geringer aus. Dadurch eignet sich LoRA besonders für ressourcenoptimiere Umgebungen oder große Modelle.

3.6 SQuAD

Ein beliebter Datensatz für QA-Systeme ist SQuAD. Dort wurden in einem strukturierten Format über 100000 Fragen zu Wikipedia-Artikeln aufbereitet (Rajpurkar et al., 2016). SQuAD 2.0 ergänzt unanswerable Fragen (Rajpurkar et al., 2018).

3.7 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)

3.8 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 (Powers, 2020), (Risch et al., 2021).

- **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.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

- **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.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

- **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.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

- **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.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

- **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.

$$\text{EM} = \frac{\text{Anzahl exakter Antworten}}{\text{Gesamtanzahl Fragen}} \tag{13}$$

- **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.

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{\{i=1\}}^{\{|Q|\}} \frac{1}{\mathrm{rank}_i} \tag{14}$$

- **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 (Evidently AI, 2023).

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.

4

Realisierung

4.1 Textkorpus

Zunächst wurde ein umfangreicher Textkorpus zusammengestellt. Als Thema wurde **Judo** gewählt, da sich der Entwickler gut damit auskennt und Judo sich für Faktenwissen-Tests eignet. Es gibt zahlreiche Details – von Technikklassifizierungen über historische Daten bis hin zu Wettkampfergebnissen, die sich gut abfragen lassen. Dabei wurden für den Textkorpus folgende Quellen gewählt:

Quelle	**Begründung**
https://en.wikipedia.org/wiki/Judo	Wikipedia liefert eine umfassende Übersicht über Geschichte, Regeln und Begriffe.
https://en.wikipedia.org/wiki/List_of_judo_techniques	Detaillierte Auflistung aller Techniken, ideal für technische Beispiele.
https://en.wikipedia.org/wiki/List_of_judoka	Informationen zu bedeutenden Judoka, nützlich für biografische Fragen.
https://martialarts.fandom.com/wiki/Judo	Populärkulturelle Perspektive und weiterführende Details.
https://chas-ma.com/JudoManual/Chapter_2%28HistoryofJudo%29.pdf	Fachlicher PDF-Quelltext zur historischen Entwicklung von Judo.
https://www.ijf.org/history	Offizielle Historie der International Judo Federation (IJF).

https://blackbelttrek.com/judo-vs-jiu-jitsu-the-ultimate-comparison/	Vergleich mit Jiu-Jitsu, um Abgrenzungen und historische Zusammenhänge zu verdeutlichen.

4.2 Fragesätze

Die Fragesätze liegen in einem flachen JSON-Format vor, bei dem jedes Objekt genau die Felder `question` und `answer` enthält. Im Laufe der Arbeit wurden drei Prototypen entwickelt und evaluiert:

Prototyp 1 – verbos und unstrukturiert Im ersten Ansatz enthielten die Antworten oft vollständige Sätze oder sogar ganze Absätze. Diese verbosen Rückgaben führten zu schlechter Performance, da das QA-Modell auf kurze, prägnante Antworten optimiert ist. Zudem waren manche Fragen nicht rein auf Faktenwissen ausgelegt, sondern erforderten etwas längere Erklärungen, was die Auswertung zusätzlich erschwerte.

Prototyp 2 – atomare Antworten Um die Performance deutlich zu steigern, wurden die Fragen so angepasst, dass jede Antwort **atomar** ist – also nur das absolut Notwendige enthält, *ground truth*. Beispiel: Anstelle „Judo bedeutet ‚der sanfte Weg' und wurde 1882 von Kanō Jigorō gegründet" steht nur noch „der sanfte Weg". Durch diese Reduktion auf einfache Stichwortantworten verringerte sich der Fehleranteil spürbar.

Prototyp 3 – Erweiterung und Strukturierung Im dritten Schritt wurde nicht nur die Atomizität beibehalten, sondern auch das Volumen der Fragen erhöht und eine zusätzliche Kategorisierung eingeführt. Alle Fragesätze wurden anhand definierter Heuristiken in die drei Schwierigkeitsstufen **Easy**, **Medium** und **Hard** sortiert. Die Kriterien hierfür – wie Termfrequenz, Antwortlänge oder semantische Komplexität – werden im Abschnitt „Klassifikation nach Schwierigkeit" ausführlich erläutert. Diese strukturierte Vorgehensweise erlaubt eine gezieltere Analyse des Modellverhaltens je nach Fragenprofil.

