



HTL Dornbirn

Höhere Lehranstalt für Wirtschaftsingenieurwesen
Ausbildungsschwerpunkt Betriebsinformatik

Diplomarbeit

**Chatbot zur gezielten Informationsfindung für die
Firma Blum**

Ausgeführt im Schuljahr 2023/24 von:

Leonardo Golubovic	5aWI
Michael Blessing	5aWI
Irsad Celik	5aWI

Dornbirn, am 02.04.2024

Danksagung

Wir möchten uns an dieser Stelle von ganzem Herzen bei allen Personen bedanken, die uns während unserer Diplomarbeit tatkräftig unterstützt, fachlich beraten und uns immer wieder motiviert haben. Eure Hilfe war sehr wertvoll und wir sind euch dankbar.

Besonders bedanken wir uns bei der Julius Blum GmbH, unserem Projektauftraggeber, bei welchem wir unsere Diplomarbeit durchführen durften. Wir bedanken uns auch bei Christian Gaisberger unserem Betreuer im Unternehmen. Er war immer für uns da, wenn wir mal Probleme hatten.

Ein großes Dankeschön an **Elias KLOCKER** für die Korrekturlesung unserer Diplomarbeit.

Zu guter Letzt bedanken wir uns bei unserem Betreuer in der Schule, Diethard Kaufmann, welcher uns im Laufe der Diplomarbeit stets zur Seite gestanden ist und uns bei der Dokumentation geholfen hat.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich den vorliegen Diplomarbeitsteil /die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe. Für die Erstellung der Arbeit habe ich auch folgendes generatives KI-Tool ChatGPT zu folgendem Zweck verwendet: Erklärungen. Die verwendeten Hilfsmittel wurden vollständig und wahrheitsgetreu inklusive Produktversion und Prompt ausgewiesen.

Dornbirn, am
.....

Michael Blessing

Dornbirn, am
.....

Irsad Celik

Dornbirn, am
.....

Leonardo Golubovic

Abstract

Our world is constantly being equipped with new technologies. Companies and businesses must also adapt to emerging technologies in order to remain competitive. Currently, the growth of one particular set of technologies, namely Artificial Intelligence, is increasing rapidly.

The aim of this thesis was to create a *chatbot* for users of any Blum customer site. This chatbot should make it easier for users to get the information and instructions they need than on the existing Blum websites. Since it would go beyond the scope of a diploma thesis to develop a chatbot of our own, a pre-trained chatbot, or a large language model (LLM), was selected based on various criteria, which is then fed with data to provide suitable results for users. This data is not available from the beginning, so it must be analysed, processed and passed to the LLM for training. The chatbot should contribute to the increased user-friendliness for website visitors. It is therefore crucial to ensure accessibility and usability in many different languages, as Blum has subsidiaries in many locations around the world where the chatbot will be used.

Vorwort

Unsere Welt wird kontinuierlich mit immer neueren Technologien ausgestattet. So müssen auch Unternehmen und Firmen sich an die anbahnenden Technologien anpassen, um ihre Marktanteile zu behalten. Momentan steigt das Wachstum von einer bestimmten Reihe an Technologien, nämlich der Künstlichen Intelligenz, stark an.

Das Ziel dieser Diplomarbeit war es, einen *Chatbot* für Benutzer jeglicher Blum Kundenseiten zu erstellen. Dieser Chatbot sollte es den Nutzern ermöglichen, leichter an gewollte Informationen, sowie Anweisungen zu kommen, als es auf den bisherigen Blum Webseiten ist. Da es den Rahmen einer Diplomarbeit sprengen würde, einen eigenen Chatbot zu entwickeln, wurde, anhand von verschiedenen Kriterien, ein vortrainierter Chatbot, oder ein *Large-Language-Model* (LLM) ausgewählt, welches dann mithilfe von Daten gefüttert wird, um passende Ergebnisse für Nutzer zu liefern. Diese Daten liegen bei Beginn nicht zur Verfügung, daher müssen sie ausgewertet, aufbereitet und dem LLM zum Training übergeben werden. Der Chatbot soll zur erhöhten Benutzerfreundlichkeit von Besuchern der Webseiten beitragen. Daher ist es auch nötig, den Chatbot für viele verschiedene Sprachen zugänglich und verwendbar zu machen, da Blum in vielen Standorten der Welt Tochtergesellschaften hat, bei denen der Chatbot dann auch in Verwendung ist.

Hinweise zur Formatierung

Die Textformatierungen, welche in dieser Diplomarbeit vorkommen und ihre Erklärung sind in der folgenden Tabelle dargestellt:

Formatierung	Erklärung
<i>Kursiv</i>	Kursiv markierte Wörter sind im Glossar aufzufinden. Dabei ist nur die erste Erscheinung des Wortes kursiv markiert.
„Anführungszeichen und Kursiv“	Textstellen, welche in Anführungszeichen stehen und kursiv markiert sind, stellen ein direktes Zitat dar.
Fett, 14P	Kleinere Überschriften werden fett mit Schriftgröße 14 markiert.
Abkürzungen (Abk.)	Abkürzungen werden beim ersten Vorkommen ausgeschrieben, mit der jeweiligen Abkürzung in einer nachfolgenden Klammer.
72 Monospace blau	Blau markierte Textstellen, welche in 72 Monospace formatiert sind, stellen einzelne Code-Stellen dar.

Inhalt

1.	Impressum.....	1
1.1.	Projektteam	1
1.2.	Projektbetreuung.....	2
2.	Julius Blum GmbH	3
3.	Beschreibung der Diplomarbeit	4
3.1.	Ausgangssituation.....	4
3.2.	Ziel der Diplomarbeit	5
3.3.	Lösungsansatz	6
4.	Projektmanagement	7
4.1.	Projektauftrag	8
4.2.	Projektstrukturplan.....	9
4.3.	Objektstrukturplan.....	11
4.4.	Projektzieleplan.....	12
4.5.	Meilensteinplan	14
4.6.	Projektumweltanalyse	15
4.7.	Zeiterfassung.....	16
4.8.	Projektabchluss	17
5.	Evaluierung.....	18
5.1.	Vorevaluierung.....	20
5.1.1.	Chatbots/Chatbot-Anbieter.....	21
5.1.2.	Large Language Models	28
5.2.	Verkleinerung der Auswahl von LLMs.....	37
5.3.	Evaluierung der übrigen Modelle	38
5.3.1.	Kriterien für die Evaluierung.....	40
5.3.2.	Relevante Kriterien	42
5.3.3.	Technische Umsetzung der Evaluierung.....	43
5.4.	Entscheidung des LLMs	45
5.5.	Evaluierung - Fazit	49
6.	Data-Cleaning.....	50
6.1.	Bedeutung des Data-Cleaning für einen Chatbot	52
6.2.	Programmiersprache	53
6.3.	Libraries.....	61
6.4.	Struktur der Daten	63
6.5.	Auswirkungen sauberer Daten auf Chatbot-Leistung.....	65
6.6.	Herausforderungen.....	66

6.7.	Technische Umsetzung	68
7.	Training der LLM	77
7.1.	Ziel des Trainings.....	77
7.2.	Architekturen von neuronalen Netzwerken	78
7.3.	Vortrainierte neuronale Netzwerke.....	82
7.3.1.	Vorteile	84
7.3.2.	Nachteile.....	85
7.4.	Arten des Finetunings	87
7.5.	Trainingsarten	90
7.6.	Einfluss der Hyperparameter	92
7.7.	Testen mit einem kleinen Datensatz	97
7.7.1.	Entscheidung der Trainingsmethode.....	98
7.7.2.	Optimale Hyperparameter Englisch	101
7.7.3.	Optimale Hyperparameter Deutsch	104
7.8.	Trainieren anhand der Blum-Daten	107
7.9.	Technische Umsetzung	117
8.	Deployment.....	126
8.1.	Deployment-Optionen	127
8.2.	Technische Umsetzung	134
8.3.	Testen.....	140
8.4.	Qualitätsmanagement	143
8.5.	Treiber.....	144
9.	API-Webseitenanbindung	148
9.1.	Programmiersprache	149
9.2.	Libraries.....	151
9.3.	Technische Umsetzung	156
10.	Fazit.....	164
11.	Verwendete Technologien.....	166
11.1.	Visual Studio Code	166
11.2.	Termius	166
11.3.	Microsoft Teams	167
11.4.	Microsoft Word.....	167
11.5.	Discord	167
11.6.	Notepad++	168
11.7.	Huggingface.....	168
11.8.	PEFT.....	168
11.9.	Transformers.....	169

11.10.	PyTorch	169
11.11.	CUDA.....	169
11.12.	Azure.....	170
11.13.	PyCharm.....	170
11.14.	Postman.....	170
11.15.	JavaScript.....	171
11.16.	Python.....	171
11.17.	Pandas.....	171
11.18.	React	172
11.19.	Axios.....	172
11.20.	CSS	173
11.21.	Docker.....	173
11.22.	OpenLLM.....	173
12.	Glossar.....	174
13.	Verzeichnisse.....	179
13.1.	Abkürzungsverzeichnis.....	179
13.2.	Autorenverzeichnis	182
13.3.	Abbildungsverzeichnis	183
13.4.	Code-Verzeichnis.....	185
13.5.	Tabellenverzeichnis.....	186
13.6.	Quellenverzeichnis.....	187

1. Impressum

Das Impressum beinhaltet alle Personen, welche hauptsächlich an der Diplomarbeit beteiligt waren, oder zu ihrer Unterstützung beigetragen haben.

1.1. Projektteam

Das Projektteam setzt sich aus folgenden Schülern, welche am Projekt arbeiteten, zusammen:

Projektleiter

Leonardo Golubovic

Leonardo Golubovic ist für die Projektleitung und die Kommunikation mit der Partnerfirma zuständig. Er recherchierte und evaluierte verschiedene LLMs und Chatbots. Zudem ist er zuständig für das Deployment des Chatbots.

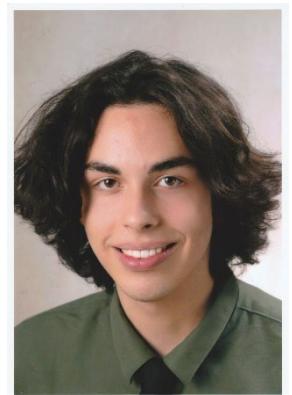


Abbildung 1: PL Leonardo Golubovic

Projektteammitglieder

Michael Blessing

Michael Blessing ist für das Aufsetzen der virtuellen Maschine (VM) und das Trainieren des LLMs mithilfe der Trainingsdaten zuständig. Er recherchierte außerdem nach mehreren Trainingsmethoden, mit denen man LLMs trainieren kann.



Abbildung 2: PM Michael Blessing

Irsad Celik

Irsad Celik ist zuständig für das Aufbereiten der Trainingsdaten und beschäftigt sich mit der Entwicklung einer zugehörigen Webapplikation. Zudem recherchierte er mehrere Methoden zur Datenverarbeitung.



Abbildung 3: PM Irsad Celik

1.2. Projektbetreuung

Die Projektbetreuung ist für die Aufsicht des Projektteams zuständig.

Betreuer der Firma Blum

Christian Gaisberger

Christian Gaisberger war für die Betreuung bei der Firma Blum zuständig.

Die Absprache über die Richtung des Projekts lief mit ihm ab.



Abbildung 4: AG Christian Gaisberger

Betreuer der HTL-Dornbirn

Diethard Kaufmann

Diethard Kaufmann war für die Betreuung seitens der HTL-Dornbirn zuständig. Er unterstützte das Team bei der Erstellung der Dokumentation.



Abbildung 5: PB Diethard Kaufmann

2. Julius Blum GmbH



Abbildung 6: Partnerfirma Julius Blum GmbH¹

Die Julius Blum GmbH ist ein Unternehmen, welches Möbelbeschläge herstellt und sich mit Wohnqualität auseinandersetzt². Blum wurde 1952 von Julius Blum als Familienunternehmen gegründet und ist auch heute noch im Besitz der Familie. Blum ist mit 32 Tochtergesellschaften und Repräsentanzen weltweit vertreten und hat zurzeit 8 Werke in Vorarlberg.

Blum bietet eine Vielzahl von Produkten und Services für die diversen Bereiche eines Wohnraums. Viele Produkte wie Kästen, Schränke und Schubladen verwenden die von Blum erstellten Scharniere und Rollschubführungen.

¹ „Firmenmitglieder Blum – Julius Blum GmbH | Bundesverband Industrie Kommunikation e.V.“

² GmbH, „Blum moving ideas“.

3. Beschreibung der Diplomarbeit

In diesem Kapitel werden die Ausgangssituation der Diplomarbeit sowie die Ziele des Projekts erläutert. Unter der Ausgangssituation ist die Problemstellung des Unternehmens zu verstehen. Zusätzlich wird der Lösungsansatz zu dieser Problemstellung kurz angesprochen.

3.1. Ausgangssituation

Das Unternehmen Julius Blum GmbH hat seine Produkte und Dienstleistungen auf verschiedenen Webseiten ihrerseits hinterlegt. Besucher und Besucherinnen dieser Webauftritte stehen dadurch vor einer Vielzahl an Kanälen, von denen sie aussuchen können. Davor führen allerdings nur bestimmte Webseiten zum Produkt oder zur Dienstleistung, die ein Kunde oder eine Kundin beziehen will. Die Suchfunktion lässt dabei auch zu wünschen übrig, da sie meistens nur auf Elemente der jeweiligen Webseite zugreifen kann und zudem ungenau bei Suchen, die auf der Katalognummer basieren, arbeiten, da sie dann meistens auf einen unübersichtlichen Blätterkatalog verweisen, in dem der Kunde oder die Kundin dann wieder sein oder ihr Produkt mühevoll aussuchen kann. Benutzer und Benutzerinnen müssen sich außerdem auf den Blum-Kanälen anmelden, um Zugriff auf die Produkte und die Suchfunktion zu erhalten.

3.2. Ziel der Diplomarbeit

Das Ziel dieser Diplomarbeit ist es dabei, einen Chatbot mit bereits zur Verfügung stehendem Inhalt zu trainieren, sodass die Antworten Sinn ergeben und einen User besser zum Ziel, in dem Fall ein Produkt oder eine Dienstleistung, führen zu können. Konkreter wurden mehrere Ziele gesetzt, welche mit dem Chatbot erreicht werden sollten. Darunter zählen zum Beispiel:

- Der Chatbot ist mit allen erforderlichen Blum Produktdaten, so wie sie von Blum öffentlich bereitgestellt werden, trainiert.
- Das *fine-tuned* Large-Language-Model gibt akkurat die geforderten Produktergebnisse zurück.
- Der Zugriff auf den Chatbot ist mittels einer *REST-API* möglich.
- Die Unterstützung von 2 Sprachen soll beim Chatbot gegeben sein, in diesem Fall für Deutsch und Englisch.

Zusätzlich wurde ein Zusatzziel gesetzt, welches die REST-API verwendet, um eine Webanwendung zu schreiben, mit welcher dem Chatbot Fragen gestellt werden können und darauf Antworten erhalten werden.

3.3. Lösungsansatz

Als Lösungsansatz wurde, wie auch mit dem Partnerunternehmen abgesprochen, ein bereits mit Grunddaten vtrainiertes Large-Language-Model (LLM) als Basis des Chatbots genommen. Dieses LLM soll dann mit den vorhandenen Blum Produktdaten erweitert werden. Anschließend soll das LLM über eine virtualisierte Umgebung, vorzugsweise Docker, bereitgestellt werden. Dabei kann man mittels eines Representational State Transfer (REST) – Application Programming Interface (API) Anfragen an den Chatbot stellen.

4. Projektmanagement

Die Projektpläne, welche im nachfolgenden Abschnitt beschrieben sind, wurden zu Beginn der Diplomarbeit erstellt. Wann es nötig war, wurden diese im Laufe der Diplomarbeit angepasst, beziehungsweise (bzw.) abgeändert. Die Projektpläne sind für die Planung unerlässlich. Ebenso helfen sie dabei, den Zeitplan für die verschiedenen Projektphasen der Diplomarbeit einzuhalten.

4.1. Projektauftrag

Im Projektauftrag, welcher in der nachfolgenden Abbildung zu sehen ist, wird der genaue Projektrahmen festgelegt und die wichtigsten Punkte werden zusammengestellt. Es kann aus dem Projektauftrag, welcher in Abbildung 7 zu sehen ist, entnommen werden, dass das Startereignis des Projekts das Kickoff-Meeting bei der Firma Blum war. Jedoch begann das Projekt erst am 26.09.2023. Die Diplomarbeit muss bis zum 02.04.2024 abgegeben werden. Die Einreichung der Dokumentation markiert das Ende der Diplomarbeit. Der formale Abschluss findet bei Präsentation der Diplomarbeit statt.³

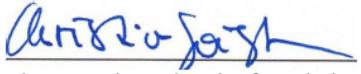
PROJEKT-AUFTAG							
Projektstartereignis: • Kickoff Meeting bei der Firma Blum	Projektstarttermin: • 26.09.2023						
Inhaltliches Projektendereignis: • Fertigstellung der Dokumentation Formales Projektendereignis: • Präsentation der Diplomarbeit	Projektendtermine: • 02.04.2024 • 13.06.2024						
Projektziele: • Chatbot ist mit allen erforderlichen Blum Produktdaten trainiert • LLM gibt akkurat die geforderten Produktergebnisse zurück • Zugriff auf den Chatbot ist möglich • Unterstützung von 2 Sprachen (Deutsch und Englisch)	Nicht-Projektziele: • Chatbot zur puren Kommunikation (ChatGPT-ähnlich) • Algorithmus • Neues LLM erstellen • Unterstützung aller Sprachen im Blum-Konzern						
Hauptaufgaben (Projektphasen): • Projektmanagement • Planungsphase • Vorbereitungsphase • Durchführungsphase • Kontrollphase • Dokumentation	Projektressourcen und -kosten: <table border="1"> <thead> <tr> <th>Ressourcen-/Kostenart</th><th>Mengeneinheit</th><th>Kosten (in Euro)</th></tr> </thead> <tbody> <tr> <td>Virtuelle Maschine</td><td>1</td><td>17.59 €/ Verwendete Stunde</td></tr> </tbody> </table>	Ressourcen-/Kostenart	Mengeneinheit	Kosten (in Euro)	Virtuelle Maschine	1	17.59 €/ Verwendete Stunde
Ressourcen-/Kostenart	Mengeneinheit	Kosten (in Euro)					
Virtuelle Maschine	1	17.59 €/ Verwendete Stunde					
Projektauftraggeber: • Christian Gaisberger	Projektleiter: • Leonardo Golubovic						
Projektteam: • Irsad Celik • Michael Blessing							
 Christian Gaisberger, (Projektauftraggeber)	 Leonardo Golubovic, (Projektleiter)						

Abbildung 7: Projektauftrag

³ Schnurr, „Was ist ein Projektauftrag?“

4.2. Projektstrukturplan

Der Projektstrukturplan (PSP) stellt eine strukturierte Übersicht eines Projekts dar. Durch den Projektstrukturplan wird das Projekt in verschiedene Phasen gegliedert. Diese enthalten dann jeweils eigene Arbeitspakete und Meilensteine. Durch den Einsatz des Projektstrukturplans können eine effektive Zeitplanung und ihre Umsetzung erreicht werden.⁴

In dieser Arbeit wurden die Phasen des Projektes und die genaue Aufteilung der Arbeitstakte am Anfang der Diplomarbeit festgelegt. Wie man in der nachfolgenden Abbildung erkennen kann, sind Meilensteine im Projektstrukturplan als Raute gekennzeichnet und sind blau, zusätzliche Meilensteine sind lila. Die normalen Arbeitspakete werden als orange Rechtecke dargestellt, Zusatzpakete als rote. Der Projektstrukturplan hat zu einem guten Überblick über den Fortschritt bei der Diplomarbeit verholfen.

⁴ „BET-PM 08 PM Projektplanung FRB 2021.pdf“.