4.3 Prototypen und Experimente

Dieser Abschnitt dokumentiert die iterative Entwicklung und Evaluierung verschiedener Ansätze zur Beantwortung von Fragen im Kontext eines umfangreichen Judo-Korpus. Ziel war es, die Antwortgenauigkeit zu maximieren und gleichzeitig die Effizienz des Systems zu verbessern.

4.3.1 Baseline: Vollständiger Korpus

Vorgehen: Für jede Frage wurde der gesamte Textkorpus (bestehend aus mehreren Quellen) als Kontext an das Frage-Antwort-Modell übergeben. Dieser Kontext wurde in einer Text-Datei abgelegt und beinhaltet die o.g. Webseitinhalte, die einfach aneinander konkateniert wurden. Dabei erreicht er eine Länge von ca. 140 000 Zeichen.

Beobachtungen:
- **Laufzeit**: Sehr hohe Antwortzeiten aufgrund des umfangreichen Kontextes: Jede Frage benötigt etwa 2 Minuten Rechenzeit.
- **Genauigkeit**: Solide, jedoch nicht optimal, da irrelevante Informationen den Kontext evtl. verwässern.
- **Token-Limit**: Gefahr des Überschreitens des maximalen Token-Limits des Modells, was zu abgeschnittenen Kontexten führen kann. Das verwendete Modell deepset-roberta-squad2

4.3.2 Kontextreduktion mittels semantischer Chunking

Vorgehen:
- Der Korpus wurde in 378 Chunks unterteilt, basierend auf Absätzen oder thematischen Einheiten.
- Für jede Frage wurde die semantische Ähnlichkeit zu jedem Chunk mittels Sentence-BERT berechnet.
- Die Top 50 Chunks mit der höchsten Ähnlichkeit wurden ausgewählt und als reduzierter Kontext verwendet.

Beobachtungen:
- **Laufzeit**: Signifikante Reduktion der Antwortzeiten durch kleineren Kontext.
- **Genauigkeit**: Leichter Rückgang der Genauigkeit, da relevante Informationen eventuell in nicht ausgewählten Chunks lagen.
- **Effizienz**: Deutliche Verbesserung der Systemeffizienz bei minimalem Genauigkeitsverlust.

4.3.3 Fine-Tuning mit Low-Rank Adaption (LoRA)

Vorgehen:
- Das vortrainierte Modell wurde mittels LoRA auf den spezifischen Judo-Korpus feinjustiert.

- LoRA ermöglichte effizientes Fine-Tuning durch Anpassung einer kleinen Anzahl von Parametern, wodurch der Speicherbedarf reduziert wurde.

Beobachtungen:
- **Genauigkeit**: Verbesserte Antwortgenauigkeit, insbesondere bei komplexen oder spezifischen Fragen.
- **Ressourcenverbrauch**: Geringer zusätzlicher Speicherbedarf durch den Einsatz von LoRA.
- **Anpassungsfähigkeit**: Das Modell zeigte eine bessere Anpassung an den spezifischen Sprachgebrauch und die Terminologie des Judo-Korpus.

4.3.4 Evaluation der Modelle

In der ersten Evaluierungsphase kam eine **rein stringbasierte** Methodik zum Einsatz, bei der Antworten als korrekt galten, wenn sie exakt mit den Musterantworten übereinstimmten oder eine hohe Zeichenübereinstimmung (z. B. ≥ 80 %) aufwiesen. Dieses Verfahren zeigte allerdings deutliche Schwächen:

- **Synonyme und Namensvarianten** werden nicht erkannt, z. B. „Jigoro Kano“ vs. „Kanō Jigorō“ oder „International Judo Federation“ vs. „IJF“.
- **Unterschiedliche Formulierungen und Satzstellungen** gelten als falsch, z. B. „sanfter Weg“ vs. „der sanfte Weg“ oder „1882 gründete Kanō Jigorō den Kōdōkan“ vs. „Der Kōdōkan wurde 1882 von Kanō Jigorō gegründet“.
- **Mehrdeutigkeit bei offenen Fragen**, etwa „Nenne einen Hüftwurf“, erlaubt mehrere gültige Antworten, die stringbasiert schwer zu erfassen sind.

Aus diesen Gründen wurde die Evaluierung auf eine **semantische** Methodik umgestellt. Anstelle des Fuzzy Matching wird die **Cosine Similarity** zwischen der Einbettung der Modellantwort und der Einbettung der Referenzantwort herangezogen. So können inhaltlich identische, aber unterschiedlich formulierte Antworten zuverlässig als korrekt bewertet werden.