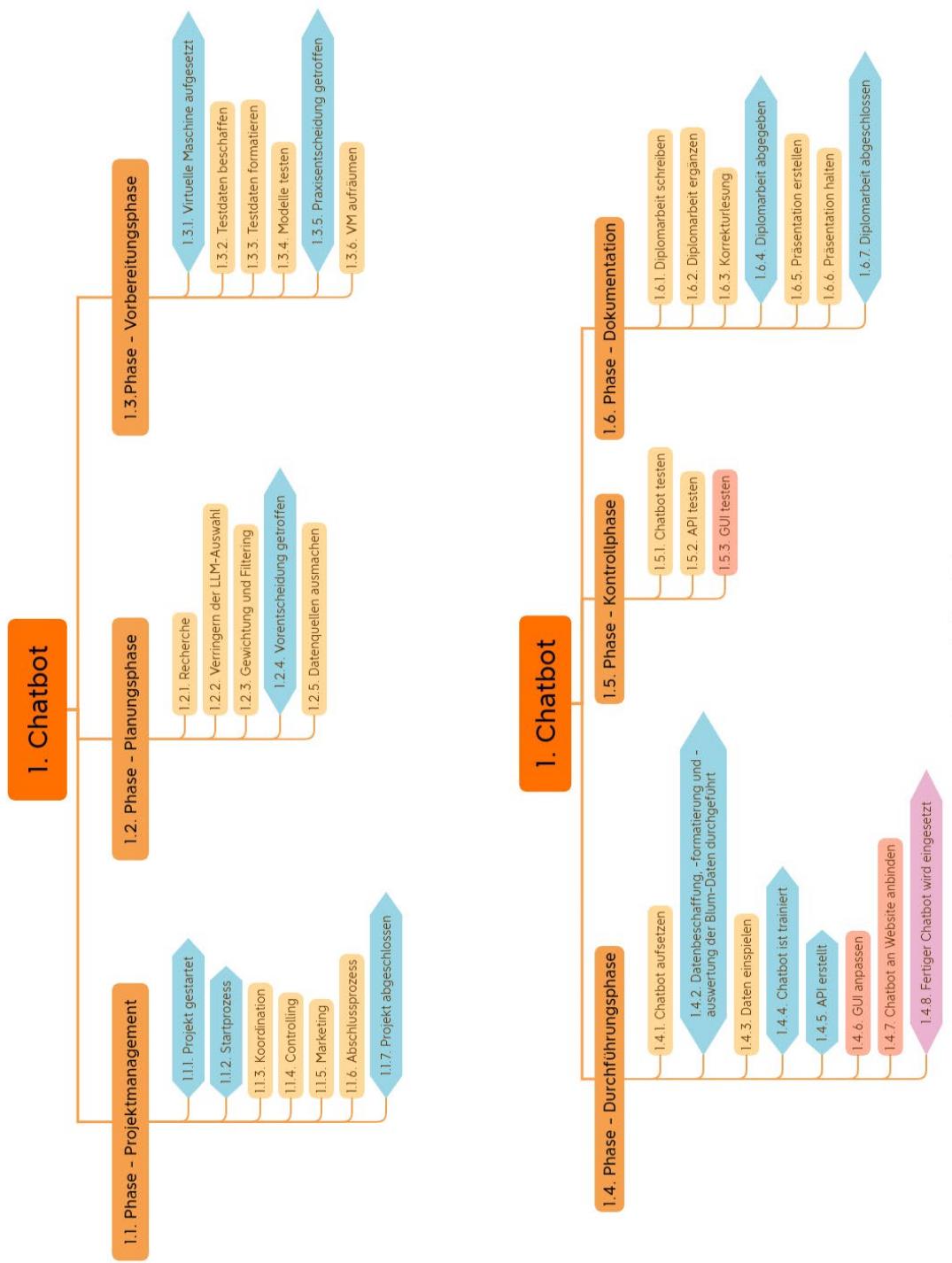


Abbildung 8: PSP

4.3. Objektstrukturplan

Der Objektstrukturplan (OSP) ist gegenüber dem Projektstrukturplan durch unterschiedliche Gliederungsprinzipien gekennzeichnet. Während der Projektstrukturplan eine organisatorische Gliederung des Projektes vornimmt, liegt der Schwerpunkt des Objektstrukturplanes auf der fachlichen bzw. technischen Gliederung des Projektes. Der Objektstrukturplan trägt dazu bei, das Projekt in klar definierte Objekte zu gliedern.⁵

Diese Diplomarbeit ist in vier Teile gegliedert, nämlich Hardware, Anbindung, Sprachmodell und Trainingsdaten. Man kann diese Aufteilung in der Abbildung darunter sehen.

Die vier Teile werden wiederum in fachliche, beziehungsweise technische Objekte unterteilt.

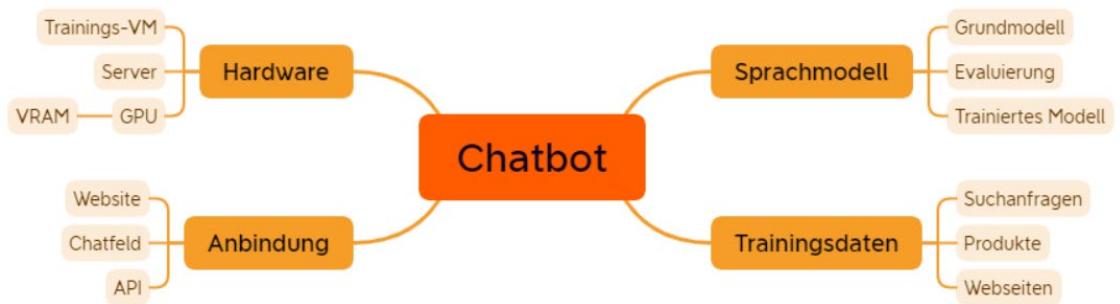


Abbildung 9: OSP

⁵ „BET-PM 08 PM Projektplanung FRB 2021.pdf“.

4.4. Projektzieleplan

Der Zieleplan legt genau fest, was Teil der Diplomarbeit ist und was nicht. Die Anforderungen an das Projekt werden präzise beschrieben und in erreichbare Ziele umgesetzt. Durch das Setzen dieser Ziele wird der Rahmen für das Projekt festgelegt. Im Projektzieleplan unterscheidet man zwischen Hauptzielen, Zusatzz Zielen und Nicht-Zielen. Die Hauptziele müssen erreicht werden, um das Projekt erfolgreich abschließen zu können, während die Zusatzz Zielen optional sind und zur Weiterentwicklung des Projekts genutzt werden können. Nicht-Zielen sind dazu da, den Projektumfang abzugrenzen. Sie legen fest, was das Projekt nicht umfassen soll. Somit stellt man sicher, dass die verfügbaren Ressourcen gezielt für die relevanten Aufgaben genutzt werden.⁶

In dieser Diplomarbeit wurden hauptsächlich jene Komponenten als Hauptziele definiert, die für die Nutzung und die Integration des Chatbots in eine Website erforderlich sind. Aus Abbildung 10 lassen sich die Hauptziele sowie die zusätzlichen Ziele und die Nicht-Ziele entnehmen.

⁶ „Projektzieleplan | Diplomarbeiten an Berufsbildenden Schulen | Bundesministerium für Bildung, Wissenschaft und Forschung“.

Projektzieleplan <small>Chatbot zur gezielten Informationsfindung</small>	
Zielart	Projektziele
Ziele: <ul style="list-style-type: none"> • Hauptziele • Zusatzziele 	<ul style="list-style-type: none"> • Chatbot ist mit allen erforderlichen Blum Produktdaten trainiert • Chatbot gibt akkurat die geforderten Produktergebnisse zurück • Zugriff auf den Chatbot ist über eine API möglich • Unterstützung von 2 Sprachen (Deutsch und Englisch) • Unterstützung von mehreren Sprachen • Zugriff auf den Chatbot ist über eine Webseite/Webapplikation möglich
Nicht-Ziele	<ul style="list-style-type: none"> • Chatbot zur puren Kommunikation (ChatGPT-ähnlich) • Algorithmus • Neues LLM erstellen • Unterstützung aller Sprachen im Blum-Konzern

Abbildung 10: Projektzieleplan

4.5. Meilensteinplan

Der Meilensteinplan ist eine Methode der groben Terminplanung für Projekte. Sein Schwerpunkt liegt auf der Definition und Festlegung von Meilensteinen, die Ereignisse von besonderer Bedeutung im Projektlauf markieren und wesentliche Fortschritte repräsentieren. Meilensteine im Projektstrukturplan werden üblicherweise als Rauten oder Sechsecke dargestellt und sind durch ihre zeitliche Ungebundenheit sowie ihre Funktion als Anfangs- oder Endereignisse von Arbeitspaketen gekennzeichnet.⁷

Wie man in der nachfolgenden Abbildung erkennen kann, unterscheiden sich die IST-Termine zum Teil von den aktuellen Planterminen. Das liegt daran, dass das *Data-Cleaning* unterschätzt wurde und das Gerät des zuständigen Projektmitgliedes zu dieser Zeit einige Probleme bereitete.

Chatbot zur gezielten Informationsfindung		PROJEKT- MEILENSTEINPLAN	
PSP- Code	Meilenstein	Aktuelle Plantermine	Ist Termine
1.1.1.	Projektstart	26.09.2023	26.09.2023
1.1.2.	Projekteinreichung	08.10.2023	07.10.2023
1.3.1.	VMs sind aufgesetzt	03.11.2023	25.10.2023
1.3.5.	Evaluierung ist abgeschlossen und Entscheidung gefällt	24.11.2023	20.11.2023
1.4.2.	Erforderliche Daten zum Modelltraining stehen bereit	14.12.2023	03.02.2024
1.4.4.	Grundmodell ist mit allen erforderlichen Daten trainiert und einsatzbereit	12.01.2024	06.02.2024
1.4.5.	Der Zugriff auf den Chatbot ist mittels API möglich	16.02.2024	21.02.2024
1.4.8	Der Zugriff auf den Chatbot ist über eine Webapplikation möglich	Zusatzmeilenstein	20.03.2024
1.6.4.	Abgabe der Diplomarbeit	02.04.2024	02.04.2024
1.6.7.	Präsentieren der Diplomarbeit	13.06.2024	13.06.2024
1.1.7.	Projektende	13.06.2024	13.06.2024

Abbildung 11: Projektmeilensteinplan

⁷ „BET-PM 11 PM Terminplanung FRB 2022.pdf“.

4.6. Projektumweltanalyse

Die Projektumweltanalyse ist von großer Bedeutung im Projektmanagement. Ihr Zweck besteht darin, die Beziehungen eines Projekts zu seiner internen und externen Umgebung zu prüfen und zu bewerten. Ein Projekt steht stets in Beziehung mit verschiedenen internen und externen Faktoren, beispielsweise dem Projektteam, dem Projektmanager, den Lieferanten, Kunden, Abteilungen und Banken.⁸

Nur die für das Projekt relevanten Beziehungen werden untersucht und bewertet. Um die Stärke und Art der Einflüsse darzustellen, werden diese Beziehungen mit bestimmten Symbolen klassifiziert. Eine unterstützende Beziehung wird durch ein positives Zeichen (+,++) gekennzeichnet, während eine hinderliche Beziehung durch ein negatives Zeichen (-,--) gekennzeichnet ist. Eine neutrale Bewertung (0) zeigt eine Beziehung an, die das Projekt weder hindert noch unterstützt.⁹

Auf der Grundlage der Bewertungsergebnisse werden zur Förderung positiver Beziehungen und zur Minimierung oder Hinderung negativer Auswirkungen geeignete Maßnahmen oder Strategien entwickelt. Wie man in Abbildung 12 sieht, gibt es in dieser Diplomarbeit keine Probleme im Zusammenhang mit den Beziehungen. Die ITO ist die Abteilung, in der das Projekt stattfindet. Unter dem Berater wird die Kundenberatung von Blum verstanden.

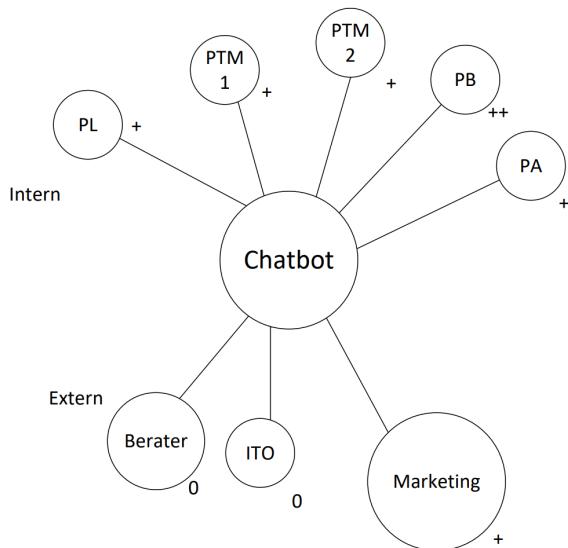


Abbildung 12: Projektumweltanalyse

⁸ „BET-PM 07a Projektumweltanalyse FRB 2021.pdf“.

⁹ „BET-PM 07a Projektumweltanalyse FRB 2021.pdf“.

4.7. Zeiterfassung

In der nachfolgenden Tabelle sind die Arbeitszeiten der Projektmitglieder und des Projektleiters eingetragen, sowie die Gesamtarbeitszeit. Die Arbeitszeiten wurden auch als Kreisdiagramm dargestellt. Diese kann man in Abbildung 13 sehen.

Tabelle 1: Zeiterfassung

Projektteam	Stunden
Leonardo Golubovic	172,26 Stunden
Michael Blessing	169,21 Stunden
Irsad Celik	163,21 Stunden
	504,7 Stunden

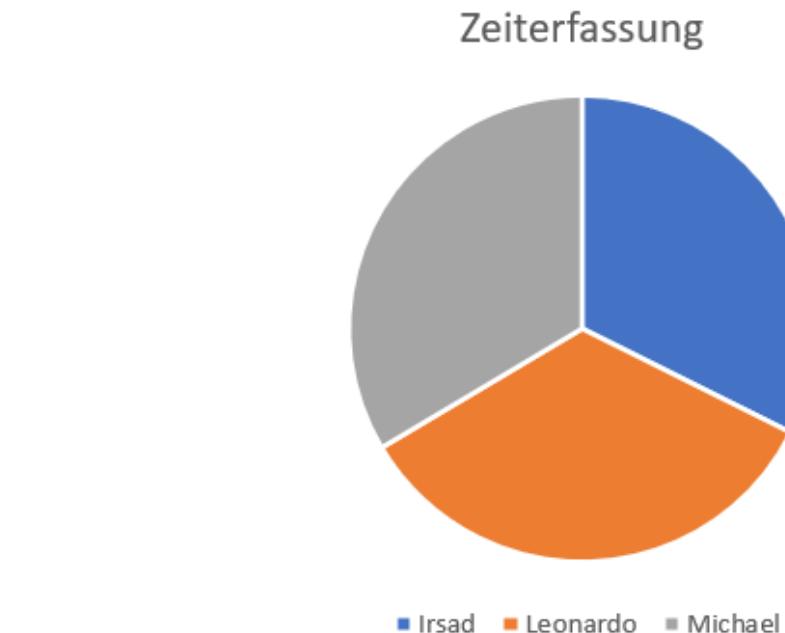


Abbildung 13: Zeiterfassung

4.8. Projektabschluss

Aufgrund dessen, dass das Projekt zwei Tage vor der Projektabgabe fertig wurde und der Projektauftraggeber zu diesem Zeitpunkt nicht erreichbar war, konnte kein Projektabschlussmeeting abgehalten werden. Deshalb erfolgt der Projektabschluss mit der Abgabe der Diplomarbeit.

5. Evaluierung

Unter der Evaluierung wird der Prozess der Bewertung eines Vorgangs, einer Leistung, eines Produkts oder einer Methode verstanden. Üblicherweise werden mehrere Angebote evaluiert, wobei eine Bewertung und zwischen verschiedenen Angeboten auch ein Vergleich entstehen kann. Es werden mehrere Kriterien bei Evaluierungen bewertet, bepunktet und gewichtet, je nach Erachten der Wichtigkeit der Kriterien der Person, welche die Evaluierung durchführt.

Bei der Evaluierung von LLMs geht es dabei um technische sowie ethische Faktoren, welche in Betracht gezogen werden. Für die Evaluierung gibt es mehrere Verfahren, welche angewendet werden können, wie zum Beispiel Bilingual Evaluation Understudy (BLEU), welches für die Übersetzung eines Textes durch das LLM zuständig ist, wobei unter Übersetzung die Umstellung eines gegebenen Satzes gemeint wird, bei dem dann geschaut wird, ob dieser noch denselben Sinn des ursprünglichen Satzes beibehält. Für die verschiedenen Evaluierungsmethoden gibt es auch verschiedene *Frameworks*, die verschiedene Aspekte eines LLMs evaluieren, wie zum Beispiel das Stanford Question Answering Dataset (SQuAD), welches Textverständnis evaluiert oder das Massive Multitask Language Understanding (MMLU), welches Multitasking evaluiert¹⁰.

Für die Evaluierung wurde aus kosten- und zeitbezogenen Gründen der Ansatz der menschlichen Evaluierung verwendet, wobei sich diese Evaluierung mehr auf qualitative Antworten und Ergebnisse durch gezielte Fragen bezieht. Diese Evaluierung erfolgt üblicherweise durch einen Menschen, der die Fragen stellt und dann bewertet. Dabei besteht jedoch die Gefahr, dass die Bewertung mehr subjektiv als objektiv ausfällt.

¹⁰ „Evaluating Large Language Models“.

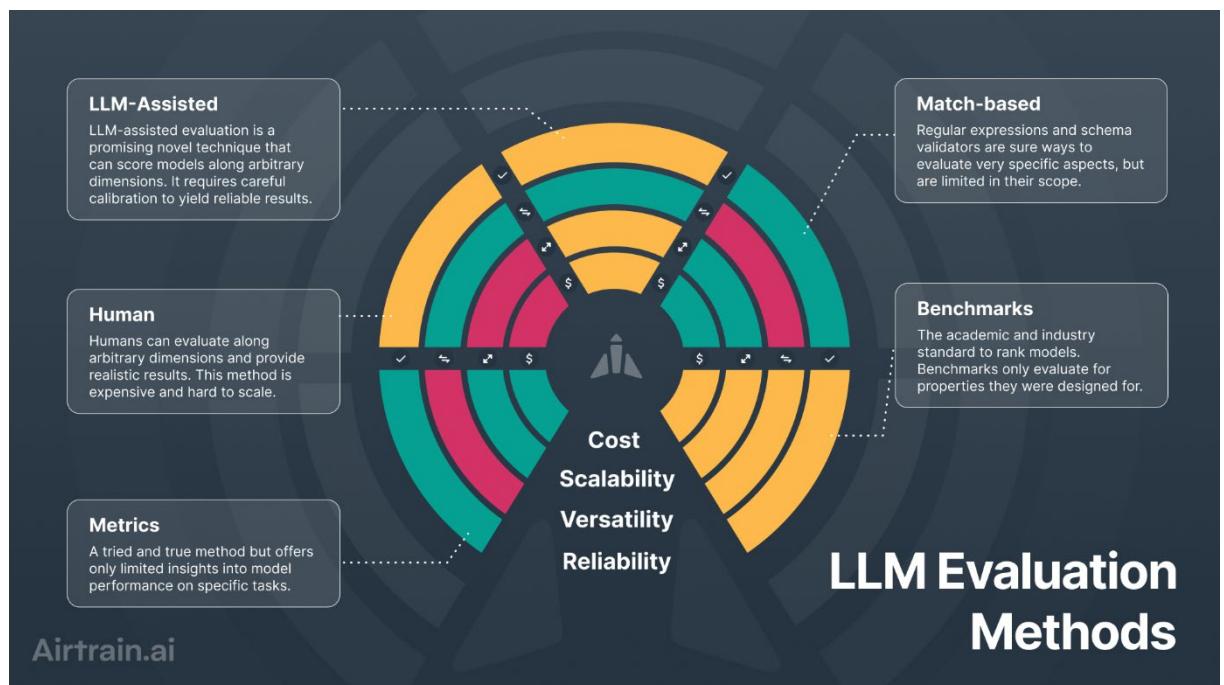


Abbildung 14: Evaluierungsmethoden von Chatbots¹¹

Abbildung 14 zeigt verschiedene Evaluierungsmethoden, die für das Evaluieren von LLMs geeignet sind. Die menschliche Evaluierung hat dabei den Nachteil, dass sie im selben Ausmaß wie andere Evaluierungsmethoden teurer und weniger skalierbar ist. Für diese Diplomarbeit eignet sich nur die Evaluierung durch einen Menschen, aufgrund der Kosten, Zeit und Verfügbarkeit, die andere Evaluierungsmethoden beanspruchen. Eine menschliche Evaluierung eignet sich, weil nur eine grobe Bewertung entstehen soll und dadurch in einem kleineren Ausmaß evaluiert wird, als es üblich ist, wodurch die menschliche Evaluierung am günstigsten ist.

¹¹ „The Comprehensive Guide to LLM Evaluation“.

5.1. Vorevaluierung

Bei der Vorevaluierung wurden mehrere LLMs und Chatbot-Anbieter in Betracht gezogen und mithilfe von bestimmten Kriterien ausgeschieden, sodass nicht alle in Betracht gezogenen LLMs dann auch in der Performanz getestet werden müssen. Dabei wurden nicht nur LLMs angeschaut, sondern auch direkte Möglichkeiten zum Erstellen oder Trainieren von Chatbots außerhalb von LLMs, wie Chatbot-Anbieter, die jedoch aufgrund von Richtlinien des Partnerunternehmens nicht verwendet werden können, da keine Transparenz beim *Training* in der Datenverarbeitung bei den Anbietern besteht und das Partnerunternehmen nicht will, dass ihre Produktdaten unbefugt verwendet werden. Da die Chatbots, ob alleingestellt oder von Anbietern zum Fine-tunen angeboten, selbst auf einem LLM basieren, wird auch das Grundmodell solcher Angebote in Erwägung gezogen. Die ausschlaggebenden Kriterien in der Vorevaluierung waren die Kosten, die ein Angebot oder LLM verursacht, bzw. das Kostenmodell dieses Angebots, ob man das Modell auf einer eigenen physischen oder virtuellen Maschine hosten konnte, sowie konkrete Anwendungsgebiete, die zum Ausschluss eines Angebots führen können. Ein weiteres Kriterium, welches in Betracht gezogen wurde, war die Anzahl der unterstützten Sprachen, da ein Chatbot für ein weltweites Unternehmen auch weltweit eingesetzt werden soll.

5.1.1. Chatbots/Chatbot-Anbieter

Online-Angebote zu Chatbots beinhalten meist Produkte, bei denen ein bereits existierendes, vortrainiertes LLM genommen wird und dann mit den eigenen Trainingsdaten zu einem Chatbot verfeinert wird, um diesen an einen bestimmten Zweck, wie zum Beispiel Kundenservice, anzupassen. Der Unterschied zum Training von LLMs ist jedoch das Level der Transparenz und Kontrolle, die man in den meisten Fällen bei Angeboten von Chatbot-Anbietern nicht auffinden kann. Anbieter solcher Chatbot-Lösungen verlangen außerdem oftmals ein Abonnement mit monatlichen oder jährlichen Zahlungen, um die ganze Funktionalität zu erhalten. So ist es zum Beispiel bei den Anbietern Jasper, Netomi, BotSonic, Bard und OpenAI (siehe Abschnitte Jasper, Netomi, BotSonic, Bard und OpenAI). Jasper, Netomi und BotSonic bieten kostenlose Demoversionen ihres Produkts, wobei man sich für diese anmelden muss. Bei Netomi und BotSonic werden zusätzlich nur Firmen-E-Mails für die Demoversion akzeptiert.

BLOOMChat v1

BLOOMChat ist ein Chatbot, welcher auf das BLOOM 176B LLM aufbaut und für Konversationen sowie Fragebeantwortung und generierendes Antworten bestimmt ist. Beim generierenden Antworten wird der Ausgabetext aktiv vom Chatbot generiert und nicht übereinstimmend aus den Trainingsdatensätzen wiedergegeben. Der Chatbot wurde von SambaNova Systems¹² entwickelt und unterstützt Konversation in einer Vielzahl von verschiedenen Sprachen, weshalb es sich auch für den Zweck eines Chatbots, welcher bei der Produktsuche helfen soll, eignen würde¹³.

BLOOMChat scheidet allerdings aus, da es beim Testen in der Web-Oberfläche eine schlechte Performanz abgeliefert hat. Darunter zählen Widersprüche und *Halluzinationen*, welche stark von den Nutzereingaben abweichen und Selbstgespräche, welche in Antworten auf andere Fragen geführt werden. Zudem hat der Chatbot Probleme, den Kontext einer Konversation zu verstehen und bei dem Kontext zu bleiben sowie sich bisherige Nachrichten zu merken. Zudem fehlt die Option, BLOOMChat zu trainieren oder lokal auszuführen. BLOOMChat basiert auf dem BLOOM-Modell, welches, trotz der schlechten Leistung von BLOOMChat, in den nächsten Schritt der Evaluierung kommt, weil eine Chatbot-Implementation eines Modells nicht immer aussagekräftig über das Modell ist.

¹² Systems, „SambaNova Systems | Generative AI, Optimized for Enterprise and Government“.

¹³ „sambanovasystems/BLOOMChat-176B-v1 · Hugging Face“.

BotSonic

BotSonic, welcher ein Service von WriteSonic ist, ermöglicht es, *Artificial-Intelligence(AI)*-Chatbots ohne eigenes Coding und nur über eigene Daten zu erstellen. Das bedeutet also, dass man dem Chatbot nur die erforderlichen Trainingsdaten übergeben muss, um einen Chatbot für einen spezifischen Anwendungszweck zu erhalten. WriteSonic verwendet dafür als Vorlage für den Chatbot das GPT-4 Modell, welches sichere, genaue Antworten liefern soll und sich deswegen für den Kundenservice eignet¹⁴.

Zudem ermöglicht es BotSonic, Konversationen, welche außerhalb des gedachten Einsatzes sind, zu blockieren, sodass Unterhaltungen mit dem Chatbot nur strikt auf den übergebenen Trainingsdaten basieren¹⁵.

BotSonic scheidet aufgrund der limitierten Gratisversion und der damit verbundenen schwierigeren Umsetzung sowie wegen der intransparenten Weiterverarbeitung von Daten aus.

¹⁴ „Custom ChatGPT - AI Chatbot for Your Website | Botsonic“.

¹⁵ „Custom ChatGPT - AI Chatbot for Your Website | Botsonic“.

Jasper

Jasper bietet eine Vielzahl von AI-Lösungen, welche in einem Unternehmen eingesetzt werden können, wobei Jasper selbst nicht die Möglichkeit bereitstellt, selbst aus Trainingsdaten einen Chatbot zu trainieren, sondern mehr auf Inhaltsgenerierung für Unternehmen angelegt ist.

Mit Jasper können Marketing-Kampagnen mithilfe eines Briefings erstellt werden, Inhalte generiert werden, welche auf den Stil und den Ton des Unternehmens abgestimmt sind oder Bilder und Illustrationen kreiert werden. Das interessanteste Angebot ist jedoch der Chatbot, welcher über eine API in eigene Plattformen und Webseiten integriert werden kann. Von Jasper werden dabei mehrere LLMs sowie aktuelle Daten von Google verwendet, um möglichst akkurat zu sein¹⁶.

Jasper scheidet aufgrund der notwendigen Funktion, Trainingsdaten zu übergeben und den Chatbot dadurch genauer anzupassen, aus, auch wenn es durch ihre Webseiten und Beschreibungen so aussieht, als ob die Funktion inkludiert wäre. Zudem müsste man ein Business-Abonnement anlegen, nur um Zugriff auf die API zu erhalten, dessen Kosten zusätzlich unbekannt sind und sich auf die Größe des anfragenden Unternehmens richten, da Jasper AI INC. in so einem Fall kontaktiert werden muss¹⁷, um den Preis zu verhandeln.

¹⁶ „Jasper | AI Copilot for Enterprise Marketing Teams“.

¹⁷ „Plans & Pricing - Jasper“.

Bard

Bard ist ein gesprächsbasierter AI-Chatbot, also ein Chatbot, welcher auf Fragen wie in einem Gespräch antwortet, welcher von Google im Jahre 2023 für die Öffentlichkeit verfügbar gemacht wurde¹⁸. Bard ist relativ neu und durchläuft dadurch konstante Veränderungen. So kam es auch zu einer Namensänderung von Bard zu Gemini¹⁹. Zudem kam es auch zu häufigeren Aktualisierungen des verwendeten LLMs, da Bard zuerst auf LaMDA, dann auf PaLM-2 und anschließend auf dem LLM Gemini aufgebaut hat²⁰.

Die Bard-API ist ein Werkzeug, welches es Entwicklern ermöglicht, aus verschiedenen Quellen auf Daten zuzugreifen und sie zu verwenden. Die Bard-API verwendet *Natural-Language-Processing* (NLP), um Informationen aus verschiedenen Arten von Dokumenten zu extrahieren und zu verstehen²¹.

Bard scheidet aufgrund von mangelnder Transparenz, was Datensicherheit und -verarbeitung angeht, aus, da das Partnerunternehmen nicht will, dass ihre Daten weiterverkauft werden oder unbefugt verwendet werden.

¹⁸ Grant und Metz, „Google Releases Bard, Its Competitor in the Race to Create A.I. Chatbots“.

¹⁹ „Home - Bard Now Gemini“.

²⁰ „What Is Google’s Gemini AI Tool (Formerly Bard)?“

²¹ „Google Bard API Documentation 2024“.

Netomi

Netomi, vormals msg.ai, ist ein AI-Unternehmen, welches es anderen Unternehmen ermöglicht, AI-Lösungen über eine No-Code-Platform bereitzustellen. Netomi eignet sich für AI-Lösungen, welche Kunden betreuen sollen und ist über Chat, E-Mail, Sprach- und Messaging-Kanäle verfügbar²².

Netomi scheidet aufgrund von mehreren Faktoren aus. Diese beinhalten zum einen die Kontaktanfrage zum Unternehmen über eine Firmen-E-Mail und die unbekannten Kosten des Services sowie die Datenverarbeitung, welche, wie bei BotSonic, nicht transparent ist, was ein konkretes Ausscheidungskriterium für das Partnerunternehmen ist. Netomi scheidet auch aus, da es nur wenige Informationen zu den vorhandenen Angeboten beinhaltet.

²² „Platform“.

OpenAI

OpenAI ist ein Unternehmen, welches sich mit der Entwicklung und Umsetzung von Künstlicher Intelligenz auseinandersetzt. Ihr Ziel ist es, sichere und nützliche *Artificial-General-Intelligence* (AGI) zu erschaffen und für die Menschheit einzusetzen²³. Dabei verwenden sie Deep Learning für ihre Modelle, in welchem ein AI-System mit einer enormen Anzahl an Daten trainiert wird, um eine Aufgabe auszuführen²⁴.

OpenAI hat dabei ChatGPT veröffentlicht, wodurch Menschen mittels einem *Chat-Interface* mit ChatGPT interagieren können. In diesem Interface können, wie bei ähnlichen Chatbots, wie zum Beispiel Bard (siehe Abschnitt Bard), Fragen über Text gestellt werden. Jedoch bietet ChatGPT auch die Möglichkeit, zusätzlich Bilder zu den Eingaben zu liefern, Bilder durch Eingaben generieren zu lassen oder die Kommunikation über Audio zu ermöglichen und selbst eigene Modelle aus Trainingsdaten zu erstellen. Jedoch sind diese Funktionen kostenpflichtig und müssen monatlich gezahlt werden, wobei diese Version von ChatGPT auf dem GPT-4-Modell basiert²⁵.

OpenAI bietet auch eine API zu ChatGPT, welche in Applikationen eingebaut werden kann, um Eingaben direkt an das Modell zu schicken. Meistens wird ein eigens trainiertes ChatGPT-Modell jedoch in Tandem mit *Cloud-Services* wie Amazon Web Services (AWS) oder Microsoft Azure, aufgrund der benötigten Infrastruktur, bereitgestellt.

ChatGPT scheidet für diese Diplomarbeit aus, weil die Umsetzung über OpenAI für ein kleines Team zu kostenaufwendig für den Auftraggeber wäre und OpenAI wenig über ihre Datenverarbeitung preisgibt. Es weist die gleichen Ausscheidungskriterien wie zuvor besprochene Chatbot-Anbieter auf.

²³ „About“.

²⁴ „Research“.

²⁵ „ChatGPT“.

5.1.2. Large Language Models

Large Language Models, abgekürzt LLMs, sind eine Kategorie von AI-Grundmodellen, welche mittels eines großen Korpus an Daten trainiert werden und genügend Grundkenntnisse besitzen, um in einer Vielzahl von Szenarien eingesetzt zu werden und Probleme zu lösen. Dies ist ein deutlicher Unterschied zum klassischen Ansatz, für jeden Anwendungsfall eine eigene künstliche Intelligenz zu erstellen, wodurch auch Kosten und Infrastruktur gespart werden²⁶.

LLMs symbolisieren einen großen Durchbruch in NLP und besonders in Bereichen der künstlichen Intelligenz. LLMs sind mittlerweile in den meisten Fällen öffentlich zugänglich. Ein gutes Beispiel stellen die GPT-3- und GPT-4-Modelle von OpenAI dar²⁷.

LLMs sind darauf ausgelegt, übergebene Texteingaben wie ein Mensch zu verstehen und schreiben zu können oder aus den Eingaben Inhalte zu erstellen, welche auf den Daten basieren, mit denen sie trainiert wurden. Sie können Schlüsse aus einem Kontext ziehen und anhand dessen relevante Inhalte und Antworten wiedergeben²⁸.

LLMs sind aufgrund von Milliarden an *Parametern* fähig, komplizierte Muster in der Sprache eines Menschen zu erkennen und ein breites Spektrum an sprachbezogenen Aufgaben zu vollziehen. Sie verändern, wie Apps in mehreren Bereichen entwickelt werden, mithilfe von Chatbots oder virtuellen Assistenten²⁹.

²⁶ „What Are Large Language Models (LLMs)?“

²⁷ „What Are Large Language Models (LLMs)?“

²⁸ „What Are Large Language Models (LLMs)?“

²⁹ „What Are Large Language Models (LLMs)?“

GPT-4 und GPT-3.5 Turbo

GPT-4 ist das neueste LLM von OpenAI, welches am 14. März 2023 erschien. Es ist ein großes multimodales Modell, was heißt, dass es Bilder und Text als Eingabe akzeptieren kann und Text zurückgibt. Das Modell ist der direkte Nachfolger des GPT-3.5 Modells und zeigt sich dadurch performanter in den meisten Anwendungsbereichen³⁰.

GPT-3.5 Turbo ist das Vorgängermodell des GPT-4 Modells und ein Nachfolger des GPT-3 Modells, welches primär die Betriebskosten und die Geschwindigkeit des GPT-3 Modells verbessern sollte und zusätzlich für den Gebrauch als Chatbot im Vergleich zu GPT-3.5, welches nicht direkt für die Öffentlichkeit bereitgestellt wurde, sondern eher in spezialisierten Abzweigungen, wie GPT-3.5 Turbo, fine-tuned wurde³¹.

Beide Modelle sind über OpenAI's API verfügbar und beide können als Basismodell in ChatGPT verwendet werden, wobei der Zugang zur API sowie der Zugang zum GPT-4-Modell in ChatGPT ein kostenpflichtiges Abonnement benötigen. Auch die Option, die Modelle für den eigenen Gebrauch zu trainieren, ist kostenpflichtig.

Als Optionen für den Chatbot scheiden beide Modelle allerdings aus, da es zurzeit keine Möglichkeit gibt, sie lokal auf dem PC laufen zu lassen und dadurch über OpenAI selbst oder Cloud-Anbieter bereitgestellt und trainiert werden müssen, wodurch die Daten intransparent während des Trainings bearbeitet werden, was, wie zuvor im Abschnitt 5.1 erwähnt wurde, ein Ausscheidungskriterium für den Auftraggeber darstellt.

³⁰ „GPT-4“.

³¹ Butler, „GPT 3.5 vs. GPT 4“.

BLOOM

Das BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) ist ein *auto-regressives LLM*, welches von BigScience veröffentlicht wurde³². Auto-regressiv bedeutet, dass vorherige Antworten des LLMs in die neue Eingabe mitgegeben werden, um einen Kontext für die nächste Antwort zu liefern³³. Dieses Modell wurde so trainiert, dass es einen Eingabetext weiterführen und beantworten kann, wobei eine klare Antwort in 46 Sprachen zurückgegeben werden kann³⁴.

BLOOM lässt sich aufgrund seiner *Open-Source*-Natur lokal ausführen und unterstützt auch das Fine-Tuning der Daten³⁵, wodurch es sich als Chatbot-Basis für das Partnerunternehmen anbietet, da die Infrastruktur lokal oder zumindest unternehmensintern über Cloud-Services bereitgestellt werden kann und die Daten auf der bereitgestellten Infrastruktur verarbeitet werden können, wodurch Transparenz entsteht. BLOOM wird deswegen weitergetestet.

³² „BigScience Research Workshop“.

³³ „Generation with LLMs“.

³⁴ „bigscience/bloom · Hugging Face“.

³⁵ „bigscience/bloom · Hugging Face“.

BERT

Bidirectional Encoder Representations from *Transformers* (BERT) ist ein Large Language Model, welches auf 104 Sprachen über Wikipedia Artikel über *Masked Language Modeling* (MLM) trainiert wurde. Es wurde nur über unstrukturierte Daten selbstüberwachend trainiert, was bedeutet, dass das LLM anhand von Fließtexten trainiert wurde, wobei das LLM selbst Eingaben und Labels zu dem Fließtext zuordnet. Das LLM wurde hauptsächlich über MLM und Next Sentence Prediction (NSP) trainiert³⁶.

Beim MLM wird ein Satz genommen, wobei das LLM 15% der erforderlichen Eingabe des Satzes zufällig maskiert und somit Wörter versteckt. Anschließend übergibt sich das LLM den teilweise versteckten Satz als Eingabe wieder, wobei es dann voraussagen muss, was die versteckten Wörter sind³⁷.

Beim NSP werden vom Modell zwei maskierte Sätze als Eingabe zusammengesetzt, wobei diese aus dem Fließtext kommenden Sätze sowohl nebeneinanderstehen können als auch getrennt im Text vorkommen können. Das Modell muss dann voraussagen, ob diese Sätze nachfolgend sind oder nicht³⁸.

BERT kann lokal ausgeführt und trainiert werden, wodurch es sich aufgrund des mehrsprachigen Aspekts geboten hat, BERT in der technischen Evaluierung weiterzuführen. Dabei wurde von den verschiedenen BERT-Modellen das Bert-Base-Multilingual-Cased-Modell verwendet, mit dem Grund, den mehrsprachigen Aspekt des Modells zu verwenden, wobei Cased hier bedeutet, dass das LLM zwischen Groß- und Kleinschreibung unterscheidet³⁹.

³⁶ „google-bert/bert-base-multilingual-cased · Hugging Face“.

³⁷ „google-bert/bert-base-multilingual-cased · Hugging Face“.

³⁸ „google-bert/bert-base-multilingual-cased · Hugging Face“.

³⁹ „google-bert/bert-base-multilingual-cased · Hugging Face“.

PaLM-2

Pathways Language Model (PaLM-2), ist ein neueres LLM, welches von Google entwickelt wurde. PaLM-2 dient als Nachfolger zu PaLM, welches bei weitem mit mehr mehrsprachigen Texten als sein Vorgänger trainiert ist und dadurch fähiger ist, komplexere Aufgaben in mehreren Sprachen zu lösen und Texte zu übersetzen. Zudem beziehen sich die Trainingsdaten von PaLM-2 zusätzlich noch auf wissenschaftliche Arbeiten und mathematikbasierte Webseiten, wodurch PaLM-2 im Vergleich zu seinem Vorgänger besser in Schlussfolgerungen und logischem Denken abschneidet⁴⁰.

PaLM-2 wurde über eine API angeboten, welche mittlerweile mit der API für das neueste Modell von Google, Gemini, welche zum Zeitpunkt der Evaluierung noch nicht bestand, ersetzt wurde. Das Prinzip, dass das Modell jedoch nicht lokal ausgeführt werden kann und wie bei den GPT-Modellen von OpenAI nur über Google selbst trainiert werden kann, bleibt dabei gleich, wobei es ein kostenloses Abonnement gibt, wobei dort die Eingaben und Antworten von Google verwendet werden, um ihre Produkte zu verbessern⁴¹.

Aufgrund dessen scheidet PaLM-2 aus, da es keine Transparenz in der Datenverarbeitung bietet und auch nicht lokal ausgeführt oder trainiert werden kann. Zusätzlich müsste ein Abonnement abgeschlossen werden, dessen Kosten sich relativ zur Verwendung der API steigern. Zusätzlich wäre die Verwendung von PaLM-2 zu unberechenbar, da jeglicher Aufwand durch eine Änderung am LLM oder Angebot von Google zurückgesetzt werden kann, wie zum Beispiel beim Umstieg des Chatbots Bard vom Modell LaMDA zu PaLM-2 und dann zu Gemini⁴² oder wie bei der Einstellung des Betriebs der PaLM-2-API für die Gemini-API⁴³.

⁴⁰ „Introducing PaLM 2“.

⁴¹ „Gemini API Pricing“.

⁴² „What Is Google’s Gemini AI Tool (Formerly Bard)?“

⁴³ „PaLM API Overview“.

LaMDA

Language-Models for Dialog Applications (LaMDA) ist ein Transformer-Modell, welches für den Dialog spezialisiert ist und 137 Milliarden Parameter besitzt. Das Modell wurde von Google veröffentlicht und ist der Vorgänger von den PaLM-Modellen und dem Gemini Modell⁴⁴.

LaMDA wurde so vorgenommen, dass es den nächsten *Token* in einem Textkorpus vorhersagen soll, wobei LaMDA dabei durch ein Dataset trainiert wurde, welches auf öffentlich verfügbare Dialoge oder Webdokumente zurückgreift.

Da LaMDA eines der älteren Google-Modelle ist, welches mit der Veröffentlichung von PaLM-2 und anschließend Gemini immer weiter eingestellt wurde, lassen sich nur erschwert Dokumentationen bezüglich der Eigenverwendung von LaMDA finden. Zudem wurde der Zugriff auf das LLM selbst für die Öffentlichkeit verweigert, aufgrund von Behauptungen eines Mitarbeiters am LLM, dass diese ein Bewusstsein erlangt hat⁴⁵ ⁴⁶, was dazu führte, dass der direkte Zugriff auf LaMDA nur über den Chatbot Bard gegeben war (siehe Abschnitt Bard

)⁴⁷.

Aufgrund der fehlenden öffentlich zugänglichen Informationen über LaMDA sowie der Einstellung des Betriebs des LLM über den Chatbot Bard und der Verweigerung des öffentlichen Zugangs des Modells selbst, gibt es keine Möglichkeit, das LLM einzusetzen. Dadurch scheidet es für die Verwendung in der Diplomarbeit aus.

⁴⁴ Thoppilan u. a., „LaMDA“.

⁴⁵ „LaMDA“.

⁴⁶ „Google Engineer Says Lamda AI System May Have Its Own Feelings“.

⁴⁷ „What Is Google LaMDA?“

Galactica

Galactica ist ein LLM von Meta, welches für wissenschaftliche Anwendungsgebiete gedacht ist. Galactica soll wissenschaftliche Fragen beantworten, mathematische Aufgaben lösen, Moleküleigenschaften vorhersagen und Protein-Annotationen vorhersagen. Das Modell wurde anhand von wissenschaftlichen Arbeiten, Referenzen und Enzyklopädien sowie Textbüchern der Chemie und Natur trainiert⁴⁸.

Galactica scheidet aufgrund von mehreren Faktoren aus. Zum einen ist das Modell nicht für den Kundenservice vorgesehen, zum anderen kann das Modell lokal laufen, allerdings kann es nicht mit eigenen Daten verfeinert auf einen spezifischen Anwendungsfall trainiert werden. Außerdem wurde Galactica für einen Vorfall kritisiert, bei dem die Demoversion von Galactica vor der Veröffentlichung falsche Daten als Fakten verbreitete und nicht zuverlässig für den vorgesehenen Anwendungsfall gehandelt hat^{49 50}.

⁴⁸ Taylor u. a., „Galactica“.

⁴⁹ „paperswithcode/galai“.

⁵⁰ „Why Meta’s Latest Large Language Model Survived Only Three Days Online“.

Falcon

Falcon ist ein generatives LLM von der Organisation Technology Innovation Institute (TII)⁵¹, welches Applikationen unterstützen soll und in vielen Anwendungsgebieten einen Nutzen finden kann. Das Ziel des TII ist es, AI öffentlich verfügbarer zu machen, sodass es Software-Entwicklern möglich ist, AI-Anwendungen und AI-Lösungen zu erstellen, ohne auf zahlungspflichtige Modelle zurückgreifen zu müssen⁵². Falcon ist daher ein Open-Source LLM, welches lokal auf der eigenen Maschine ausgeführt und für bestimmte Anwendungsgebiete weiter trainiert werden kann. Zudem ist Falcon multilingual und unterstützt mehrere Sprachen, wie zum Beispiel Spanisch, Italienisch, Schwedisch und weitere⁵³.