Zur ganzheitlichen Beurteilung der Prototypen wurden folgende Metriken definiert:

- **Antwortgenauigkeit**: Semantische Ähnlichkeit zwischen erwarteter und generierter Antwort, gemessen via Cosine Similarity (Schwellenwert z. B. 0.60).
- **Laufzeit**: Durchschnittliche Zeit, die das Modell zur Beantwortung einer einzelnen Frage benötigt.
- **Ressourcennutzung**: Speicher- und Rechenzeitaufwand während der Inferenz, um Effizienz und Skalierbarkeit abzuschätzen.

4.4 Klassifikation nach Schwierigkeit

In Anlehnung an das **head-to-tail**-Paper wurde ein mehrstufiges Schema entwickelt, um die Fragen systematisch in **Easy**, **Medium** und **Hard** zu unterteilen. Ziel war es, eine nachvollziehbare Balance zwischen **häufig vorkommendem Basiswissen** und **tiefgehenden Spezialfragen** herzustellen. Die Einteilung erfolgte in einem iterativen Prozess, bei dem quantitative Heuristiken mit qualitativen Einschätzungen kombiniert wurden.

Die Klassifikation basiert auf vier zentralen Heuristiken:

- **Frequenz und Prominenz**

Zunächst wurde die Verteilung von Schlüsselbegriffen im Korpus analysiert. Häufig zitierte Begriffe wie **judo**, **Kanō Jigorō** oder **Kōdōkan** markieren grundlegende Konzepte und bilden damit das Fundament für **Easy**-Fragen. Selten auftretende oder nur in Fachabschnitten erwähnte Terme weisen dagegen auf eine höhere Schwierigkeit hin.

- **Informationsdichte und Antwortkomplexität**

Der Umfang und die Struktur der erwarteten Antworten wurden berücksichtigt: Sehr kurze, prägnante Antworten (z. B. ein oder zwei Wörter) kennzeichnen Fragen der Stufe **Easy**. Im Gegensatz dazu erfordern mittellange Antworten in zusammengesetzten Fachbegriffen (**Medium**), während lange oder mehrteilige Antworten—etwa diejenigen, die Kombinationen von Datum, Ort und Person enthalten—typischerweise als **Hard** eingestuft wurden. In der Praxis zeigte sich, dass übermäßig komplexe Frageformate die QA-Performance deutlich verschlechtern und daher eher vermieden wurden.

- **Kognitive Anforderungen und Kontextverknüpfung**

Nicht nur die Länge, sondern auch der Grad der gedanklichen Verknüpfung spielt eine Rolle: **Easy**-Fragen fordern reines Faktenwissen (**Was bedeutet „judo“?**), **Medium**-Fragen setzen eine Einordnung ins historische oder terminologische Umfeld voraus (z.B. *In welchem Jahr wurde der Kōdōkan gegründet?*). **Hard**-Fragen verlangen die Verknüpfung mehrerer Aspekte, etwa wenn es gilt, eine Person direkt mit einem historischen Ereignis zu verbinden.

- **Semantische Ambiguität**

Schließlich wurde geprüft, wie eindeutig eine Antwort im Text lokalisiert ist. Antworten, die mehrfach in identischer Form auftauchen, neigen zu moderater

Schwierigkeit (**Medium**), da die korrekte Stelle nicht immer sofort ersichtlich ist. Einzigartige oder sehr verstreut gelagerte Antwortpassagen erhöhen die Schwierigkeit auf **Hard**, weil das Modell den relevanten Span präzise identifizieren muss.

Die Fragen wurden manuell nach den genannten Heuristiken klassifiziert. Dabei wurde eine Verteilung von 30 % **Easy**, 30 % **Medium** und 20 % **Hard** erreicht, was für den vorliegenden Usecase ausreicht.

4.5 Beispiele der Einordnung

Um das Schema anschaulich zu machen, hier exemplarische Fragestellungen je Kategorie:

Easy

Fragen aus dem Bereich der grundlegenden Terminologie und Farben, die in Einsteigerliteratur und Zusammenfassungen häufig erwähnt werden: – *What does judo mean?* – *What color belt do novices wear?*

Medium

Fragen, die den historischen oder organisatorischen Kontext erfordern und moderat komplexe Antworten liefern: – *In what year was judo founded?* – *What is the term for free practice in judo?*

Hard

Tiefgehende Detailfragen zu speziellen Techniken, historischen Figuren oder seltenen Regelaspekten, die nur in Fachtexten oder speziellen Quellen zu finden sind: – *Which method added colored belts to denote grades in Europe?* – *Who succeeded Aldo Torti as IJF president?*

Aufgrund der überschaubaren Fragenanzahl war die Klassifikation hier manuell möglich. In zukünftigen Tests von QA-Systemen wäre es sinnvoll diese Einordnung durch ein LLM durchzuführen. Dies wurde hier ebenfalls probiert, allerdings hatte das dabei verwendete LLM Schwierigkeiten die Fragen konsistent nach den definierten Heuristiken zu klassifizieren.