Falcon's Training beruht größtenteils auf dem RefinedWeb-Dataset⁵⁴, wobei dieses sich im Vergleich zu anderen Datasets in dem Aspekt unterscheidet, dass nur Webdaten genommen werden, welche so aufbereitet wurden, dass keine redundanten Daten vorhanden sind und nur qualitativ hochwertige Daten verwendet werden. Das Dataset besteht aus 5 Billionen Token, wobei die Falcon-Modelle nur 7 Milliarden, 40 Milliarden und 180 Milliarden Parameter umfassen, wobei sie mit 1 Billion, 1,5 Billion und 3,5 Billion Token aus dem Dataset trainiert wurden. Das resultiert in eine Reihe von LLMs, die sich für ihre Parameteranzahl sehr performant zeigen⁵⁵.

Die Falcon-Reihe an Modellen eignet sich als Grundlage für den Chatbot, da die Modelle vollständig offengelegt sind und auch lokal ausgeführt und trainiert werden können. Zusätzlich kann der Fakt, dass Falcon mehrere Sprachen beherrscht und zudem aufgrund seines Datensatzes eine gute Performanz aufzeigen soll, berücksichtigt werden.

⁵¹ „Technology Innovation Institute | Advanced Technology Research Institute | International Center for Technology Research | TII“.

⁵² „Falcon LLM“.

⁵³ „Falcon LLM“.

⁵⁴ Penedo u. a., „The RefinedWeb Dataset for Falcon LLM“.

⁵⁵ Almazrouei u. a., „The Falcon Series of Open Language Models“.

Llama-2

Die Llama-2 Reihe an Modellen, welche von Meta erstellt und veröffentlicht wurde und Modellgrößen von 7, 13 und 70 Milliarden Parametern beinhaltet, sind auto-regressive Open-Source-Sprachmodelle, welche eine optimierte Transformer-Architektur verwenden. Es gibt zusätzlich eine Variante vom Llama-2-Modell, welche für Chats spezialisiert wurde. Dabei können die vortrainierten LLMs für eigene Anwendungsgebiete trainiert werden⁵⁶.

Das Modell wurde anhand von öffentlich verfügbaren Daten trainiert, wobei beachtet wurde, keine persönlichen Informationen innerhalb der Daten zu inkludieren. Das Modell wurde anhand von 2 Billionen Tokens an Daten trainiert und ist in der größten Parametergröße, 70B (70 Milliarden), ähnlich performant wie Closed-Source Modelle wie GPT-4 oder Gemini (zu dem Zeitpunkt noch Bard mit PaLM-2)⁵⁷.

Llama-2 eignet sich für die Umsetzung als Chatbot, da es als Open-Source-Modell lokal ausgeführt werden kann und selbst transparent trainiert werden kann. Zudem kann die bereits für den Chat angepasste Version des LLMs mit weiteren Daten trainiert werden.

⁵⁶ Touvron u. a., „Llama 2“.

⁵⁷ Touvron u. a.

5.2. Verkleinerung der Auswahl von LLMs

Mithilfe der Vorevaluierung wurden LLMs ausgeschieden, mit denen eine Umsetzung, einen passenden Chatbot für Blum zu trainieren, nur schwer möglich wäre. Zusätzlich scheiden alle Chatbot-Anbieter aufgrund der fehlenden Trainingsmöglichkeiten oder Transparenz in der Datenverarbeitung aus. Zu den ausgeschiedenen LLMs zählen dabei die GPT-Modelle aufgrund ihres Abonnement-Modells sowie der intransparenten Datenverarbeitung beim Training und PaLM-2 aufgrund der Veraltung des Modells und der damit verbundenen Schwierigkeit, eine passende API und Dokumentation zu finden, aus. LaMDA gilt als Vorgänger von PaLM-2, welcher zusätzlich zu den Problemen, die PaLM-2 aufweist, auch häufig halluziniert. Allerdings kann LaMDA weder getestet noch verwendet werden, da das Modell zum einen aufgrund seiner veralteten Art nicht mehr über Bard zugänglich ist und zum anderen als eigenes Modell nie zugänglich war. Galactica scheidet auch aus, da es nicht für den gegebenen Anwendungsfall geeignet ist und in der Demoversion falsche Antworten zurückgab.

5.3. Evaluierung der übrigen Modelle

Aufgrund der Vorevaluierung und der Ausscheidung anderer Angebote bleiben nur noch 4 LLMs übrig, welche näher in einer virtuellen Umgebung getestet werden sollen. Diese 4 Modelle sind:

- BLOOM-7b
- Falcon-7b
- Llama-2-7b
- BERT-Base-Multilingual-Cased

Hierbei hat sich jedoch herausgestellt, dass BERT-Base-Multilingual-Cased keine Textgenerierung unterstützt, sondern nur Eingaben zusammenfassen kann. Darum scheidet BERT schon frühzeitig aus, ohne in der VM ausführlich getestet werden zu können.

Für die Evaluierung wurden diese Modelle auf der virtuellen Maschine gespeichert, wobei jedes Modell anhand von 8 Fragen auf bestimmte Kriterien getestet wurde, auf welche später eingegangen wird. Während der Verarbeitung der Fragen wird die Antwortzeit der LLMs als zusätzliches Kriterium gemessen. Dabei wurden die folgenden Zeiten erfasst:

	Antwortzeiten		
Frage	BLOOM 7b	Falcon 7b	Llama2-7b
1	01:27.1	00:15.3	00:04.2
2	01:31.3	00:06.4	00:04.4
3	07:03.4	00:13.6	00:08.5
4	00:13.8	00:03.3	00:09.4
5	06:54.8	00:02.6	00:06.6
6	02:40.6	00:04.4	00:20.8
7	07:33.1	00:06.8	00:12.2
8	02:14.9	00:13.3	00:08.8
Durschn.	03:42.4	00:08.2	00:09.4
Median	02:27.8	00:06.6	00:08.6

Abbildung 15: Erfasste Antwortzeiten

Der Median dient hierbei als Absicherung gegen Ausreißer, wobei in der nachherigen Bewertung die Punktzahl trotz der Verwendung des Medians gleichbleibt.

Die Hochfahrzeiten für die LLMs wurden auch erfasst, jedoch nur einmal am Anfang des jeweiligen Testvorgangs. Die Tabelle der Bootzeiten sieht wie folgt aus:

Bootzeiten		
BLOOM 7b	Falcon 7b	Llama2-7b
00:47.8	02:31.4	00:04.2

Abbildung 16: Erfasste Bootzeiten

5.3.1. Kriterien für die Evaluierung

Für die Evaluierung der restlichen Modelle wurde nach mehreren relevanten Kriterien gesucht, von denen dann nur die wichtigsten genommen wurden, welche hauptsächlich zur Bestimmung der allgemeinen Performanz dienten. Alle Kriterien sind wie folgt aufgelistet:

- **Boot-Zeit:**

Die Boot-Zeit beschreibt die Zeit, welche das LLM zum Starten benötigt, bevor es eine Eingabe erstmalig in einer Session verarbeiten kann.

- **Sprachverständnis**

Die Fähigkeit des LLMs, menschlichen Text zu verstehen und für die Spracherzeugung zu verwenden. Hierbei wird bewertet, ob das Modell in der Lage ist, Texte zu verstehen und Zusammenhänge zu erkennen.

- **Spracherzeugung**

Die Fähigkeit des Modells, menschenähnlichen Text zu generieren. Hierbei wird bewertet, wie gut das LLM verständliche Texte generieren kann.

- **Antwortrelevanz**

Die Antwortrelevanz wird anhand der Fähigkeit des LLM bewertet, auf gestellte Fragen angemessen zu antworten und relevante Informationen bereitzustellen.

- **Bias**

Wie sehr das LLM zu einer bestimmten ethischen Handlungsweise tendiert. Hierbei wird bewertet, ob sich das LLM neutral zu bestimmten Thematiken verhält, wie beispielsweise bei politischen Angelegenheiten oder ethischen Fragen.

- **Performanz**

Wie schnell das LLM auf gegebene Anfragen antwortet, wie gut die Qualität der Antwort ist (faktuelle Korrektheit) und wie ausführlich die Antwort ist (Länge der Antwort). Es werden hierbei die Faktoren Antwortgeschwindigkeit, Qualität und Wahrheitsgehalt bewertet.

- **Ressourceneffizienz**

Die Effizienz von LLMs in Bezug auf Rechenleistung, Speicherbedarf und Energieverbrauch zum Betreiben des LLMs wird bewertet.

- **Skalierbarkeit**

Wie leicht das LLM sich auf mehrere Parameter bzw. Trainingsdaten im Nachhinein erweitern lässt.

- **Trainingszeit**

Wie lange es braucht, das Modell vollständig zu trainieren, in Abhängigkeit von der Länge der Trainingsdaten.

- **Allgemeine Leistung**

Wie gut das LLM auf Anfragen reagiert, welche keinen Zusammenhang zu jeglichen Trainingsdaten haben. Bewertet wird die Ausgabe des LLM.

- **Ethik**

Anhand welcher Moralen das LLM Antworten gibt und ob das LLM auf anstößige Anfragen antwortet. Bewertet wird die Antwort des LLM.

- **Freundlichkeit**

Die Fähigkeit des Modells, auf natürliche und hilfsbereite Art und Weise mit Benutzern zu interagieren und die Erwartungen der Benutzer zu erfüllen. Hier wird bewertet, wie freundlich das LLM auf Fragen antwortet.

- **Benutzerfreundlichkeit**

Wie leicht das LLM von einem Nutzer ohne Vorkenntnisse verwendet werden kann. Hier wird bewertet, wie leicht man eine Anfrage stellen kann und wie leicht man Antworten (falls überhaupt) konfigurieren kann.

Dies sind generelle Kriterien, die bei der Evaluierung von LLMs erfasst werden können, wobei nur bestimmte Kriterien bei dieser Diplomarbeit in Betracht gezogen wurden.

5.3.2. Relevante Kriterien

Die wichtigsten Kriterien für die Evaluierung sind:

- Sprachverständnis
- Performanz (Antwortzeit, Antwortlänge, Antwortkorrektheit)
- Sprachzeugung
- Relevanz
- Boot-Zeit

Diese Kriterien waren praktisch mit den gegebenen Methoden am messbarsten und relevantesten für die Entscheidung. Aus zeitlichen und kostenbezogenen Gründen war es bei den meisten Kriterien nicht möglich, sie zu bewerten, da sie einen größeren Datensatz benötigt hätten, der antrainiert werden müsste.

5.3.3. Technische Umsetzung der Evaluierung

Es gibt grundsätzlich mehrere Arten, wie man LLMs richtig evaluiert (siehe Abschnitt 5), allerdings bestehen in diesem Fall weder die Zeit noch die liquiden Mittel, um eine vollständige Evaluierung durchzuführen, wie sie üblicherweise gemacht wird. In diesem Fall wurde eine menschliche Evaluierung mithilfe von 8 Fragen durchgeführt, wobei 4 davon auf Englisch und 4 davon auf Deutsch gestellt wurden. Diese Fragen wurden jedem LLM auf dieselbe Weise gestellt und anschließend wurden die jeweiligen Antworten zu den Fragen bewertet. Jede Frage ist dazu gedacht, eine oder mehrere Kriterien der Kriterien zu testen, die als wichtig erachtet wurden, jeweils in beiden Sprachen. Die Fragen sind größtenteils eigenhändig erstellt worden, wobei manche auf dem SQuAD-2-Evaluierungs- bzw. Trainingsdatensatz basieren und wie folgt lauten⁵⁸:

Englisch

- **The atomic number of the periodic table for oxygen?**

Mit dieser Frage wurde beabsichtigt, eine faktual korrekte Antwort, welche auch Sinn im gegebenen Kontext macht, zu erhalten. Diese Frage wurde aus dem SQuAD-Datensatz entnommen⁵⁹.

- **Explain the impact of World War 2 on Europe and its consequences on the economy.**

Diese Frage diente dazu, abzuschätzen, wie das LLM relevant zur Frage einen sprachlich korrekten Text wiedergeben kann, der, wenn länger geschrieben wird, weiterhin relevant bleibt.

- **Make clear why protecting the climate is important in your own words.**

Hier werden die gleichen Kriterien wie in der vorigen Frage beachtet.

- **Explain the history of the combustible-powered automobile, in German language.**

Hier werden das Sprachverständnis sowie die Sprachzeugung des LLMs gefordert, da die Antwort in einer anderen Sprache erwartet wird als in der, in welcher die Frage gestellt wird. Die Relevanz zur gestellten Frage ist auch hier ein wichtiges Kriterium, da überprüft werden kann, ob das LLM in einer anderen Sprache weiterhin relevant bleibt.

⁵⁸ „The Stanford Question Answering Dataset“.

⁵⁹ „SQuAD - the Stanford Question Answering Dataset“.

Deutsch

Bei den deutschen Fragen handelt es sich fast immer um die Sprachzeugung in der deutschen Sprache sowie die Antwortlänge und ob das LLM nach einigen Sätzen wieder ins Englische zurückfällt.

- **Wie denkst du wird sich die Technologie in der Zukunft entwickeln?**

Hier muss die LLM selbst denken und es wird beobachtet, ob die erhaltene Antwort relevant für die Frage ist.
- **Vergleiche Flugzeuge und Automobile. Welche Zusammenhänge erkennst du in ihrer Geschichte?**

Diese anspruchsvolle Frage wird zum Testen des Sprachverständnisses und der faktuellen Korrektheit, sowie Antwortrelevanz verwendet.
- **Wie würdest du bei der Ideenfindung einer Innovation vorgehen? Beschreibe deine ersten drei Schritte.**

Auch hier gilt neben der Sprachzeugung und der Antwortlänge die Relevanz als zusätzlicher Faktor.
- **Gib mir eine Zusammenfassung des Buches „Löcher“ auf Englisch.**

Da diese Frage auf einem Roman⁶⁰ basiert, welches den LLMs, welche evaluiert werden, nicht bekannt ist und danach noch auf Englisch gefragt wird, wird mit dem LLM auf ihre Grenzen gestoßen und somit überprüft, wie das LLM auf Anfragen reagiert, mit denen es prinzipiell nichts anfangen kann. Im Grunde wird hier keine ernsthafte Antwort von der LLM erwartet. Es geht mehr darum, wie es aussieht, wenn die LLM mit Fragen konfrontiert wird, die weit über ihrer Kompetenz liegen.

⁶⁰ „„Löcher“ von Louis Sachar“ - Buch - ,978-3-407-74098-4“.

5.4. Entscheidung des LLMs

Die Entscheidung des LLMs wurde mithilfe einer Scoring-Analyse gemacht, wobei die Kriterien eine Punktzahl von 1-5 erhalten, wobei eine Punktzahl von Eins die schlechteste Bewertung und eine von Fünf die beste Bewertung ist. Die Wertung erfolgt hierbei durch die Bildung des Durchschnitts einer Kategorie. Es werden also alle Punkte der jeweiligen Kategorie addiert und dann durch die vorkommende Häufigkeit der Kategorie dividiert. Anschließend werden die einzelnen Kategorien nach Prozenten gewichtet und mit den Durchschnittspunkten der Kategorie multipliziert. Die Fragen dienen hierbei also nur zur Bestimmung und Bewertung der einzelnen Kriterien, welche stark in Erwägung gezogen werden sollen⁶¹.

Die folgende Abbildung zeigt die Scoring-Analyse anhand der durchschnittlich erzielten Punkte in den jeweiligen Kriterien und einer zugehörigen Gewichtung der Kriterien. Hierbei ist das Llama2-7b sichtlich als bestes LLM nach den festgelegten Kriterien bewertet, deutlich über Falcon und BLOOM. Die Kriterien Spracherzeugung und Sprachverständnis berücksichtigen beide jeweils die Deutsche sowie die Englische Sprache und werden nicht getrennt gewichtet.

	K.O. Kriterien	durchschn. Antwortgeschwindigkeit	durschn. Bootzeit	durchschn. Korrektheit	durchsch. Sprachzeugung	durschn. Antwortlänge	durchschn. Sprachverständnis	durchschn. Relevanz
BLOOM 7b		1	4	1.33	2.57	3.5	1.666666667	1.75
Falcon 7b		5	1	2.00	2.57	2.8	2	2.5
Llama2-7b		5	5	5.00	3.71	3.6	4.666666667	4.5
BERT	kann keine Texte generieren sondern nur zusammenfassen							
Gewichtung	100%		10%	5%	25%	10%	10%	20%
BLOOM 7b		0.1	0.2	0.33333333	0.257142857	0.35	0.333333333	0.35
Falcon 7b		0.5	0.05	0.5	0.257142857	0.28	0.4	0.5
Llama2-7b		0.5	0.25	1.25	0.371428571	0.36	0.933333333	0.9

Abbildung 17: Scoring-Analyse der LLMs

⁶¹ „▷ Nutzwertanalyse » Definition, Erklärung & Beispiele + Übungsfragen“.

Dabei wurden die Kriterien Antwortzeit und Bootzeit mit Grenzen anhand ihrer Werte bewertet.

Gewichtung Bootzeiten	
> 2	1
1:59.9 - 1:30	2
1:29.9 - 1:00	3
0:59.9 - 0:30	4
< 0:30	5

Abbildung 18: Gewichtung der Bootzeiten

Gewichtung Antwortzeiten	
> 2:00.0	1
1:59.9 - 1:30	2
1:29.9 - 1:00	3
0:59.9 - 0:30	4
< 0:30	5

Abbildung 19: Gewichtung der Antwortzeiten

Die Gewichtung der Kriterien kann wie folgt begründet werden:

- **Antwortzeit:**

Dieses Kriterium wird mit 10% gewichtet, da in den meisten Fällen die Ausgabezeit abhängig von der Länge ist und somit dementsprechend länger dauern kann. Allerdings ist das Kriterium nicht unwichtig, da BLOOM in den meisten Fällen für die gleiche Anzahl an Tokens (die Wortlänge der Ausgabe) deutlich länger als Falcon oder Llama gebraucht hat, weshalb es auch mit einer Eins bewertet ist.

- **Bootzeit:**

Dieses Kriterium wurde mit 5% gewichtet, da es für den Betrieb der LLM keine gravierende Rolle spielt, wie lange das Modell starten muss. Das Kriterium behält jedoch an Relevanz in Testumgebungen oder Deployments, wo der Chatbot eventuell mehrmals neugestartet werden muss und es zeitsparender ist, wenn der Chatbot schnell hochfahren kann.

- **Korrektheit:**

Dieses Kriterium wird als wichtigstes erachtet und darum beträgt die Gewichtung auch 25%, da ein Chatbot im Endeffekt so faktual korrekt wie möglich sein soll und nicht Antworten erfinden soll.

- **Sprachzeugung:**

Dieses Kriterium erhält eine Gewichtung von 10%. Es hat keine so gravierende Bedeutung wie die Korrektheit oder Relevanz, da LLMs in der Regel in der Lage sein sollten, menschlichen Text zu generieren. Jedoch bewertet dieses Kriterium, ob der Text, welcher von der LLM zurückgegeben wird, kontextbasiert verständlich ist und nicht nur eine zufällige Wiedergabe von Wörtern aus dessen Parametern darstellt.

- **Antwortlänge:**

Auch die Antwortlänge wird mit 10% gewichtet, da in der Regel die Antwortlänge von der Eingabe abhängig ist und davon ob diese eine kurze Antwort (siehe Frage „The atomic number of the periodic table for oxygen?“) oder eine lange Antwort erwartet (siehe Frage „Explain the impact of World War 2 on Europe and its consequences on the economy.“). Außerdem lässt sich die Antwortlänge nicht pauschalisieren, da selbst kurze Antworten sehr aussagekräftig sein können. Jedoch ist die Antwortlänge dennoch wichtig, da abgeschätzt werden kann, ob ein Chatbot sich auf das Notwendigste fokussieren kann oder umgekehrt bei langen Fragen, weit ausholen kann, wenn es gefordert wird.

- **Sprachverständnis:**

Das Sprachverständnis wird mit 20% gewichtet, weil es wichtig ist, dass das LLM den Kontext der Eingabe versteht und darauf auch eine passende Antwort liefern kann. Durch die deutschen Fragen wird das LLM in diesem Bereich besonders gefordert, da ihr Verständnis in einer anderen Sprache getestet wird, auch wenn auf Englisch geantwortet wird.

- **Relevanz:**

Dieses Kriterium wird ebenfalls mit 20% gewichtet, mit der Begründung, dass eine LLM stets relevante Antworten zu Nutzereingaben liefern sollte, sofern die Antwort keine anstößigen Inhalte enthält.

5.5. Evaluierung - Fazit

Um einen geeigneten Chatbot zu finden, müssen verschiedene Angebote in Erwägung gezogen werden. Zu diesen Angeboten zählen Chatbot-Anbieter und Large-Language-Models. Die verfügbaren Angebote müssen, aufgrund ihrer Vielfältigkeit und ihrer Anzahl, anhand der für das Endprodukt gewünschten Kriterien, evaluiert werden. Für die Evaluierung stehen dabei mehrere Methoden und Frameworks zur Verfügung. Als geeignete Methode bietet sich das menschliche Evaluieren durch eigene Fragen, aufgrund der ansonsten anfallenden Kosten, an. Um die Anzahl der zu evaluierenden Modelle zu reduzieren, müssen alle Angebote gründlich angeschaut werden, damit ungeeignete Angebote herausgefiltert werden und nur relevante Modelle technisch evaluiert werden.

Die technische Evaluierung wurde mithilfe einer Scoring-Analyse gemacht, wobei die wichtigsten Kriterien für den Chatbot erfasst wurden und dementsprechend gewichtet wurden. Durch die menschliche Evaluierung von drei ausgewählten LLMs wurde festgestellt, dass zwischen BLOOM, Falcon und Llama-2, Llama-2 die akkuratesten und fehlerfreisten Antworten liefern konnte, welche sprachlich auch verständlich waren. Llama-2 ist auch in der Lage, deutschen Text zu verstehen und richtig interpretieren zu können, eine Kompetenz, welche bei BLOOM und Falcon in der Evaluierung gefehlt hat.

6. Data-Cleaning

Für diese Arbeit hat die Firma Blum Trainingsdaten in Form von JavaScript Object Notation (*JSON*)-Dateien bereitgestellt, welche den Inhalt des *Content-Management-Systems* von blum.com darstellen. Allerdings entsprachen die Struktur und der Inhalt nicht den Anforderungen, die für das Training eines Chatbots gebraucht werden. Die Daten enthielten zahlreiche Informationen, die für die Anwendung eines Chatbots irrelevant sind, wie beispielsweise Navigationslinks, Werbeinhalte und mehr. Aus diesen Gründen war es notwendig ein Data-Cleaning durchzuführen.

JSON-Files

Die JavaScript Object Notation (*JSON*) ist ein Datenformat, welches eine einfache und lesbare Form hat. Sie wird für den Datenaustausch zwischen verschiedenen Anwendungen genutzt. Ebenso ist sie programmiersprachenunabhängig. Ein *JSON*-Objekt besteht immer aus einem Schlüssel (*key*) und einem Wert (*value*), wie in Abbildung 20 dargestellt. Der Key ist immer ein String, während der Value entweder eine Zahl, ein String, ein Boolean, ein *Array*, ein anderes *JSON*-Objekt oder null sein kann.⁶²

```
{   key      value
  "Herausgeber": "Xema",
  "Nummer": "1234-5678-9012-3456",
  "Deckung": 2e+6,
  "Währung": "EURO",
  "Inhaber": {
    "Name": "Mustermann",
    "Vorname": "Max",
    "maennlich": true,
    "Hobbys": ["Reiten", "Golfen", "Lesen"],
    "Alter": 42,
    "Kinder": [],
    "Partner": null
  }
}
```

Abbildung 20: *JSON*-Datei⁶³

⁶² HTL-Unterrichtsfach_BDAS „4_NoSQL_Document_storage_v1.4.pdf“ Folie_6-7, 14.03.2024.

⁶³ HTL-Unterrichtsfach_BDAS „4_NoSQL_Document_storage_v1.4.pdf“ Folie_6-7, 14.03.2024.

Data-Cleaning

Einer der wichtigsten Vorgänge in der Datenanalyse und Verarbeitung ist die Datenbereinigung. Diese wird auch Data-Cleaning genannt. Beim Data-Cleaning handelt es sich um das Erkennen und Beheben von fehlerhaften, unvollständigen, irrelevanten, inkonsistenten und falschen Daten. Mit Hilfe einer Reihe an Prozessen sollen alle Datensätze fehlerfrei gemacht und die Datenqualität verbessert werden.⁶⁴

Um glaubwürdige Ergebnisse und sinnhafte Antworten von einem Chatbot zu bekommen, ist es erforderlich, *saubere Daten* zu haben, da fehlende oder unreine Daten zu ineffizienten Prozessen, falschen Antworten und erheblichen finanziellen Verlusten führen. Denn dann wäre das ganze Training umsonst. Am Ende einer sorgfältig durchgeföhrten Datenbereinigung sollten alle Datensätze fehlerfrei sein. In dieser Arbeit erfolgte ein gründliches Data-Cleaning der zur Verfügung gestellten Trainingsdaten, um so die Qualität der Daten zu erhöhen.⁶⁵

⁶⁴ „Data Cleaning: Definition, Techniken, Bedeutung in der Data Science“.

⁶⁵ „Data Cleaning: Definition, Techniken, Bedeutung in der Data Science“.

6.1. Bedeutung des Data-Cleaning für einen Chatbot

Saubere Daten sind für die Entwicklung von Künstliche Intelligenz (KI)-Systemen insbesondere Chatbots sehr wichtig. Damit ein Chatbot eine präzise Analyse von Benutzeranfragen durchführen kann und ein tiefes Verständnis für Kontext, Absicht und die Feinheiten menschlicher Kommunikation hat, benötigt dieser saubere Daten. Dafür ist ein gründliches Data-Cleaning entscheidend.⁶⁶

Saubere Daten stärken das Vertrauen in den Chatbot, da dieser zuverlässig reagieren kann. Zudem sind saubere Daten notwendig für die Skalierbarkeit und Weiterentwicklung des Chatbots. Der Chatbot kann dank der sauberen Daten weiterlernen und sich weiterentwickeln, ohne dass Fehler aus den Trainingsdaten sich anhäufen und ausbreiten.⁶⁷

Die Datenbereinigung ist ein fortlaufender Prozess bei der Entwicklung des Chatbots, der die Grundlage für den Chatbot legt. Dabei werden qualitative Datensätze gesammelt, die der Chatbot im Laufe des Trainings verarbeiten sollte. Um die Qualität des Chatbots zu maximieren, spielt die Reinigung der Daten eine wichtige Rolle. Wenn ein Datensatz viele Inkonsistenzen enthält, dann wird der Lernprozess des Chatbots behindert. In solch einem Fall kann man mit einigen unerwünschten Antworten rechnen.⁶⁸

Diese verschiedenen Maßnahmen stellen sicher, dass der Chatbot einen umfangreichen und vielfältigen Datensatz erhält. Ebenso wird gewährleistet, dass alle Nutzer gleich behandelt werden und dass der Chatbot ein tiefes Verständnis von Sprachmustern entwickelt. Außerdem werden die Genauigkeit, Relevanz und Robustheit der Daten erhöht. Die Datenbereinigung erfordert Programmiersprachen wie Python, um den Data-Cleaning-Prozess zu vereinfachen.⁶⁹

⁶⁶ „The Essential Role of Data Cleaning in Chatbot Training“, 22. November 2023.

⁶⁷ „The Essential Role of Data Cleaning in Chatbot Training“.

⁶⁸ „The Essential Role of Data Cleaning in Chatbot Training“.

⁶⁹ „The Essential Role of Data Cleaning in Chatbot Training“.

6.2. Programmiersprache

Was ist eine Programmiersprache?

Eine Programmiersprache besteht aus einer Abfolge von Befehlen, die der Computer analysiert, versteht und dann ausführt, um eine spezifische Aufgabe zu lösen beziehungsweise zu bearbeiten. In der Regel schreibt ein Programmierer diese Anweisungen. Für jemanden, der nicht mit der *Syntax* der Programmiersprache vertraut ist, sind diese Anweisungen unverständlich.⁷⁰

Programmiersprachen werden anhand ihrer unterstützten Merkmale und Programmierstile in verschiedene Kategorien unterteilt. Einige Beispiele sind:

- *objektorientiertes Programmieren*
- *logisches Programmieren*
- *prozedurales Programmieren*
- *funktionales Programmieren*

Für unterschiedliche Anwendungsfälle sind verschiedene Programmiersprachen geeignet. Somit können sie für verschiedene Arten der Softwareentwicklung verwendet werden.⁷¹

Auch dazu einige Beispiele:

- Webanwendungen
- mobile Anwendungen
- Desktop-Software
- Spiele
- Data-Cleaning
- Systemprogrammierung

Python, Java, C#, JavaScript, Ruby und C++ gehören zu den bekanntesten Programmiersprachen. Jede Sprache hat ihre eigenen Vor- und Nachteile.

⁷⁰ Tuama, „What Is A Programming Language?“

⁷¹ Tuama.

Eine Programmiersprache wird anhand dieser Vorteile beziehungsweise Nachteile und anhand der Anforderungen des Projekts ausgewählt. Leistungsanforderungen, Plattformkompatibilität, Entwicklerkompetenz und Unterstützung durch die Community, sind Beispiele für solche Anforderungen.⁷²

Low-Level-Programmiersprachen und High-Level-Programmiersprachen

Es wird zwischen zwei Arten von Programmiersprachen unterschieden, nämlich Low-Level und High-Level Programmiersprachen. Diese basieren auf dem Verständnisniveau des Menschen.⁷³

High-Level Programmiersprachen

High-Level Programmiersprachen sind im Vergleich zu Low-Level Programmiersprachen menschenfreundlicher. Das bedeutet, dass der Code, aufgrund weniger Komplexität, einfacher zu interpretieren ist. Unter anderem erleichtert sich dadurch auch die Fehlersuche. Aber da High-Level Programmiersprachen weiter entfernt der Hardware liegen, sind sie weniger speichereffizient und oft langsamer. Ein Vorteil, der sich dadurch ergibt, ist die Kompatibilität, da der Code nahezu auf jeder Plattform bearbeitet und ausgeführt werden kann, da dieser plattformübergreifend ist.⁷⁴

Ein paar Beispiele für High-Level Programmiersprachen sind:

- Python
- JavaScript
- R

Low-Level Programmiersprachen

Low-Level Programmiersprachen sind mehr maschinenfreundlich und weniger menschenfreundlich. Das bedeutet, dass der Code komplex ist und daher eher schwer zu interpretieren. Die Fehlersuche bei Low-Level Programmiersprachen ist mühsamer, aufgrund dieser Komplexität des Codes. Im Gegensatz zu High-Level Programmiersprachen sind sie jedoch speichereffizienter und näher an der Computer-Hardware. Deswegen ist diese Art

⁷² Tuama.

⁷³ Tuama.

⁷⁴ Tuama.

von Programmiersprache effizienter und schneller im Vergleich zu den High-Level Programmiersprachen. Ein Nachteil der Low-Level Programmiersprachen ist die fehlende Kompatibilität, da oft der Code maschinenabhängig ist und somit nur auf einer bestimmten Plattform ausgeführt werden kann.⁷⁵

Ein paar Beispiele für Low-level Programmiersprachen sind:

- C
- C++
- RUST⁷⁶

⁷⁵ Tuama.

⁷⁶ Saxena, „Low Level Programming Language Examples“.

Fazit

Aufgrund ihrer Verständlichkeit und Plattformunabhängigkeit werden High-Level Programmiersprachen bei der Entwicklung von Software bevorzugt und somit öfter eingesetzt. Low-Level-Programmiersprachen hingegen werden zwar immer noch verwendet, sind aber seltener. Mithilfe der oben genannten Gründe wurde bei dieser Diplomarbeit entschieden eine High-Level-Programmiersprache zu verwenden.

Evaluierung Programmiersprachen für Data-Cleaning

Bei der Evaluierung standen 3 Programmiersprachen zur Auswahl, da diese sich am besten für das Data-Cleaning eignen. Diese sind Python, R und Structured Query Language (SQL).⁷⁷

Python

Python ist eine dynamisch typisierte und objektorientierte High-Level-Programmiersprache, welche sich durch ihre einfache Syntax, welche nah an der englischen Sprache liegt und durch die Menge an *Bibliotheken*, die sie bietet und die oft viel Arbeit sparen, in den letzten Jahren an großer Beliebtheit gewonnen. Sie eignet sich für die schnelle Anwendungsentwicklung, kann aber auch als Skript- oder Interpretiersprache verwendet werden. Durch ihre einfache Syntax ist es einfacher den Code zu verstehen, was die Kosten für die Fehlerfindung reduziert. Einer der größten Stärken von Python sind seine Module und Pakete, welche einfach im Code integrierbar sind und eine fertige Lösung für verschiedene Probleme bieten. Der Python-Interpreter und die umfassende Standardbibliothek sind kostenlos und frei verfügbar für alle wichtigen Plattformen. Durch die Menge an Bibliotheken und Tools, welche in Python verfügbar sind, ist Python eine gute Entscheidung, um das Data-Cleaning auszuführen. Unter diese Bibliotheken fallen beispielsweise Pandas, Numpy und Matplotlib, welche das Data-Cleaning mit ihren eingebauten *Funktionen* und Datentypen erleichtern. Zu diesen Funktionen zählt auch die Visualisierung von Daten, die oft Ausreißer oder andere Abhängigkeiten ersichtlich machen. Zusätzlich existieren zahlreiche andere Tools, welche häufig vorkommende Anforderungen des Datenbereinigens wie das Filtern, Sortieren und Transformieren von Daten erleichtern.⁷⁸

⁷⁷ TheKnowledgeAcademy, „What Is Data Cleaning?“

⁷⁸ „What Is Python?“

R

R ist eine Open-Source-Programmiersprache. Sie ist als Statistiksoftware und Datenanalysewerkzeug weit verbreitet. Im Allgemeinen wird die Programmiersprache mit einer Befehlszeilenschnittstelle geliefert. Sie ist auf weit verbreiteten Plattformen wie Windows, Linux und macOS verfügbar.⁷⁹

In den Bereichen maschinelles Lernen, Statistik und Datenanalyse gilt R oft als führend. Die Sprache ist plattformunabhängig, sodass sie auf allen Betriebssystemen genutzt werden kann. Es handelt sich bei R um eine Open-Source-Sprache. Sie kann von jeder Organisation ohne Lizenz erworben werden.⁸⁰

R ermöglicht außer der statistischen Analyse auch die Integration von anderen Sprachen wie C und C++. Die Interaktion mit einer Vielzahl von Datenquellen und Statistikpaketen ist mit R problemlos möglich. R wird als das umfassendste Paket für statistische Analysen angesehen.⁸¹

Neue Technologien und Konzepte werden oft zuerst in R eingeführt, wodurch R plattformübergreifend und jederzeit ausführbar ist. Einige Pakete haben jedoch den Nachteil, dass ihr Standard nicht perfekt ist und die R-Befehle wenig Einfluss auf die Speicherverwaltung haben. Es kann schwer sein Fehler zu finden. Daher sollte der Code in R möglichst korrekt und fehlerfrei sein.⁸²

Verglichen mit anderen Programmiersprachen wie Python ist die Programmiersprache R langsamer, da sie mehr Speicher verbraucht.⁸³

⁷⁹ „R Programming Language - Introduction“.

⁸⁰ „R Programming Language - Introduction“.

⁸¹ „R Programming Language - Introduction“.

⁸² „R Programming Language - Introduction“.

⁸³ „R Programming Language - Introduction“.

SQL

SQL bedeutet Structured Query Language. Es ist eine Programmiersprache, um relationale Datenbanken zu verwalten und zu bearbeiten. SQL ermöglicht effizientes Abfragen, Filtern, Aktualisieren und Löschen. Auch Datenbereinigungsaufgaben lassen sich damit durchführen. Sie bietet einen Ansatz, mit dem die gewünschten Daten spezifiziert werden können. Die Datenbank-Engine regelt dann die Details des Abrufs oder der Änderung dieser Daten. Ebenso zeichnet sich SQL durch seine Fähigkeit aus, große Datenmengen schnell und effizient zu verarbeiten. Aufgrund der oben aufgezählten Gründe ist SQL ein Werkzeug, das sich besonders gut für das Data-Cleaning eignet.⁸⁴

⁸⁴ „SQL Data Cleaning Techniques for Accurate Analytics | Airbyte“.

Entscheidung der zu verwendenden Programmiersprache

Nach sorgfältigem Analysieren der oben beschriebenen Programmiersprachen, welche ihre Stärken und Anwendungsbereiche im Kontext des Data-Cleanings haben, wurde Python gewählt. Die Gründe, die zu dieser Entscheidung geführt haben, werden in den nächsten Absätzen aufgezählt.

Python ist sehr vielseitig und benutzerfreundlich. Außerdem hat Python eine einfache Syntax die Produktivität, was besonders vorteilhaft für die Entwicklung von Data-Cleaning-Prozessen ist. Durch die einfache Lesbarkeit des Codes reduziert sich die Einarbeitungszeit für neue Entwickler. Ebenso minimiert sich der Aufwand für die Wartung und Aktualisierung bestehender Skripte.

Python hat weitere Vorteile, wie das umfangreiche Ökosystem an spezialisierten Bibliotheken und Tools für Data Science und Data-Cleaning. Für die Datenbereinigung und -analyse können verschiedene Bibliotheken genutzt werden. Einige Beispiele sind:

- Pandas für Datenmanipulation
- NumPy für numerische Berechnungen
- Matplotlib & Seaborn für die Datenvizualisierung
- Scikit-learn für maschinelles Lernen.

Diese Bibliotheken bieten Funktionen an, welche direkt auf die Bedürfnisse des Data-Cleaning zugeschnitten sind. Sie erleichtern komplexe Aufgaben wie das Filtern, Sortieren, Umwandeln und Darstellen von Daten und verbessern effiziente Arbeitsabläufe.

Zusammenfassend kann man sagen, dass Python, dank ihres umfangreichen Angebots an Bibliotheken und Tools, der einfachen Handhabung, Skalierbarkeit und Vielseitigkeit, eine gute Wahl für das Data-Cleaning ist. Data Scientists können mithilfe von Python unterschiedliche Aufgaben zur Bereinigung von Daten ausführen, um die Daten schnell und effizient für kommende Aufgaben vorzubereiten.

6.3. Libraries

Die folgenden Python-Libraries wurden in diesem Projekt verwendet, um die Data-Cleaning-Aufgaben durchzuführen:

os: Diese Standardbibliothek wird genutzt, um mit dem Betriebssystem zu interagieren. Mit dieser Bibliothek ist es möglich, Dateipfade zu manipulieren, neue Verzeichnisse zu erstellen und den Inhalt von Verzeichnissen darzustellen. Die Verwendung von os ist unerlässlich, um die plattformübergreifende Kompatibilität sicherzustellen und um die Kommunikation mit den Daten auf Dateisystemebene zu ermöglichen.⁸⁵

pandas: Pandas baut auf Python auf und ist eine Open-Source-Softwarebibliothek. Diese Library wurde speziell für die Datenmanipulation und -analyse entwickelt. Eine Vielzahl von Datenstrukturen und -operationen, welche eine leistungsstarke, flexible und benutzerfreundliche Datenanalyse und -manipulation ermöglichen, wird von dieser Bibliothek angeboten. Pandas trägt zur Stärkung von Python bei. Dies tut es, indem es der Programmiersprache die Fähigkeit verleiht, mit Daten, ähnlich einer Tabellenkalkulation, zu arbeiten. Dabei bietet es schnelles Laden, Ausrichten, Manipulieren, Zusammenführen und weitere Schlüsselfunktionen. Pandas wird geschätzt, da sie eine hochgradig optimierte Leistung bietet.⁸⁶

json: JSON ist die am meisten verwendete Methode, um Daten im Web zu senden und zu empfangen. JSON, steht für JavaScript Object Notation und wird auch in vielen anderen Programmiersprachen verwendet. Um in Python mit JSON zu arbeiten, kann man die json-Bibliothek von Python nutzen, die JSON-Strings in ein Python-Dict (Dictionary) umwandeln kann, welches dann weiterverwendet wird. Die Bibliothek bietet zusätzlich Funktionen zur Bereinigung und Verarbeitung dieser Daten, welche oft gebraucht werden.⁸⁷

⁸⁵ „Os — Miscellaneous Operating System Interfaces“.

⁸⁶ „What Is Pandas Python?“

⁸⁷ „Python JSON“.

random: Die random Bibliothek bietet Funktionen, zur Generierung von Zufallszahlen oder zur Auswahl zufälliger Elemente aus einer Sequenz. Dies ist nützlich, um Zufälligkeit in Anwendungen zu bringen. In dieser Arbeit wird diese Library zur zufälligen Auswahl von Anweisungen oder Fragen, die mit Produkttiteln kombiniert werden, genutzt. Durch diese Methode wird eine Vielfalt in den generierten Daten erzeugt. Das ist besonders bei der Erstellung von Trainingsdaten von Vorteil.⁸⁸

⁸⁸ „Python Random Module“.

6.4. Struktur der Daten

Die Trainingsdaten wurden von der Firma Blum in Form von JSON-Dateien, welche in Abbildung 21 zu sehen sind, bereitgestellt. Allerdings enthielten sie zahlreiche Informationen, die für den Chatbot irrelevant sind, wie beispielsweise Werbeinhalte und Navigationslinks. Diese könnten die Qualität und Effektivität des Trainingsprozesses beeinträchtigen. Für das Training dieser Daten war eine sorgfältige Bereinigung und Umstrukturierung erforderlich, um sie in ein Chatbot-freundliches Format zu bringen. Dieses Format ist am Modell von Stanford Alpaca angelehnt, welches in Abbildung 22 dargestellt ist und sich durch seine strukturierte und klare Datenorganisation auszeichnet. Es besteht aus Anweisungen, optionalen Kontextinformationen und erwarteten Antworten, was die Effizienz des Lernprozesses und die Relevanz der generierten Antworten optimiert. Dieses Format wurde gewählt, da es sich für das Training des LLMs Llama durchgesetzt hat und sich durch seine effektive und effiziente Lehrstruktur auszeichnet.