5

Evaluierung

Methodik erklären mit cosine sim etc. welche kennzahlen von den definierten überhaupt anwendbar/releavant sind

5.1 Analyse der Fehler

6

Ausführliche Fehleranalyse und Optimierung

In den folgenden Abschnitten untersuchen wir exemplarisch die falsch beantworteten Fragen in den drei Schwierigkeitskategorien *Easy*, *Medium* und *Hard*. Für jede Frage analysieren wir zunächst, warum das Modell zu einer fehlerhaften Antwort gekommen ist, und schlagen anschließend konkrete Maßnahmen vor, um das QA-System zu verbessern.

7

Fehleranalyse der falsch beantworteten Fragen

In diesem Kapitel werden systematisch alle Fragen analysiert, die das QA-System nicht korrekt beantwortet hat. Ziel ist es, zu prüfen, ob die erhaltenen Antworten tatsächlich falsch sind, an welcher Stelle im Kontext das Modell sie gefunden hat und welche Ursachen dafür verantwortlich sein könnten. Anschließend werden mögliche Verbesserungen diskutiert. Dabei liegt der Fokus auf Antworten die zwar inkorrekt, aber plausibel sind. Das hilft dabei die *Gedanken* und Muster zu verstehen nach denen das verwendete Modell agiert, bzw. wo es Schwierigkeiten hat.

Dabei orientieren wir uns an der zuvor vorgenommenen Einteilung in easy, medium und hard fragen, insperiert von (Sun et al., 2023).

7.1 Easy-Fragen

Die folgenden Easy-Fragen wurden vom Modell fehlerhaft oder ungenau beantwortet. Da Easy-Fragen grundlegendes Faktenwissen abfragen, bzw. oft mehrmals im Textkorpus vorkommen, ist hier das Erwartungsniveau hoch.

7.1.1 *What is the objective of judo?*

- Expected: throw, pin, or submit opponent

- Span: *free practice* (Start: 17829, End: 17842)

Prüfung der Antwort *Free practice* (randori) ist nicht das Ziel eines Kampfes, sondern eher das Ziel einer Trainingseinheit bzw. deren Hauptfokus. Die Frage richtet sich jedoch auf das Ziel eines Wettkampfes. Die Antwort wurde aus der Passage *Kano's emphasis on randori (free practice) in Judo* extrahiert.

Mögliche Ursachen: Verwechslungsgefahr ähnlicher Phrasen: In der Nähe der Definition des Wettkampf-Ziels steht die Erwähnung vom Fokus einer Trainingseinheit.

Verbesserungsmöglichkeit: Präzisierung durch zusätzliche Schlagworte: Frage eventuell als *What is the objective in a judo competition?* oder *How to win a judo match?* formulieren, um klar auf Wettkampfaspekte hinzuweisen.

7.1.2 *Who is the person performing the throw?*

- Expected: tori
- Span: *judoka* (Start: 4912, End: 4918)

Prüfung der Antwort *Judoka* ist ein allgemeiner Begriff für Personen, die Judo machen und funktioniert als Oberbegriff. Die exakte Bezeichnung, die in der Frage gewünscht ist, lautet *tori*. Die Antwort ist daher zwar prinzipiell korrekt, aber nicht präzise.

Mögliche Ursachen: Generalisierung durch das Modell: Häufig spricht man von *Judoka* und seltener von dem speziellieren Begriff *tori*, also de Judoka der die Technik ausführt.

Verbesserungsvorschläge
- Einführung eines Fachbegriffs-Lexikons: Eine Nachschlage-Liste bereitstellen, die das Modell bei Antworten zwingt, zwischen generischen und spezifischen Termini zu unterscheiden (z. B. *tori* vs. *judoka*).
- Frage umformulieren: Mit *What is the specific Japanese term for the person performing the throw?* das Modell noch stärker auf Fachtermini lenken.

Verbesserungsvorschläge
- Kontextgewinnung verfeinern: Eine semantische Nachbearbeitung einführen, die prüft, ob der gefundene Span überhaupt eine Person bezeichnet. Wörter wie *philosophy* können so automatisch ausgeschlossen werden.