Technisch wurde die Datenaufbereitung mit einem Skript umgesetzt, welches die Daten bereinigt und sie in das gewünschte Format überführt. Dieses Skript enthält Abläufe für die Vorbereitung von Daten durch Initialisierung, Modulimporte für das Einlesen von Daten, Datenbereinigung und Datenextraktion sowie Datenverarbeitung und -speicherung, um die Daten effizient für das Training des Chatbot-Systems zu verwenden. Indem die Datenstruktur an das Standford Alpaca-Modell angepasst wurde, wurde eine solide Grundlage für die Entwicklung eines leistungsfähigen und effektiven Chatbot-Systems geschaffen.

```
{
    "category": "products",
    "name_autocomplete": ["products",
        "Anleitungen zur Fehlerbehebung bei Problemen mit der Inbetriebnahme oder Benutzung von SERVO-DRIVE",
        "SERVO-DRIVE Anleitung zur Fehlerbehebung | Blum",
        "https://www.blum.com/at/de/landingpages/produkte/servo-drive/stoerungsbehebung/",
        "SERVO-DRIVE Anleitung zur Fehlerbehebung | Blum Suche Produkte Services Unternehmen Kontakt Katalog Blum Inspirations E-SERVICES Produkte Neuheit
        "Tue Oct 10 10:10:25 UTC 2023",
        "de",
        "/corporate/media/bilder/produkte/bewegungstechnologien/Blum_IMG1252_TableImageLarge.jpg",
        "false",
        "20231010121011",
        "<span>landingpages</span> <span>Produkte</span> <span>servodrive</span> <span>Störungsbehebung</span> ",
        "6661354480e89837de5c11d56af786bb",
        "www.blum.com",
        "Finden Sie Neuigkeiten, Produkte, Services und Inspirationen SERVO-DRIVE Anleitung zur Fehlerbehebung Der nachstehenden Troubleshooting Guide ent
        "4.499938E-5",
        "https://www.blum.com/at/de/landingpages/produkte/servo-drive/stoerungsbehebung/",
        "AT",
        "SERVO-DRIVE, Störungsbehebung, Fehlersuche, Fehlerbehebung",
        "177936287183824486"],
        "description": "Anleitungen zur Fehlerbehebung bei Problemen mit der Inbetriebnahme oder Benutzung von SERVO-DRIVE",
        "title": "SERVO-DRIVE Anleitung zur Fehlerbehebung | Blum",
        "url": ["https://www.blum.com/at/de/landingpages/produkte/servo-drive/stoerungsbehebung/"],
        "content": ["SERVO-DRIVE Anleitung zur Fehlerbehebung | Blum Suche Produkte Services Unternehmen Kontakt Katalog Blum Inspirations E-SERVICES Produkte
        "tstamp": ["2023-10-10T10:10:25.379Z"],
        "language": "de",
        "searchresult_image": "/corporate/media/bilder/produkte/bewegungstechnologien/Blum_IMG1252_TableImageLarge.jpg",
        "exclude_from_searchresult": "false",
        "segment": [20231010121011],
        "metatag_breadcrumb_for_searchresult": ["<span>landingpages</span> <span>Produkte</span> <span>servodrive</span> <span>Störungsbehebung</span> "],
        "digest": ["6661354480e89837de5c11d56af786bb"],
        "host": ["www.blum.com"],
        "strippedContent": ["Finden Sie Neuigkeiten, Produkte, Services und Inspirationen SERVO-DRIVE Anleitung zur Fehlerbehebung Der nachstehenden Troublesh
        "boost": [4.499938E-5],
        "id": "https://www.blum.com/at/de/landingpages/produkte/servo-drive/stoerungsbehebung/",
        "country": "AT",
        "keywords": ["SERVO-DRIVE, Störungsbehebung, Fehlersuche, Fehlerbehebung"],
        "_version_": 177936287183824486
    ]
}
```

Abbildung 21: Blum-Quelldatei vor Data-Cleaning

```
{
    "instruction": "Was ist der Zweck von Bewegungstechnologien ?",
    "input": "BLUMOTION, SERVO-DRIVE, TIP-ON, TIP-ON BLUMOTION, handless, grifflose Möbel, grifflos, Soft-Close, softclose, tipon bluemotion, bluemotion, tipon, serv
    "output": "Die vier Bewegungstechnologien BLUMOTION, SERVO-DRIVE, TIP-ON BLUMOTION und TIP-ON lassen Klappen, Scharniere und Auszüge faszinierend öffnen und schlie
    "index": 0
}
```

Abbildung 22: Blum-Quelldatei nach Data-Cleaning

Standford Alpaca

Standford Alpaca ist ein Projekt, dessen Ziel es ist ein Llama-Modell zu erstellen das Anweisungen befolgen kann. Das Alpaca-Modell wurde anhand eines 7B Llama-Modells auf 52.000 *instruction-following-Daten* abgestimmt. Diese Daten wurden durch die Techniken der Self-Instruct-Veröffentlichung generiert. Die folgenden Personen waren gleichermaßen an diesem Projekt beteiligt:

- Rohan Taori
- Ishaan Gulrajani
- Tianyi Zhang
- Yann Dubois
- Xuechen Li⁸⁹

⁸⁹ „tatsu-lab/stanford_alpaca“.

6.5. Auswirkungen sauberer Daten auf Chatbot-Leistung

Die Qualität der Daten hat einen direkten Einfluss auf die Leistung von Chatbot-Systemen. Deshalb sind sie unerlässlich, um die Leistung und die Effektivität des Chatbots zu gewährleisten. Die Datenbereinigung verbessert die Genauigkeit und das Verständnis des Chatbots. Dies führt zu einer ausgefilterten Interpretation von Nutzeranfragen, was wiederum die Genauigkeit der Antworten des Chatbots erhöht.⁹⁰

Ein Chatbot, der auf sauberen Daten basiert, kann Fragen besser interpretieren, wodurch die Zufriedenheit der Benutzer gesteigert werden kann. Qualitativ hochwertige Daten dienen außerdem als Prävention gegen ungewollte Vorurteile und fördern somit eine faire und ethische Interaktion mit dem Benutzer.⁹¹

Ebenso beeinflusst die Sauberkeit der Daten die Zuverlässigkeit des Chatbots. Ein Chatbot, der durch saubere Daten immer angemessen und konsistent reagiert, kann das Vertrauen der Benutzer festigen und somit zu einer langfristigen Benutzerbindung beitragen.⁹²

Mit anderen Worten kann man sagen, dass saubere, gut strukturierte Daten die Fähigkeit des Chatbots, relevante und genaue Antworten zu generieren, verbessern, was zu einer besseren Benutzererfahrung führt.

⁹⁰ „The Essential Role of Data Cleaning in Chatbot Training“, 22. November 2023.

⁹¹ „The Essential Role of Data Cleaning in Chatbot Training“.

⁹² „The Essential Role of Data Cleaning in Chatbot Training“.

6.6. Herausforderungen

Das Data-Cleaning hat einige Herausforderungen mit sich gebracht, wie das Erkennen und Beheben von inkonsistenten Datenformaten, Duplikaten, fehlenden Werten und Ausreißern.

Herausforderung 1

Eine Schwierigkeit bestand darin, den verbleibenden Text nach dem Abschnitt „Das könnte Sie auch interessieren“ zu löschen. Diese Abschnitte wurden als irrelevant eingestuft, weil sie keine Produktinformationen enthielten und für das Training von dem Chatbot somit überflüssig sind.

Um diese Aufgabe zu lösen, wurde eine Data-Cleaning-Funktion programmiert, die im nachfolgenden Code-Abschnitt zu sehen ist. Diese Funktion ermöglicht es, den Inhalt ab diesem bestimmten Punkt zu entfernen. Dadurch wird sichergestellt, dass alle nicht relevanten Informationen ab diesem Punkt entfernt worden sind.



```
30 # Datacleaningfunktion
31 def clean_content(content):
32     if isinstance(content, list):
33         content = " ".join(content)
34
35     #Das könnte Sie auch interessieren
36     lösche_index = content.find("Das könnte Sie auch interessieren")
37     if lösche_index != -1:
38         content = content[:lösche_index]
```

Code-Abschnitt 1: Data-Cleaning-Funktion

Herausforderung 2

Ein weiteres Problem stellte die Notwendigkeit dar, die Datenbasis auf spezifische Bereiche zu filtern, um nur bestimmte Blum-Produkte anzeigen zu können. Daher war eine Lösung mit flexiblen Filteroptionen nötig.

Um dieses Problem zu lösen, wurde eine Filterfunktion entwickelt, die auf Parameter für spezifische Datenbereiche oder individuelle Indizes reagiert. Indizes werden auf zwei Arten bestimmt. Die erste Art ist mit Bereichsangaben. Hier definiert der Nutzer eine oder mehrere Bereiche mit jeweils einem Anfangs- und einem End-Index. Diese Funktion erzeugt dann eine Folge von Indizes für jeden angegebenen Bereich. Beispielsweise wird durch die Angabe eines Bereichs (2, 5) eine Folge von Indizes [2, 3, 4, 5] erstellt. Die zweite Möglichkeit Indizes zu bestimmen ist mit einzelnen Indizes. Hier gibt der Nutzer eine Folge von bestimmten Indizes an, welche unabhängig von einem Bereich ausgewählt werden können. Zum Beispiel kann diese Liste [1, 4, 8] enthalten. Dann werden auch nur die drei Indizes erstellt. Mit dieser Funktion wird den Benutzern dieses Skriptes ermöglicht, die Datensätze anzuzeigen, die sie sehen möchten. Durch Verwendung dieser Funktion ist es möglich, das finale *DataFrame* zu aktualisieren, um nur die selektierten Daten darin zu behalten. Im unten stehenden Code-Abschnitt ist die Funktion zu sehen.



```

115 # Funktion um Filterung auf spezifische Bereiche möglich zu machen
116 if ranges or individual_indices:
117     selected_indices = set()
118     if ranges:
119         for start, end in ranges:
120             selected_indices.update(range(start, end + 1))
121     if individual_indices:
122         selected_indices.update(individual_indices)
123     df_final = df_final[df_final['index'].isin(selected_indices)]

```

Code-Abschnitt 2: Filterfunktion

Fazit

Die Implementierung dieser Ansätze hat wesentlich zur Verbesserung der Datenverarbeitung und -analyse beigetragen, indem sie die Relevanz und Flexibilität der Daten erhöht hat.

6.7. Technische Umsetzung

Die technische Umsetzung von Data-Cleaning umfasst den Einsatz von verschiedenen Methoden und Techniken, um Daten zu reinigen. Unter diese Methoden und Techniken fallen die Identifikation von Fehlern, das Normalisieren von Datenformaten, das Bearbeiten von fehlenden Werten und das Entfernen von Duplikaten. Python und die Pandas-Bibliothek eignen sich perfekt, da sie diverse Funktionen für das Data-Cleaning und zur Datenmanipulation anbieten, welche den Prozess effizient gestalten.

Ziel des Skripts

Das Hauptziel des Skripts ist es die von Blum bereitgestellten Daten so zu formatieren und zu reinigen, dass sie keine Irrelevanzen oder Fehler enthalten und somit als Trainingsdaten für einen Chatbot verwendet werden können.

Anleitung

Um diesen Code einsetzen zu können, müssen zunächst Python sowie die notwendigen Libraries installiert werden. Ebenso muss man sicherstellen, dass die Daten, die gesäubert werden müssen, in Form einer JSON-File verfügbar sind.

Danach muss man im Code das korrekte Verzeichnis angeben, in der die JSON-Dateien sind. Als nächstes muss man im Code das gewünschte Format zur Ausgabe formatieren. In diesem Projekt wurde das Format von Standford Alpaca ausgesucht. Im nächsten Schritt muss man alle inkonsistenten, nicht relevanten und falschen Informationen aus den JSON-Daten kopieren und in die Replacements einfügen. Diese befindet sich innerhalb der clean_content-Funktion. Nun muss man die Fragen und Anweisungen, welche verwendet werden sollen, in die instructions und questions referenzieren. Man muss diese den Anforderungen des Projektes anpassen.

Nach der Anpassung des Codes kann man die Python-Datei ausführen. Dieser Schritt setzt die Datenverarbeitung in Gang und das Ergebnis wird im Zielpfad erstellt. Als nächstes kann man die neue JSON-Datei öffnen und kontrollieren, ob alles passt und den Erwartungen entspricht.

Durch das Befolgen dieser Schritte lässt sich der Code erfolgreich in ein Projekt integrieren und für die gewünschten Datenverarbeitungsaufgaben nutzen.

Skript-Ablauf

Initialisierung und Modulimporte (Zeilen 1-4)

Wie man im unteren Code-Abschnitt sehen kann, beginnt das Skript mit dem Import der erforderlichen Module:

- Das Modul **os** wird für Operationen verwendet, die mit dem Betriebssystem interagieren, wie das Navigieren in Dateisystemen.
- **pandas**, importiert als **pd**, dient der effizienten Datenmanipulation und -analyse.
- **json** ermöglicht das Parsen und Speichern von Daten im JSON-Format.
- Das Modul **random** wird genutzt, um Zufälligkeit bei der Auswahl von Daten zu gewährleisten.



```
1 import os  
2 import pandas as pd  
3 import json  
4 import random
```

Code-Abschnitt 3: Importierte Libraries

Pfadbestimmung (Zeilen 6-11)

Wie im nachfolgenden Code-Abschnitt zu sehen ist bestimmt die Funktion zu Beginn den Pfad zur JSON-Datei, die die Produktinformationen enthält. Durch Kombination der Methoden `os.getcwd()` und `os.path.dirname()` wird der übergeordnete Verzeichnispfad ermittelt, in dem sich die Datei befindet.



```
6  def main(ranges=None, individual_indices=None):
7      # Pfadangaben
8      current_directory = os.getcwd()
9      parent_directory = os.path.dirname(current_directory)
10     trainings_folder_path = os.path.join(parent_directory, "Trainingsdateien")
11     file_path = os.path.join(trainings_folder_path, "solr-index-blum.com-products-AT-de.json")
```

Code-Abschnitt 4: Pfadbestimmung

Daten Einlesen (Zeilen 13-29)

Die JSON-Datei wird mithilfe von **pandas** eingelesen. Zusätzlich werden Listen mit Anweisungen und Fragen initialisiert, die zur späteren Formatierung der Daten verwendet werden. Dies ist im nachfolgenden Code-Abschnitt dargestellt.

```
● ● ●  
13 # Einlesen der JSON-Datei mit Pandas  
14 df = pd.read_json(file_path)  
15  
16 # Liste aus Fragen und Anweisungen  
17 instructions = [  
18     "Zeig mir die Eigenschaften von ",  
19     "Erkläre mir die Funktionen von ",  
20     "Erzähl mir etwas über ",  
21  
22 ]  
23  
24 questions = [  
25     "Was sind die verschiedenen Eigenschaften von ",  
26     "Was sind die technischen Spezifikationen von ",  
27     "Was ist der Zweck von "  
28 ]  
29
```

Code-Abschnitt 5: Einlesen der Daten, Fragen und Anweisungen

Datenbereinigungsfunktion clean_content() (Zeilen 30-47)

Wie in Code-Abschnitt 6 zu sehen ist, ist diese Funktion für die Entfernung nicht relevanter Inhalte aus den Textdaten zuständig. Speziell wird nach einem Marker (z.B.: Das könnte Sie auch interessieren) gesucht und alles danach entfernt, um nur den relevanten Text beizubehalten.

```
30 # Datacleaningfunktion
31 def clean_content(content):
32     if isinstance(content, list):
33         content = " ".join(content)
34
35     #Das könnte Sie auch interessieren
36     lösche_index = content.find("Das könnte Sie auch interessieren")
37     if lösche_index != -1:
38         content = content[:lösche_index]
39
40     replacements = [
41
42     ]
43
44     for r in replacements:
45         content = content.replace(r, "")
46
47     return content
```

Code-Abschnitt 6: Komplette Data-Cleaning-Funktion

Datenextraktionsfunktion extract_data() (Zeilen 49-65)

extract_data() extrahiert wichtige Informationen aus jedem Dokument. Dabei wird zufällig entschieden, ob eine Anweisung oder eine Frage mit dem Titel des Dokuments kombiniert wird, um Vielfalt in die Datenpräsentation zu bringen. Dieser Teil vom Code ist im nachfolgenden Code-Abschnitt zu sehen.



```
49 # Funktion zur Extraktion und Formatierung der Daten
50 def extract_data(doc, index):
51
52     title = "".join(clean_content(doc.get("title", [""])))
53
54     if random.choice([True, False]):
55         instruction_or_question = random.choice(instructions) + title
56     else:
57         instruction_or_question = random.choice(questions) + title + "?"
58
59     return {
60         "instruction": instruction_or_question,
61         "input": " ".join(doc.get("keywords", [])),
62         "output": " ".join(doc.get("description", "")) + " " + clean_content(doc.get("strippedContent", "")),
63         "index": index
64     }
65
```

Code-Abschnitt 7: Daten-Extraktions-Funktion

Datenverarbeitung und -speicherung (Zeilen 68-94)

Wie im nachfolgenden Code-Abschnitt **Error! Reference source not found.** dargestellt wird, verarbeitet das Skript in den Zeilen 68-94 die Dokumente und filtert sie anhand der übergebenen Parameter. Die verarbeiteten Daten werden in ein neues JSON-Format umgewandelt und unter einem spezifizierten Pfad gespeichert.



```

66     # Extraktion der Dokumente
67     docs = df.loc['docs', 'response']
68
69     processed_docs = [extract_data(doc, i) for i, doc in enumerate(docs) if isinstance(doc, dict)]
70
71     # Verarbeitete Dokumente in ein DataFrame konvertieren
72     df_final = pd.DataFrame(processed_docs)
73
74     # Funktion um Filterung auf spezifische Bereiche möglich zu machen
75     if ranges or individual_indices:
76         selected_indices = set()
77         if ranges:
78             for start, end in ranges:
79                 selected_indices.update(range(start, end + 1))
80         if individual_indices:
81             selected_indices.update(individual_indices)
82         df_final = df_final[df_final['index'].isin(selected_indices)]
83
84
85     data_list = df_final.to_dict(orient='records')
86
87     # Speichern der Daten als JSON
88     output_file_path = os.path.join(trainings_folder_path, "product-data-de.json")
89     with open(output_file_path, 'w', encoding='utf-8') as f:
90         json.dump(data_list, f, ensure_ascii=False, indent=4)
91
92     print("Data saved to:", output_file_path)
93
94     # Beispiel um die Datenn zwischen dem index 180 to 184 und index 5 zu sehen folgendenes unten anschreiben:
95     # main(ranges=[(180, 184)], individual_indices=[5])
96     # wenn man alles sehen möchte: main()
97     main()

```

Code-Abschnitt 8: Datenverarbeitung und Speicherung

Beispielnutzung (Zeilen 96-99)

Ein Beispielauftrag der `main()` Funktion ist kommentiert angegeben. Er zeigt, wie Daten für bestimmte Indizes gefiltert und verarbeitet werden können. Die Funktion wird am Ende des Skripts aufgerufen, um den Verarbeitungsprozess zu initiieren. In der folgenden Abbildung sieht man eine JSON-Datei welche die Blum-Produkte 12-14 und Produkt Nummer 5 bereinigt und richtig strukturiert darstellt, wenn man die `main()` Funktion für diese Indizes filtert. Dafür muss man im Code nur die `main()` zu

`main(ranges=[(12, 14)], individual_indices=[5])` abändern. In der Abbildung 23 sieht man die in der `main()`-Funktion angegebenen 4 Produkte.

```

1  [
2   {
3     "instruction": "Was sind die verschiedenen Eigenschaften von SERVO-DRIVE ?",
4     "input": "elektrische Bewegungsunterstützung, SERVO-DRIVE, Öffnungsunterstützung, Antrieb, Elektroantrieb, elektrischer An
5     "output": "Mit der elektrischen Öffnungsunterstützung öffnen Klappen, Auszüge und Kühlgeräte wie von selbst. Finden Sie Ne
6     "index": 5
7   },
8   {
9     "instruction": "Zeig mir die Eigenschaften von Boxsysteme ",
10    "input": "TANDEMBOX, LEGRABOX, Schublade, Schubkasten, Frontauszug, Auszug, Lade, Schubladensystem, Schubsystem",
11    "output": "Blum bietet 3 doppelwandige Boxsysteme für unterschiedliche Anforderungen in der Möbelgestaltung: LEGRABOX, MER
12    "index": 12
13  },
14  {
15    "instruction": "Was sind die technischen Spezifikationen von MOVENTO ?",
16    "input": "Führungssystem, synchronisierter Schwebelauf, 4-dimensionale Fronteinstellung, dynamische Belastbarkeit 40 kg un
17    "output": "Das Raffinierte unter den Führungssystemen. Finden Sie Neuigkeiten, Produkte, Services und Inspirationen MOVENTO
18    "index": 13
19  },
20  {
21    "instruction": "Erkläre mir die Funktionen von Tablar-Arretierung ",
22    "input": "Fixierung, Verriegelung, Regalbrett ausziehbar, Regalboden ausziebar, verriegelbar, Auszugstablar, ausziehbare A
23    "output": "Arretierung für Tablare mit TANDEM und MOVENTO Finden Sie Neuigkeiten, Produkte, Services und Inspirationen Tab
24    "index": 14
25  }
26 ]
27 ]

```

Abbildung 23: Zurückgegebene Produkte

Bedeutung für die Diplomarbeit

In der Diplomarbeit wird die Entwicklung eines Chatbots durch den Einsatz von strukturierten und bereinigten Trainingsdaten mithilfe dieses Codes unterstützt. Dieser Code ist sehr wichtig, um die Effektivität und Genauigkeit des Chatbots zu erhöhen. Der für die Diplomarbeit entwickelte Code bietet zudem die Flexibilität, für spezifische Anforderungen oder Datensätze angepasst zu werden.

7. Training der LLM

Das Training einer LLM umfasst die Anpassung an eine neue Aufgabe mittels eines spezifischen Datensatzes, der für diese Aufgabe geeignet ist. Dabei lernt die LLM, Muster und Zusammenhänge in den Daten zu erkennen und kann entsprechend den Anforderungen der neuen Aufgabe Vorhersagen treffen. Es werden verschiedene Arten des Finetunings und Trainierens sowie die Architekturen von neuronalen Netzwerken oder LLMs und die Erreichung ihrer Leistung behandelt. Im Anschluss wird auch die technische Umsetzung anhand der Blum-Produktdaten erläutert.

7.1. Ziel des Trainings

Das Hauptziel des Trainings in diesem Projekt ist die Anpassung eines sorgfältig ausgewählten Large-Language-Models, damit es Kundenanfragen spezifisch mit Blum-Produktdaten annehmen kann. Das bedeutet, die LLM so zu trainieren, dass sie als effektiver Chatbot dient, der präzise Produktinformationen sowohl auf Englisch als auch auf Deutsch liefern kann.

7.2. Architekturen von neuronalen Netzwerken

Die Grundlage eines vortrainierten Modells ist seine Architektur, die je nach beabsichtigter Aufgabe anders konzipiert ist. Solch eine Architektur ist in der Regel ein komplexes *neuronales Netzwerk*, das für generelle Aufgaben wie beispielsweise Sprachverständnis oder Bildklassifizierung entworfen ist. Für das Sprachverständnis werden häufig Recurrent Neural Networks (RNN) oder die neuere Transformer-Architektur verwendet, während im Bereich der Computer-Vision oft Convolutional Neural Networks (CNN) zum Einsatz kommen. Nachfolgend sind einige der bekanntesten Architekturen für das Sprachverständnis aufgelistet:

Convolutional Neural Networks (CNN)

Convolutional Neural Networks, auch kurz als CNN bekannt, sind eine Art von neuronalen Netzwerken, welche spezifisch für die Verarbeitung von Bildern entwickelt wurden, aber auch im Bereich des Sprachverständnisses oder der Audioverarbeitung eingesetzt werden kann. Die Architektur von CNNs ist stark von der Vernetzung des menschlichen Gehirns inspiriert, weil sie eine zentrale Rolle für die Verarbeitung der visuellen Reize spielen und die Architektur dieses Verhalts imitieren will. Dabei nutzen CNNs eine Reihe von Schichten, die verschiedene Merkmale des Eingabebildes erkennen und diese an das neuronale Netzwerk weiterleiten. Somit wird die effiziente Interpretation der visuellen Informationen aus ganzen Bildern möglich gemacht. Ihre Größe ist abhängig von der Komplexität der zu erkennenden Merkmale und kann dabei hunderte bis tausende von Schichten groß werden.⁹³

⁹³ „What Is a Convolutional Neural Network (CNN)?“

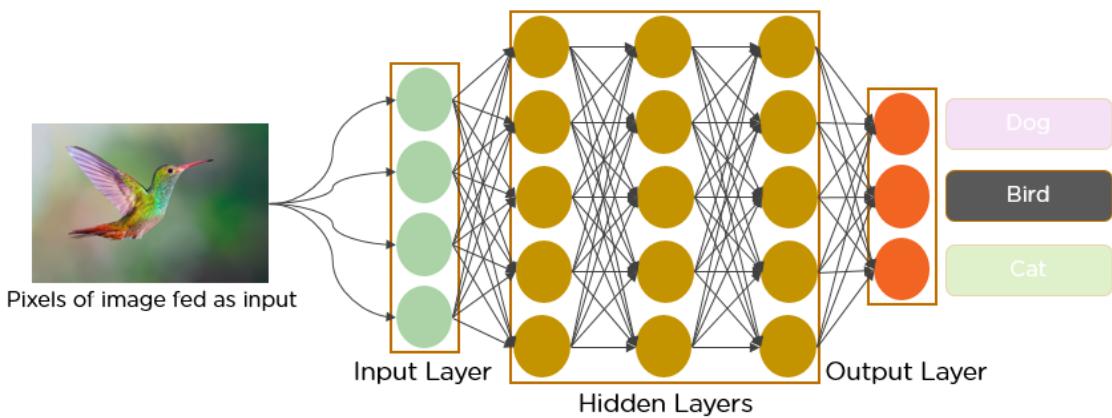


Abbildung 24: CNN⁹⁴

In der obigen Abbildung 24 sieht man die Architektur eines CNN und wie ein Eingangsbild eines Vogels in eine der verschiedenen Tierarten klassifiziert wird.

⁹⁴ https://editor.analyticsvidhya.com/uploads/25366Convolutional_Neural_Network_to_identify_the_image_of_a_bird.png

Recurrent Neural Networks (RNN)

Recurrent Neural Networks, oder auch kurz als RNN bekannt, sind eine Architektur von neuronalen Netzwerken, welche spezifisch für das Vorhersagen mit der Berücksichtigung des Kontextes entwickelt wurden. Um das wahrscheinlichste Ergebnis zu vorhersagen, erkennen sie wiederholende Merkmale in dem Kontext, um akkurate Vorhersagen zu treffen. Deshalb werden sie auch im Bereich der Large Language Models eingesetzt, da sie anhand des Kontexts präzise den nächsten Buchstaben in einem Wort, oder auch das nächste Wort in einem Satz vorhersagen können. Die Art von neuronalem Netzwerk führt auch eigene Antworten als Eingabe wieder zurück. Diese Fähigkeit wird auch als Recurrence bezeichnet, weil somit auch frühere Eingaben des Benutzers oder Ausgaben der LLM in den Antworten einbezogen werden.⁹⁵

Recurrent Neural Network

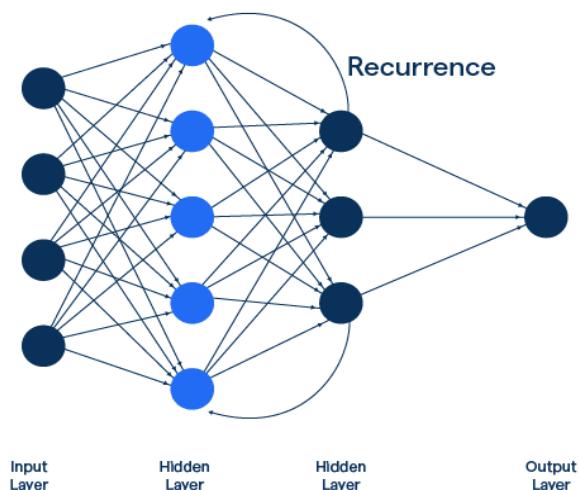


Abbildung 25: RNN⁹⁶

In der Abbildung 25 ist die RNN-Architektur abgebildet, und die Recurrence, die innerhalb dieser Architektur üblich sind.

⁹⁵ „What Are Recurrent Neural Networks and How Do They Work?“

⁹⁶ https://cdn.botpenguin.com/assets/website/Recurrent_Neural_Network_508b372642.png

Transformer

Das Transformer-Modell ist eine neuronale Netzwerk Architektur, welche den Typ eines Inputs auf einen anderen Typ als Output übertragen kann. Der Begriff hat sich durch eine wissenschaftliche Arbeit von Google im Jahre 2017 geprägt, in welchem ein Weg gefunden wurde, ein neuronales Netzwerk, für das Übersetzen vom Englischen ins Französische, mit einer höheren Genauigkeit und einem Bruchteil von der Trainingszeit von anderen neuronalen Netzwerken, zu trainieren. Die Transformer-Architektur besteht aus einem Encoder, welcher jedes Wort oder Token in Vektoren, die dabei durch die Wichtigkeit von anderen Tokens *gewichtet* werden, verwandelt und einem Decoder, der diesen Prozess im Wesentlichen ins Zielbereich umdreht, die zusammenarbeiten. Ein Vektor umfasst komplexe Muster, welche als Zahlenfolgen gespeichert werden.⁹⁷ Fast alle modernen Large-Language-Models und auch andere neuronale Netzwerke basieren auf dieser Architektur.⁹⁸

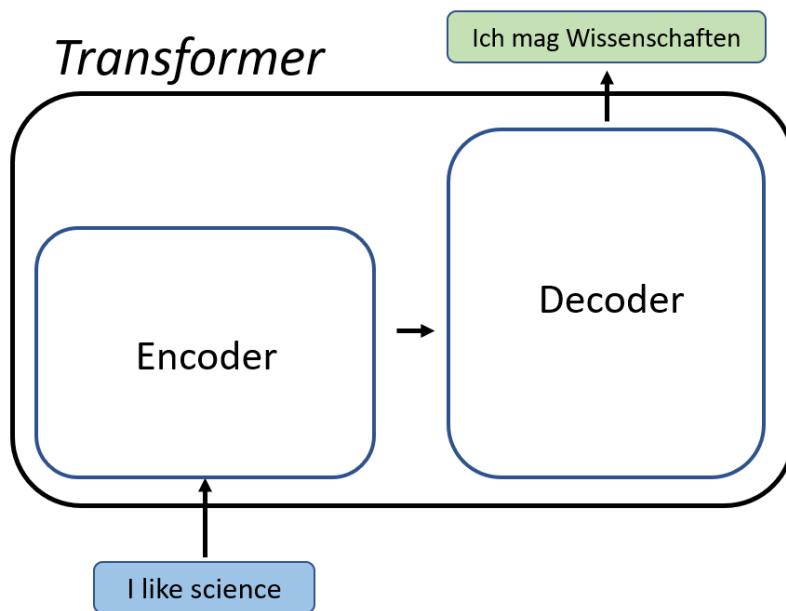


Abbildung 26: Transformer⁹⁹

In der obigen Abbildung 26 ist die Architektur von Transformers ersichtlich, welche einen englischen Input in die deutsche Sprache übersetzt.

⁹⁷ „What is a Vector?“

⁹⁸ „What Is a Transformer Model?“

⁹⁹ https://miro.medium.com/v2/resize:fit:904/1*GAQrbFIV-G5cT3-OchMEHg.png

7.3. Vortrainierte neuronale Netzwerke

Vortrainierte Modelle sind komplexe neuronale Netzwerke, die sowohl das Vortraining als auch eine gewisse Feinabstimmung durchlaufen haben. Beim Vortrainieren werden die Modelle anhand eines umfangreichen Datensatzes trainiert, der oft unstrukturierte und unbeschriftete Daten umfasst. Der Datensatz umfasst Daten zu dem generellen Bereich, auf welche das Modell antrainiert werden soll. Vor dem Vortraining ist es deshalb wichtig, die richtige Architektur für den eigenen Fall zu wählen, um die Ergebnisse zu optimieren. So wird beispielsweise ein Modell, welches eine Architektur für die Verarbeitung natürlicher Sprache besitzt, eher mit einem umfangreichen Textkorpus trainiert, der eine riesige Menge an schriftlichen Texten befasst. Während ein Modell für die Bilderkennung eher mit einer enormen Menge an Bildern trainiert wird. Das Vortraining schafft die Grundlage für das neuronale Netz, da es nach dem Training, Muster innerhalb des Datensatzes erkennen kann, welche für den Menschen oft nicht offensichtlich sind. Dadurch werden die Modelle mit dem generellen Aufbau der Sprache, den verschiedenen Merkmalen in Bildern oder anderen komplexen Daten vertraut gemacht und eignen sich dabei generelles Wissen an. Dieses Wissen wird in den Parametern und Gewichten des Modells gespeichert. Das neuronale Netzwerk ist zunächst noch nicht auf eine spezifische Aufgabe, sondern nur grob in einem bestimmten Gebiet ausgerichtet und wird erst mit der Feinabstimmung auf eine spezialisiert. Im Gegensatz zu dem Vortraining, wird bei der Feinabstimmung ein kleinerer und aufgabenspezifischer Datensatz verwendet, der oft beschriftete Daten enthält. Dabei ist es beispielsweise sinnvoll, ein auf Sprachverständnis vortrainiertes Modell auf sprachlich spezifische Aufgaben wie die Beantwortung von Fragen fein abzustimmen. Diese Art der Lernmethode wird auch als Transferlernen bezeichnet und da es effizient einem Modell neues Wissen beibringen kann, wird es oft eingesetzt. Dies kann viel Zeit und Ressourcen sparen, weil somit kein Modell von Grund auf neu trainiert werden muss.¹⁰⁰

¹⁰⁰ Otten, „Pre-Trained Models Complete Guide [How To & 21 Top Models In PyTorch, TensorFlow & Hugging-Face]“.

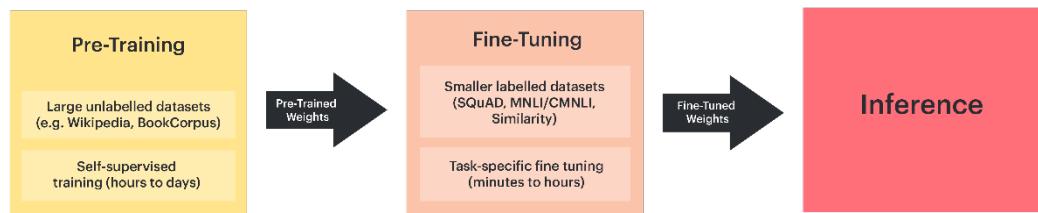


Abbildung 27: Vortrainingsablauf einer LLM¹⁰¹

In der Abbildung 27 ist der Ablauf eines vortrainierten LLMs zu sehen, einschließlich der Inference-Phase, die spezifisch den Prozess des Generierens von Antworten auf Benutzerfragen beschreibt.¹⁰²

¹⁰¹ https://docs.graphcore.ai/projects/bert-training/en/latest/_images/training-tuning-inference.png

¹⁰² „Der Unterschied zwischen KI-Inferenz und Training“.

7.3.1. Vorteile

Es gibt einige Vorteile, welche die Verwendung von neuronalen Netzwerken mit sich bringt, weshalb nur in seltenen Fällen das Training von Grund auf angebracht ist. Hierzu sind einige Vorteile aufgelistet:

Geringere Trainingskosten und Trainingsdauer

Einer der größten Gründe, warum die Entscheidung getroffen wird, ein vortrainiertes neuronales Netzwerk zu verwenden, statt eines von null auf anzutrainieren, sind die enormen Trainingskosten und die riesige Trainingsdauer. So brauchte der US-amerikanische Internetkonzern Meta, vormals bekannt unter dem Namen Facebook, für das Training ihrer Llama 2 LLM-Reihe, insgesamt 3.3 Millionen *Graphics-Processing-Unit*(GPU)-Stunden bei einem Energieverbrauch von 0.4 Kilowatt pro Stunde von der *Nvidia* Grafikkarte A100, welche mit 80 Gigabyte Video Random Access Memory (VRAM) ausgestattet ist, an Rechenleistung. Dies führt zu einem gesamten Stromverbrauch von 1.32 Millionen kWh, mit einem gerundeten Preis pro kWh von 30 Cent in Österreich, würde dies zu Stromkosten von 396 000 Euro führen¹⁰³. Dieser exorbitante Aufwand an Hardware-Kosten und Trainingszeit wird durch das Verwenden von vortrainierten Modellen eingespart.

Leistungsstarke Modelle

Ein anderer wichtiger Gesichtspunkt ist, dass die Leistung der vortrainierten Modelle, durch ihre immens hohen Trainingskosten, oft auf so einem hohen Niveau liegen, welche schwer oder gar unmöglich für ein Unternehmen wäre, wenn man das neuronale Netz von Grund auf trainieren würde.¹⁰⁴

Anpassungsmöglichkeiten

Zudem gibt es Möglichkeiten, die hochleistungsfähigen Modelle, wie mithilfe von Finetuning und einem kleinen Datensatz, oder das Training von *Adapters*, auf seinen eigenen spezifischen Fall anzupassen.¹⁰⁵

¹⁰³ „Llama/MODEL_CARD.Md at Main · Meta-Llama/Llama“.

¹⁰⁴ Otten, „Pre-Trained Models Complete Guide [How To & 21 Top Models In PyTorch, TensorFlow & Hugging-Face]“.

¹⁰⁵

Otten.

7.3.2. Nachteile

Auch wenn die Nutzung vortrainierter Modelle einige Vorteile mit sich bringt, sind jedoch auch einige Nachteile zu beachten, welche aber oft stark von dem Modell abhängig sind. Hier sind einige Nachteile aufgelistet:

Hohe Hardwareanforderungen

Eines der größten Nachteile von vortrainierten Modellen sind die hohen Parametergrößen, welche auch als Modellgrößen betrachtet werden können, sowohl die hohen Hardwareanforderungen, die sie mit sich bringen. In der Regel werden für jede Milliarde Parameter etwa 2 Gigabyte VRAM benötigt, um das Modell laufen zu lassen. Falls das Modell noch angepasst werden soll, sind etwa 4 Gigabyte VRAM pro Milliarde Parameter benötigt.

¹⁰⁶

Vorurteile (Bias)

Die vortrainierten Modelle könnten unerwünschte Vorurteile, welche in ihren Trainingsdaten vorliegen, enthalten. Dies sollte man stets beim Einsatz dieser Modelle beachten, um mögliche rassistische oder unerwünschte Antworten zu vermeiden. So könnten solche Modelle beispielsweise einige Präferenzen gegenüber einem bestimmten Geschlecht hegen.¹⁰⁷

¹⁰⁶ „A Guide to LLM Inference and Performance“.

¹⁰⁷ Otten, „Pre-Trained Models Complete Guide [How To & 21 Top Models In PyTorch, TensorFlow & HuggingFace]“.

Überanpassung (Overfitting)

Das gewählte Modell könnte auch überangepasst oder, auf Englisch, *Overfitted* sein. Man spricht von einer Überanpassung, wenn das neuronale Netz sich zu stark an die eigenen Trainingsdaten anpasst und dabei keine akkurate Ergebnisse auf neue Daten mehr generieren kann.¹⁰⁸

Blackbox

Large Language Models werden oft aufgrund ihrer Parametergrößen mit Milliarden an Parametern als Black-Boxes betrachtet, da sich oft nicht genau erschließen lässt, welche Anpassungen getroffen werden müssen, um die gewünschte Ausgabe zu erzielen. Selbst mit einem Verständnis der komplexen mathematischen *Algorithmen* hinter den Sprachmodellen, bleibt es trotzdem ein Herantasten, bis man eine angemessene Ausgabe erhält.¹⁰⁹

¹⁰⁸ „What is Overfitting? | IBM“.

¹⁰⁹ „The Black Box Problem“.

7.4. Arten des Finetunings

Die Auswahl der Trainingsmethode hängt von verschiedenen Faktoren ab, darunter die verfügbare Rechenleistung, der gewünschte Grad an Änderungen, die Qualität und das Format der Trainingsdaten sowie die gewünschte Zielanwendung des LLM. Die bekannten Optionen sind:

Full Parameter-Finetuning

Die Idee von Finetuning besteht darin, ein vorhandenes vortrainiertes Modell mithilfe eines Datensatzes an die eigenen Ziele anzupassen. Da beim Finetuning in der Regel alle Parameter des Modells angepasst werden, ein Prozess, der häufig als Full Parameter-Finetuning oder auch als Full Fine-Tuning bezeichnet wird, benötigt dieser eine erhebliche Menge an Rechenleistung und VRAM. Es gibt auch Alternativen, bei denen nicht alle Parameter verändert werden müssen, bekannt unter dem Namen Parameter-Efficient Fine-Tuning (PEFT)¹¹⁰, welche auch beim Supervised Fine-Tuning (SFT) verwendet werden kann. Diese fallen jedoch eher in den Bereich der Adapter-Anpassungsmethoden. Allerdings gibt es andere Methoden, die nicht alle Parameter anpassen und trotzdem ähnlich qualitative Ergebnisse liefern können, während sie viel effizienter sind. Aus diesem Grund scheidet diese Anpassungsmethode für das Training an Blum-Produktdaten aus.¹¹¹

¹¹⁰ „Parameter-Efficient Fine-Tuning using ☺ PEFT“.

¹¹¹ Drost, „Different Ways of Training LLMs“.

RLHF-Finetuning

Beim RLHF-Finetuning produziert das Modell mehrere Ausgaben für einen Prompt, die anschließend von einer Person nach verschiedenen Kriterien wie Nützlichkeit oder Angemessenheit bewertet werden. Eine mögliche Reihenfolge von 3 Ausgaben wäre, dass A > B > C ist, die dann verwendet wird, um ein Belohnungsmodell zu trainieren. Dies ist ein völlig neues Modell, welches dazu dient, die Ausgaben eines LLMs zu bewerten, indem es Belohnungen verteilt, die die Vorlieben des Menschen widerspiegeln. Sobald das Modell trainiert ist, kann es den Menschen ersetzen und der Prozess kann automatisiert werden. Allerdings erfordert es bis dahin, dass das Belohnungsmodell eigenständig die Ausgaben bewerten kann, einen erheblichen Aufwand an menschlich überarbeiteten Daten. RLHF wird oft verwendet, um die Ausgaben des LLMs mehr wie eine Unterhaltung scheinen zu lassen oder unerwünschtes Verhalten zu vermeiden, wie die Produktion von rassistischen, diskriminierenden oder anderweitig unangebrachten Inhalten. Diese spezielle Art der Anpassung stößt bei der Zielsetzung, das Wissensspektrum einer LLM substanzial zu erweitern, an ihre Grenzen. Daher eignet sie sich nicht, um die LLM an spezifische Datensätze, wie etwa Blum-Produktinformationen, anzupassen.¹¹²

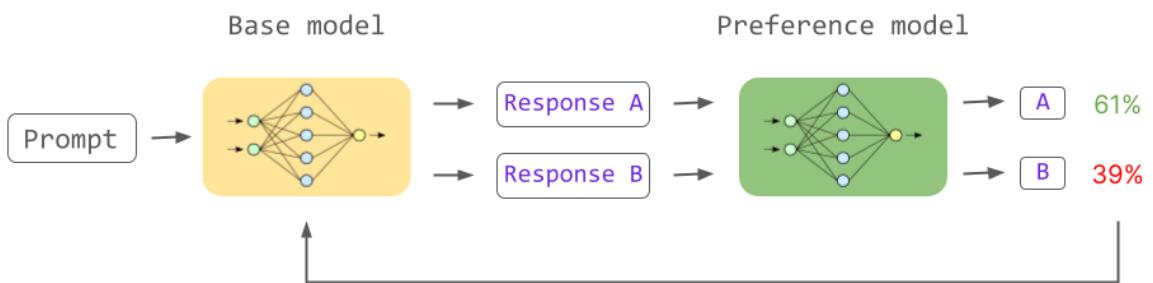


Abbildung 28: RHLF-Fine-Tuning¹¹³

In der obigen Abbildung 28 wird der Ablauf des RLHF-Fine-Tuning visualisiert.