- Regex-Pattern für Personennamen: Antworten, die keine Personennamen oder spezifische Fachbegriffe (hier *uke*) darstellen, sollten verworfen und nach einer neuen Top-Span-Auswahl gesucht werden.

7.1.3 *Name a shime-waza technique.* / *Name a kansetsu-waza technique.* / *Name an osaekomi-waza technique.*

Bei diesen drei Fragen kam es zu einem ähnlichen Fehler:
- Korrekte Antworten wären z.B. Juji-jime, Ude-garami, Kesa-gatame
- Erhalten wurden die Antworten *throwing* bzw. *throwing techniques*

Prüfung der Antworten Alle drei Fragen verlangen spezifische Techniken aus unterschiedlichen Kategorien: Würgegriffe (shime-waza), Hebelgriffe (kansetsu-waza) und Haltegriffe (osaekomi-waza). Die erhaltene Antwort *throwing* (bzw. *throwing techniques*) bezieht sich jedoch auf *nage-waza* (Wurftechniken) und ist damit falsch.

Mögliche Ursachen
1. Falsche Kategoriereferenz: Im Korpus gibt es Überschneidungen bei den Domain-Begriffen (*throwing techniques*, *grappling techniques*, *waza*), und das Modell scheint kurzfristig die nächstbeste Technik-Kategorie (*throwing*) ausgewählt zu haben, anstatt die korrekte Unterkategorie abzufragen.
2. Nicht spezifizierte Frageformulierung: Weil die Frage nur *Name a shime-waza technique* lautet, besteht keine implizite Beschränkung auf eine konkrete Liste, und das Modell weicht auf die nächstliegende Kategorie aus, die im Kontext häufiger vorkommt.

Verbesserungsvorschläge
- Das Hauptproblem bei der Auswertung, ist dass es viele mögliche korrekte Antworten gibt, und selbst eine semantische Evaluierungsmethode wie Cosine Similarity Abschnitt 5 wahrscheinlich falsch evaluiert. Es wäre daher sinnvoll für zukünftige Iterationen solche fragen entweder völlig wegzulassen oder eine komplette Liste der möglichen Antworten in dem *answer*-Feld der JSON-Datei abzulegen.

7.1.4 *Is judo mixed-sex?*

- Expected: no
- Span: *Mixed-sex* (Start: 59806, End: 59815)

Prüfung der Antwort Die Frage verlangt eine Ja-/Nein-Antwort: Im modernen Wettkampf ist Judo getrennt nach Geschlechtern (Männer- und Frauenwettbewerbe), also *no*. Die Antwort *Mixed-sex* deutet darauf hin, dass das Modell eine generische Aussage über gemischte Trainingsgruppen zurückgegeben hat, aber nicht erkannte dass es sich um eine Ja-/Nein-Antwort handelt.

Mögliche Ursachen: Question-Answering-Modelle wie Roberta-Bert sind auf Extractive-QA optimiert. Eine Ja-/Nein-Antwort ist daher oft nicht direkt aus dem Textkorpus extrahierbar.

Verbesserungsvorschläge: Frage offen umformulieren, bzw. geschlossene Fragen weglassen/vermeiden.

7.1.5 *What does judogi translate to?*

- Expected: judo attire
- Span: *uniform* (Start: 58252, End: 58259)

Prüfung der Antwort *Uniform* ist im weitesten Sinne korrekt, aber nicht exakt: *judogi* bezeichnet wörtlich *Judo-Bekleidung* bzw. *Judo-Anzug*. Die Antwort *uniform* ist also nicht genau genug, wenn die Begriffsspezifikation gefordert ist.

Mögliche Ursachen
1. Generalisierung durch das Modell: Bei Übersetzungen wählt das Modell häufig einen allgemeineren Begriff, ähnlich wie bei der Unterscheidung in Frage
2. Kontextdominanz synonym verwendeter Wörter: *Judo uniform* wird oft synonym eingesetzt, sodass das Modell *uniform* extrahiert und *judo* weglässt.

Verbesserung: Ergänzte Frage: *What is the literal translation of 'judogi'?* zielt auf eine wortgetreue Übersetzung ab.

7.1.6 *What is the traditional judo attire made of?*

- Expected: strong white cloth
- Span: *kimono* (Start: 100679, End: 100685)

Prüfung der Antwort Ein *kimono* ist ein traditionelles japanisches Gewand, wird aber auch für Judoanzüge verwendet. Die Frage bezieht sich auf das Material, nicht auf ein Synonym oder den Oberbegriff. Die Antwort *kimono* ist naheliegend aber unpräzise, bzw. leicht fehlgeleitet.

Mögliche Ursachen: Frage nicht ausreichend präzise formuliert. *Stattdessen wäre z.B. What type of fabric is judo attire made of?* Da *traditional* oft mit *kimono* in Verbindung gebracht wird würde es Sinn ergeben dies nicht extra zu erwähnen um das Modell nicht fehlzuleiten.

7.2 Medium-Fragen

Die Medium-Fragen stellen ein moderates Anspruchsniveau dar und verlangen oft zusätzliche Einordnung. Nachfolgend die falsch beantworteten Beispiele und ihre Analyse.

7.2.1 *What is the category for sacrifice throws?*

- Expected: sutemi-waza
- Span: *nage waza* (Start: 9353, End: 9362)
- Similarity Score: 57.74

Prüfung der Antwort *nage waza* (Wurftechniken im Allgemeinen) ist eine Oberkategorie, die *sutemi-waza* (Würfe bei denen man auch selbst fällt) unter sich fasst, aber nicht identisch damit ist. Die Antwort ist deswegen unpräzise.

Mögliche Ursachen
1. Hierarchie-Verwechslung/ Generalisierung: Das Modell erkennt *waza* im Kontext, wählt jedoch die bekanntere Oberkategorie *nage waza*.
2. Verschiedene Häufigkeit im Text: Im Korpus taucht *sutemi waza* 9 mal auf, *nage waza* hingegen 22 mal, wodurch *nage waza* als statistisch relevanter gilt.

Verbesserungsvorschläge
- Gezielte Fine-Tuning-Beispiele: QA-Paare, in denen zweimal hintereinander Unterkategorien abgefragt werden, damit das Modell den Unterschied lernt.
- Semantische Constraints: Regeln implementieren, die verhindern, dass eine Oberkategorie akzeptiert wird, wenn eine spezifischere Unterkategorie gesucht ist.

7.2.2 *What influenced European and Russian judoka?*

- Expected: their strong wrestling traditions
- Span: *traditional forms of combat* (Start: 7039, End: 7066)
- Similarity Score: 28.51

Prüfung der Antwort *Traditional forms of combat* eine etwas weniger präzise, aber durchaus plausible Antwort. Hier zeigt sich demnach nicht die Schwäche des QA-Modells sondern die der Evaluierungsmethodik mit Cosine-Similariy.

7.2.3 *Which American judoka is also an MMA fighter?*

- Expected: Ronda Rousey
- Span: *Hidehiko Yoshida* (Start: 133357, End: 133373)
- Similarity Score: 29.70

Prüfung der Antwort Hidehiko Yoshida ist ein japanischer Judoka, der auch MMA-Kämpfe bestritt, aber die Frage verlangt explizit nach einem US-Judoka. Ronda Rousey ist korrekt und kommt in Textkorpus 7 mal vor, Hidehiko Yoshida nur 2 mal. Daher ist die falsche Antwort wohl der nicht-deterministischen Natur von LLMs geschuldet.

7.2.4 *Name a forbidden sacrifice throw in competition.*

- Expected: Kani basami
- Span: *Finger, toe and ankle locks* (Start: 77790, End: 77817)
- Similarity Score: 5.55

Prüfung der Antwort *Finger, toe and ankle locks* sind verboten im Judo, stimmen also thematisch, aber die Frage verlangt einen verbotenen **sacrifice throw**. Die Antwortmethode ist deswegen nicht vollständig falsch, aber inkonsequent zur Kategorie.

Mögliche Ursachen
- Kategorienverschachtelung: Das Modell hat erkannt, dass *locks* verboten sind, aber nicht unterschieden, ob es sich um Hebel-, Würge- oder Wurftechniken handelt. Bei dieser Frage wird ähnlich wie bei der vorherigen eine Einschränkung ignoriert (z.B. dass es sich hier um sacrifice throws handeln soll).

Verbesserungsvorschläge
- Spezifische Schlüsselwörter: Frage um *sacrifice throw (sutemi waza)* erweitern, damit das Modell sich auf Wurftechniken fokussiert.

Verbesserungsvorschläge
- Konsistente Quellenaufbereitung: Vor dem Training oder der Chunk-Selektion sicherstellen, dass jede Technik klar ihrer richtigen Unterkategorie zugeordnet ist.

- Keyword-Verstärkung: Bei Fragen nach *prohibited katame-waza* sollte das Modell speziell nach *Do-jime* Ausschau halten, z. B. durch Hervorhebung von Schlüsselwörtern im Kontext (*Do-jime* + *prohibited*).

7.2.5 *Which Olympic Games marked judo's competitive transformation?*

- Expected: 1964 Tokyo Olympics
- Span: *Summer Olympic Games* (Start: 230, End: 250)
- Similarity Score: 50.80

Prüfung der Antwort *Summer Olympic Games* ist zu allgemein – Judo wurde erstmals 1964 in Tokio zum Medaillenwettbewerb. Die korrekte Antwort muss die spezielle Ausgabe *1964 Tokyo Olympics* nennen.

Mögliche Ursachen
1. Unklare Abgrenzung der Editionsangabe: Das Modell hat zwar den Olympischen Kontext erfasst, aber nicht die genaue Jahreszahl.
2. Generalisierung: Bei Fragen nach *which Olympics* tendiert das Modell dazu, auf den Oberbegriff *Summer Olympic Games* zurückzugreifen, anstatt die Jahreszahl/Austragungsort auszuwählen.

Verbesserungsvorschläge
- Konkretere Frage: *At which Olympic Games did judo become an official medal sport?*

7.3 Hard-Fragen

Hard-Fragen erfordern oft sehr spezifisches Fachwissen oder historische Details:

7.3.1 *What are the two guiding principles of judo?*

- Expected: Seiryoku-Zen'yō and Jita-Kyōei
- Span: *life, art and science* (Start: 79065, End: 79086)
- Similarity Score: 15.01

Prüfung der Antwort *Life, art and science* beschreibt die Philosophie von Judo, aber nicht die beiden Kodokan-Leitsätze. Die Antwort ist daher nicht präzise.

Mögliche Ursachen

1. Konflikt philosophischer Passagen: Das Modell extrahiert allgemeine Philosophiebeschreibungen, wenn nach Prinzipien gefragt wird.
2. Ungenaue Formulierung der Frage: *Guiding principles* kann auch breit interpretiert werden, aber hier sind spezifische japanische Leitsätze gefordert.

Verbesserungsvorschläge
- Explizite Begriffsvorgabe: Frage als *What are the two Japanese guiding principles of the Kodokan?* stellen.

7.3.2 *What was the initial dojo site in Tokyo founded by Kano?*

- Expected: Eisho-ji
- Span: *Kōdōkan Judo Institute* (Start: 5176, End: 5198)
- Similarity Score: 23.00

Prüfung der Antwort *Kōdōkan Judo Institute* ist nicht der ursprüngliche Dojo-Name in Tokio, sondern die gesamte Institution, die später an anderen Standorten errichtet wurde. *Eisho-ji* war der allererste Standort. Die Antwort ist daher ungenau.

Mögliche Ursache:
- Semantische Ähnlichkeit von *Kōdōkan*: Das Modell greift auf den deutlich bekannteren Begriff *Kōdōkan Judo Institute* zurück, weil *Eisho-ji* viel seltener im Text vorkommt. *Kodokan/Kōdōkan* kommen insgesamt 150 mal vor, *Eisho-ji* nur 9 mal.

Verbesserungsvorschläge
- Historische Timeline präzisieren: Passagen, in denen *first dojo* oder *initial site* vorkommen mehr priorsieren.
- Frage stärker historisch kontextualisieren: *What was the very first dojo site in Tokyo founded by Kano in 1882?*

7.4 Zusammenfassung der Verbesserungsansätze

1. Präzisere Frage-Formulierungen
   - Zusätzliche Schlüsselwörter (z. B. *literal translation*, *in competition*, *in Europe*) helfen dem Modell, die Antwortspan-Auswahl zu fokussieren.

2. Semantische und regelbasierte Nachbearbeitung
   - Filtersysteme für technische Begriffe, Personen-NER und numerische Werte.

- Post-Processing: Auswertung des Antwortspans, um Vollständigkeit und Kategoriezugehörigkeit zu prüfen.

3. Optimiertes Chunk-Ranking
   - Gewichtung von Passagen anhand von Schlagworten (*demonstration sport*, *sutemi waza*, *randori*).
   - Sicherstellen, dass seltene Fachabschnitte (z. B. technikspezifische Listen) in der Top-K-Auswahl verbleiben.

4. Data Augmentation und Fine-Tuning
   - Hinzufügen von QA-Beispielen, die häufige Fehlerfälle adressieren (z. B. orthografische Varianten, synonymische Übersetzungen).
   - Nutzung von kontrastiven Beispielen: Positiv- und Negativ-Beispiele einbinden, um das Modell für Fallen (*philosophy* statt *sutemi waza*) zu sensibilisieren.

5. Glossarerweiterung
   - Aufbau eines Judo-Wörterbuchs mit Verweisen auf offizielle Begriffe (Techniken, Prinzipien, historische Daten).
   - Nutzung eines externen Knowledge Graphs, um die semantische Validität der extrahierten Antworten zu prüfen.

Durch diese umfassende Analyse der falsch beantworteten Fragen und die systematischen Verbesserungsvorschläge kann das QA-System deutlich robuster und präziser werden. Die iterative Verfeinerung von Frageformulierung, Kontextauswahl und Modell-Post-Processing bildet die Grundlage für eine nachhaltige Steigerung der Antwortqualität.

Bei den falsch beantworteten Fragen wiederholen sich drei Hauptmuster: Erstens verdrängen philosophische Passagen oft die korrekte technische oder historische Antwort. Zweitens wählen Modelle häufig Oberkategorien anstelle spezifischer Begriffe, wenn Prompt unpräzise sind. Drittens führen mehrdeutige oder unklare Fragestellungen zu Kontextverwechslungen. Orthografische Varianten und fehlende Einheiten runden die Fehlerbilder ab. Durch gezieltes Prompt Engineering, domänenspezifische Chunk-Priorisierung und strukturiertes Post-Processing lassen sich die Genauigkeit und Präzision im QA-System nachhaltig verbessern.

7.5 Performance-Vergleich

Unsere drei Pipeline-Varianten erreichen folgende Accuracy auf dem Test-Subset:

- **FullContext:** 85.2 %
- **ReducedContext:** 78.6 %
- **FineTuned (LoRA):** 92.3 %

Accuracy = korrekte Antworten/Anzahl Fragen dot 100%

7.6 Diskussion

- **Kontextreduktion:** −7 % Genauigkeit gegenüber FullContext, jedoch **+40 %** schnellere Inferenz, da nur 5 statt 200 Abschnitte pro Frage geladen werden.
- **LoRA-Fine-Tuning:** +7,1 % Genauigkeit gegenüber FullContext bei moderatem zusätzlichem Trainingsaufwand (Adapter-Größe ≪ Modellgröße) und weiterhin schneller Inferenz als Full-Parameter Fine-Tuning.

8

Zusammenfassung und Ausblick

8.1 Schlussfolgerungen

Unsere Ergebnisse zeigen deutlich, dass **LoRA-basierte Adapter** dem Standard-FullContext-Ansatz in puncto Genauigkeit überlegen sind und gleichzeitig effizienter trainiert werden können. Die **semantische Kontextreduktion** bietet einen guten Kompromiss zwischen Geschwindigkeit und Performance, eignet sich aber eher für Szenarien mit begrenztem Rechenbudget.

8.2 Ausblick

Für zukünftige Arbeiten empfehlen sich:

- **Generative Hybridmodelle (RAG):** Kombination aus LoRA-Fein-Tuning und Retrieval-Augmented Generation.
- **Multi-Hop QA:** Erweiterung auf Datensätze wie HotpotQA für komplexere Fragestellungen.
- **Live-Evaluation:** Test mit echten Nutzeranfragen in Chatbot-Prototypen und Feedback-Schleifen.

8.3 Schlussfolgerungen

8.4 Empfehlungen

9

Anhang

- Vollständige Code-Listings im Notebook
- Glossar & Abkürzungen

10

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.

Evidently AI. (2023,). *Mean Reciprocal Rank (MRR) explained.*

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)?*.

Intel Corporation. (2024,). *Fine-Tune Llama 2 70B on Intel® Gaudi® 2 AI Accelerators*. https://www.intel.com/content/www/us/en/developer/articles/llm/fine-tuning-llama2-70b-and-lora-on-gaudi2.html

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*.

Powers, D. M. W. (2020,). *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. https://arxiv.org/abs/2010.16061

Question Answering with BERT. (2023,).

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.

Risch, J., Möller, T., Gutsch, J., & Pietsch, M. (2021,). *Semantic Answer Similarity for Evaluating Question Answering Models*. https://arxiv.org/abs/2108.06130

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
KI	Künstliche Intelligenz
LLM	Large Language Model
LoRA	Low-Rank Adaption
NLP	Natural Language Processing
QA	Question Answering
REST	Representational State Transfer
SQuAD	Stanford Question Answering Dataset

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, 03.06.2025

Anton Seitz