¹¹² Drost.

¹¹³ https://lh6.googleusercontent.com/i91Ps5p_H6KP2nv4D1kBr_wmaf_pEks93gpKNRs01ip_rrnvzxN37Bjj2e3AXaJCRyFsT2CJpkgPMtcO2hr1xAieLJv4BFgR-NxZvN0T5moWbJsH0PMWkM4P7jTSTovuNr5H_0sS7fOEaoApZl38UtM

Adapter

Die Verwendung von Adapters bedeutet, zusätzliche Schichten zu einem bereits trainierten Modell hinzuzufügen. Während des Adaptertrainings wird nur der Adapter trainiert, während der Rest der Modellparameter eingefroren bleibt und somit überhaupt nicht verändert wird. Da diese Schichten um einiges kleiner sind als jene, die von dem Modell ausgeliefert werden, ist es einfacher, sie anzupassen. Dies macht Adapter besonders attraktiv bei großen LLMs, welche ein erhebliches Maß an VRAM und Rechenleistung brauchen, um diese zu trainieren, oder falls schnelle Anpassungen an spezifische Aufgabenstellungen gefordert sind. Zusätzlich können Adapter an verschiedenen Stellen innerhalb des Modells eingefügt werden, was eine Optimierung der Modellleistung auf spezifische Aufgaben ermöglicht. Da diese Methode zwischen den zwei anderen Methoden die effizienteste ist, wurde entschieden, mit Adapters das Modell auf die Blum-Daten anzupassen. Unter anderem löst sie auch mögliche GPU-VRAM-Überlastung, die auftreten könnten. Zudem erleichtert sie das Training zwischen Deutsch und Englisch durch die Flexibilität, die die Adapter mit sich bringen.¹¹⁴

¹¹⁴ Drost, „Different Ways of Training LLMs“.

7.5. Trainingsarten

Es gibt verschiedene Arten, wie die Blum-Produkte über Adapter trainiert werden können.

Die beiden Trainingsarten, die sich am stärksten durchgesetzt haben, sind Low Rank Adaptation (LoRA) und SFT:

Low Rank Adaptation (LoRA)

Die Entwicklung von LoRA geht auf eine Zusammenarbeit von Microsoft mit OpenAI im Jahr 2021 zurück, um die Herausforderung zu meistern, große Modelle wie GPT-3 wirtschaftlich effizient anzupassen. Riesige Sprachmodelle, wie GPT-4, Claude-2, Llama-70b, erweisen sich aufgrund ihrer enormen Größe und der damit verbundenen hohen Kosten als äußerst aufwendig durch traditionelles Full Fine-Tuning zu trainieren. LoRA nimmt dabei einen anderen Ansatz. Es friert das ursprüngliche Modell ein und fügt jeder Schicht einen kleinen, anpassbaren Teil hinzu. Dadurch wird die Anzahl der trainierbaren Parameter drastisch reduziert, was ebenfalls den benötigten GPU-Speicher erheblich verringert. Zum Beispiel würde die Anpassung des GPT-3-Modells durch Full Fine-Tuning das Training von 175 Milliarden Parametern erfordern. Mit LoRA sinkt diese Zahl um das 10.000-fache, und der benötigte GPU-Speicher verringert sich um das Dreifache. Das Training mit LoRA bringt eine erhöhte Geschwindigkeit mit sich, da es weniger Parameter zu trainieren gibt. Es benötigt weniger Rechenleistung, weil weniger Parameter auch weniger Rechenkapazität beanspruchen, und es ist speichereffizienter, was die Anforderungen an den VRAM erheblich reduziert. Aufgrund dieser Vorteile scheidet das Training über LoRA nicht aus.¹¹⁵

¹¹⁵ „Mastering Low-Rank Adaptation (LoRA)“.

Supervised Fine-Tuning (SFT)

Für das Training mit SFT werden zunächst eine Reihe hochwertiger und gewünschter LLM-Ausgaben gesammelt, die oft mit einem Prompt, also einer Benutzereingabe, in einem Datensatz zusammengeführt werden. Das „Supervised“ im Namen deutet darauf hin, dass ein Datensatz gesammelt wird, den das Modell nach dem Training nachbilden sollte. Im Laufe des Trainings lernt das Modell dann, den Stil der Datensätze zu imitieren. SFT unterscheidet sich ein wenig vom üblichen Fine-Tuning, genauer vom Full-Parameter-Fine-Tuning, da es während des Trainings Aufgaben außerhalb des Trainingsgebiets nicht verlernt, weil es lediglich einen korrekten Stil oder ein Verhalten imitiert. Zudem bietet es die Möglichkeit, mithilfe von PEFT, nicht alle Parameter zu trainieren, sondern nur einen Bruchteil davon. Das größte Problem von SFT ist, dass die Qualität des Trainings am stärksten von seinem Datensatz abhängig ist und einen qualitativen Datensatz zu beschaffen viel Zeit und Geld kostet. Da die Blum-Daten zu jedem Produkt auch eine Produktbeschreibung liefern, ist es möglich, über SFT das Llama-Modell an die Blum-Produktdaten anzupassen. Da Supervised Fine-Tuning in der Theorie mehr GPU-VRAM LoRA nutzt, aber möglicherweise bessere Ergebnisse liefern könnte, scheidet diese Trainingsmethode ebenfalls nicht aus. Um die Trainingsentscheidung zu treffen, werden die Ergebnisse anhand eines kleinen Blum-Produktdatensatzes beider Trainingsmethoden verglichen.¹¹⁶

¹¹⁶ Ph.D., „Understanding and Using Supervised Fine-Tuning (SFT) for Language Models“.

7.6. Einfluss der Hyperparameter

Das Training ist stark von den *Hyperparametern* beeinflusst deswegen sind hier einige verwendete Hyperparameter und deren kurze Definitionen. Zuerst werden die *Trainingsargumente* von Huggingface erläutert.

Tabelle 2: Hyperparameter-Definition 1

Parameter	Definition
Steps	Die Anzahl der Schritte, die das Modell während des Trainings pro <i>Epoche</i> durchläuft. Ein Schritt entspricht normalerweise einer Iteration über einen Teil des Datensatzes. ¹¹⁷
Batches	Batches sind kleine Gruppen an Datenpunkten aus dem Datensatz, welche gemeinsam zur Aktualisierung der Modellparameter verwendet werden. Dies ermöglicht, dass der gesamte Datensatz in kleineren Teilen aufgeteilt werden, welches ein effizientes Training ermöglicht. ¹¹⁸
learning rate	Die Lernrate ist ein Parameter, der die Anpassung der Gewichte des Netzwerks basierend auf seiner Fehlerrate steuert. Es gibt keine perfekte Lernrate für jedes Modell, und sie muss immer durch Trial and error für den spezifischen Fall gefunden werden. Die Reichweite der Lernraten, die in Betracht gezogen werden können, beträgt 1.0 bis zu 10e-6. ¹¹⁹
Optimizer	Der Optimierungsalgorithmus bestimmt, wie die Gewichte des Modells während des Trainings angepasst werden, um eine Zielfunktion, wie den Verlust oder die Genauigkeit, zu minimieren oder zu maximieren. Beliebte Optimizer sind Adam, SGD und RMSprop. ¹²⁰

¹¹⁷ Duca, „What Are Steps, Epochs, and Batch Size in Deep Learning“.

¹¹⁸ Editor, „What Is Batch Size“.

¹¹⁹ Brownlee, „Understand the Impact of Learning Rate on Neural Network Performance“.

¹²⁰ „Introduction to Optimizers“.

Tabelle 3: Hyperparameter-Definition 2

Parameter	Definition
Lr-scheduler-type	Der Lernratenplan ist ein Parameter, der die Lernrate zwischen Epochen oder Iterationen anpasst, während das Training fortschreitet. Er kann konstant sein, sodass sich die Lernrate während des Trainings nicht ändert, oder abnehmend, wobei die ursprüngliche Lernrate gemäß dem Zeitplan reduziert wird. Der Typ des Zeitplans hängt oft von dem verwendeten Optimierer ab und was dieser empfiehlt. ¹²¹
Warmup-ratio	Das Warmup-Verhältnis gibt an, in welchem Verhältnis die Trainingsschritte von 0 auf die gewünschte Lernrate linear ansteigen sollen. Dies vermeidet eine frühzeitige Überanpassung, bei der sich das Modell stark an unerwünschte Merkmale anlehnt, welche dann das gesamte Training beeinflussen können. ¹²²
Gradient accumulation steps	Gradientenakkumulation ist ein Mechanismus, der das Problem adressiert, dass Batchgrößen durch den GPU-Speicher limitiert sind. Dabei werden die Batches in mehrere Mini-Batches aufgeteilt, die sequenziell verarbeitet werden. Es wird eine festgelegte Anzahl von Schritten durchgeführt, während die Parameteränderungen berechnet, aber die Modellparameter nicht direkt angepasst werden. Dies ermöglicht, dass alle Mini-Batches dieselben Modellparameter verwenden, um die Parameteränderungen zu berechnen, die dann aufsummiert und dem Original-Batch wieder zugeführt werden. ¹²³

¹²¹ Li, „How to Choose a Learning Rate Scheduler for Neural Networks“.¹²² Prune, „Answer to "What does 'learning rate warm-up' mean?“¹²³ Rotenberg, „What Is Gradient Accumulation in Deep Learning?“

Tabelle 4: Hyperparameter-Definition 3

Parameter	Definition
Gradient-checkpointing	Der Parameter Gradient Checkpointing legt fest, ob die Gradienakkumulation aktiviert ist oder nicht.
Weight decay	Gewichtsabnahme, auch bekannt als L2-Regularisierung, ist eine Methode zur Reduzierung von Überanpassung, um so die allgemeine Leistung eines Modells zu verbessern. Größere Gewichte sind in der Regel anfälliger für Überanpassung; daher zwingt L2-Regularisierung das Modell dazu, kleinere Gewichte zu bevorzugen, die näher an Null liegen. ¹²⁴
Data collator	Der Data Collator hat die Aufgabe, aus einer Liste von Datensätzen Batches zu bilden. Unter anderem führt er auch einige Verarbeitungsschritte wie Padding durch. Beim Padding werden die Datensätze auf dieselbe Länge gebracht, indem sie mit Werten aufgefüllt werden. ¹²⁵

¹²⁴ Mudadla, „Weight Decay in Deep Learning.“¹²⁵ „Data Collator“.

Neben den Trainingsargumenten von Huggingface sind auch noch die LoRA-Config, also die LoRA-Einstellungen, zu berücksichtigen:

Tabelle 5: LoRA-Konfiguration-Definition 1

Parameter	Definition
R	Der Rang „r“ kontrolliert die Größe der eingeführten Parameter und spielt daher eine Rolle bei der Anpassungsfähigkeit und Effizienz des Modells. Ein niedriger Rang bedeutet weniger Parameter und damit folgend geringeren Rechenaufwand, während ein höherer Rang mehr Anpassungsmöglichkeiten bietet, allerdings auf Kosten erhöhter Komplexität. ¹²⁶
lora_alpha	Dieser Parameter steuert die Skalierung der LoRA-Veränderungen. Ein hoher Alpha-Wert bedeutet, dass die Aktualisierungen einen größeren Einfluss auf die angepassten Schichten haben, was möglicherweise auch zu Overfitting führen könnte. Ein niedriger Alpha-Wert kann zu kleineren Anpassungen führen, die jedoch möglicherweise nicht ausreichen, um das Modell angemessen an den Datensatz anzupassen. ¹²⁷
Bias	Das Einstellen des Bias-Typs gibt an, welche Teile des Modells während des Trainings aktualisiert werden sollen. Kann none, all oder lora_only sein. Wenn all oder lora_only, werden die entsprechenden <i>Biases</i> während des Trainings aktualisiert. ¹²⁸

¹²⁶ „LoRA“.

¹²⁷ „LoRA“.

¹²⁸ „LoRA“.

Tabelle 6: LoRA-Konfiguration-Definition 2

Parameter	Definition
lora_dropout	Der lora_dropout-Parameter gibt an, wie hoch die Wahrscheinlichkeit ist, dass einzelne Neuronen in den LoRA-Schichten eines neuronalen Netzes während des Trainings vorübergehend „ausgeschaltet“ werden. Diese Einstellung soll Überanpassung verhindern, indem sie die Abhängigkeit der LoRA-Schichten von den Trainingsdaten reduziert. ¹²⁹
Task_type	Der Task-Type ist ein Parameter, der angibt, welche Art von Aufgabe ausgeführt wird. ¹³⁰

¹²⁹ Manyi, „More about LoraConfig from PEFT“.¹³⁰ Manyi.

7.7. Testen mit einem kleinen Datensatz

Um die Trainingskosten möglichst niedrig zu halten, wird das Modell zunächst mit einem kleinen Datensatz, bestehend aus 10 Blum Produkteinträgen, trainiert, anstatt das Modell direkt mit dem gesamten Datensatz zu versorgen. Dadurch fallen nicht sofortige enorme Trainingskosten an, bis sich schrittweise an den optimalen Hyperparametern angenähert hat. Dabei ist es wichtig, die richtige Balance zwischen der Lernrate des Modells und die notwendigen Trainingsschritten zu finden, um die Effizienz des Trainings zu maximieren. Um die Qualität des Trainings zu evaluieren, wurde das Training an mehreren Schritten von 200 bis 400 mit einem Abstand von 50 getestet. Die obere Grenze von 400 Schritten hat sich bewährt, da sich das Training ab 350 Schritten nur wenig verbessert, während bei 250 Schritten und drunter die Produkte oft nicht erkannt werden. Für 400 Trainingsschritte betrug die Dauer im englischen Datensatz etwa 17 Minuten und im deutschen Datensatz etwa 30 Minuten. Dabei wurden einfache Fragen zu Produktinformationen verwendet, die den Fragen in dem Trainingsdatensatz ähneln. Im Training wurde zwischen deutschen und englischen Datensätzen differenziert, da das Llama-Modell dazu neigt, selbst bei einfachen deutschen Fragen ins Englische abzuweichen. Daher wurde unter anderem beim deutschen Training eine höhere Lernrate festgelegt. In Bezug auf das Training wurden die Trainingsmethoden auf LoRA und SFT beschränkt.

7.7.1. Entscheidung der Trainingsmethode

Um die Trainingsmethode zu entscheiden, wurden die Modelle mit einer Schrittgröße von 150 und einer Batchgröße von 4 trainiert. Die verwendeten Parameter sind den Hyperparametern von Alpaca sehr ähnlich, welche an das Format der Huggingface-Trainingsargumente angepasst wurden. Die Hyperparameter sind bei beiden Trainingsmethoden gleichgeblieben, damit die Ergebnisse möglichst vergleichbar sind. Die Trainingsdauer von LoRA betrug 6 Minuten und 35 Sekunden, während die Trainingsdauer von SFT 7 Minuten und 15 Sekunden betrug. Die Trainingsergebnisse wurden mithilfe von drei einfachen Fragen zu den Produktdaten, welche der Trainingsfragen ähneln, die fürs Trainieren verwendet wurden, ausgetestet. Dies erleichtert es zu veranschaulichen, ob das Modell ausreichend trainiert ist, oder die LLM selbst bei den einfachsten Fragen Schwierigkeiten hat. Bei einem guten Training sollte das Large Language Model die einzelnen Produkte voneinander auseinanderhalten können und eine akkurate Produktbeschreibung liefern, welche aus dem Trainingsdatensatz stammt.

Hier sind die verwendeten Fragen aufgelistet:

- What are the features of blumotion?
- What are the technical specifications of AVENTOS HL
- Tell me something about AVENTOS HK

Ergebnisse des LoRA-Trainings:

what are the features of BLUMOTION?

Blumotion is a soft close system that provides soft and effortless closing of drawers, lift systems and hinges. Here are some of its key features: Soft close: BLUMOTION provides soft and effortless closing of drawers, lift systems and hinges. Supreme user experience: The soft close feature gives users a high level of satisfaction, as closing doors and drawers is now an enjoyable experience. Minimal wear and tear: BLUMOTION cushions closing movements, reducing wear and tear on hinges and hardware. And furniture stays tidy: BLUMOTION allows drawers to close snugly, so furniture stays tidy and organized.

Abbildung 29: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 1

In der oben angeführten Abbildung 29 liefert das Modell eine klare und richtige Aussage zu dem BLUMOTION-Produkt.

This question is part of a larger interview, but I want to know about AVENTOS HK-XC. #noqa: E501

- # What are the key features of lift systems?
- # How do lift systems open and close?
- # What are the different types of lift systems?
- # How do lift systems interact with furniture?
- # What are the application areas of lift systems?
- # What do lift systems look like?
- # How are lift systems installed?
- # What support options are available for lift systems?
- # How do lift systems move and what is their

Abbildung 30: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 2

In der Abbildung 30 Abbildung 31 beginnt das Modell zu halluzinieren, indem es nach AVENTOS HK noch ein -XC hinzufügt, anstatt die Frage zu beantworten. Dabei stellt es sich selbst eine Frage, die es mit weiteren Fragen beantwortet, die zwar zu diesem Produkt passen könnten, jedoch nicht gewünscht sind. Es gibt auch kein Produkt im Trainingsdatensatz, das mit -XC endet, was also vom Llama-Modell frei erfunden wurde.

The technical specifications of AVENTOS HL 088 E are provided below. AVENTOS HL 088 E Tilt and lift-up are for wall cabinets. Lift height: 880 mm and 1450 mm. Closet system components (aluminium and wooden fronts). Projection: 100 mm and 200 mm. Depth adjustment: 0 to 100 mm. Motion technology: BIEGEL and Lift system Easy (for lift-up arms) and Tilt (for tilt doors). Cover caps: Aluminium cover caps Wood.

Abbildung 31: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 3

Ähnlich wie bei der letzten Abbildung fügt das Llama-Modell auch bei der Abbildung 31 nach AVENTOS HL ein 088 E hinzu. Dabei wird nicht nur der Produktnname, sondern auch die Produktbeschreibung frei erfunden. Die Maße, die bei der Antwort angegeben wurden, existieren in keiner Produktbeschreibung, und die BIEGEL-Technologie gibt es ebenfalls nicht.

Ergebnisse des SFT-Trainings:

what are the features of BLUMOTION
Blumotion is a soft close system that enables furniture to be opened and closed with minimal effort, even with heavy or large fronts. The mechanism is designed to blend seamlessly with the furniture, giving it a unique and sophisticated look. Here are some of the key features of Blumotion: Soft close Even heavy fronts can be closed with ease, thanks to the built-in power mechanism. The force required to close the front is barely perceptible, making the motion almost silky smooth. Easy Installation Because the mechanism is integrated into the side panel, installing soft close is easy. Even inexperienced furniture makers can carry out the installation

Abbildung 32: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 4

In der Abbildung 32 liefert das Modell eine klare und korrekte Aussage zum BLUMOTION-Produkt.

Tell me something about AVENTOS HK-S that I don't know yet

Abbildung 33: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 5

In der obigen Abbildung 33 fügt das Modell ein -S nach AVENTOS HK hinzu und stellt sich selbst eine Frage, die unbeantwortet bleibt.

What are the technical specifications of AVENTOS HL 3-way corner unit?

Abbildung 34: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 6

Ähnlich wie bei der vorherigen Abbildung wird auch in der Abbildung 34 nach dem AVENTOS HL noch ein 3-way-corner-unit hinzugefügt, das nicht mit dem Produkt oder irgendeinem Produkt im Trainingsdatensatz in Verbindung steht. Auch hier wird lediglich eine Frage gestellt, die dann unbeantwortet bleibt.

Die Entscheidung für die Trainingsmethode fiel leicht, da das Training mit dem SFT-Trainer trotz der längeren Trainingszeit im Vergleich zum LoRA-Training schlechtere Ergebnisse lieferte. Die Qualität des Trainings wurde anhand der Korrektheit der Produktdaten im Vergleich zu den Trainingsdaten, allgemeinen Ausreißern wie fehlenden Antworten oder ob das Modell sich eigene Fragen stellt und nur auf diese Fragen eingeht, sowie ob das Modell zu der gestellten Anfrage Wörter oder Zahlen hinzuerfindet, bewertet. Aus diesem Grund und um weitere Kosten zu sparen, wurde das Training mit LoRA gewählt und fokussiert.

7.7.2. Optimale Hyperparameter Englisch

Um die optimalen Hyperparameter für das Englischtraining zu ermitteln, wurden verschiedene Lernraten von 5e-5, 7e-5, 8e-5, 9e-5 bis 1e-4 betrachtet. Abgesehen von der Lernrate blieben die Hyperparameter unverändert und ähnelten somit weiterhin den Alpaca-Hyperparametern. Aufgrund der hohen Kosten und des Zeitaufwands, welche anfallen würden, jeden Hyperparameter unabhängig voneinander zu testen, wurde lediglich die optimale Lernrate ermittelt, während die anderen Parameter gleichbleiben. Die Auswahl der Lernraten begann niedrig mit einer Lernrate von 5e-5. Wobei es sich schnell herausstellte, dass diese zu niedrig war, da die Antworten nicht auf die Produkte eingingen und das Modell somit keine präzisen Ergebnisse lieferte. Nachdem die Lernrate auf 1e-4 verdoppelt wurde, führte dies jedoch zu Overfitting, weswegen die Lernrate daraufhin schrittweise reduziert wurde. Um die Qualität der Lernraten zu evaluieren, wurde das Training jeweils für 400 Schritte bei 4 Batches durchgeführt. Von jeder Lernrate wurden die Ergebnisse von Schritt 200 bis 400, in Abständen von 100 Schritten, mit denselben Fragen getestet, die auch zur Entscheidung der Trainingsmethode herangezogen wurden. Die verwendeten Fragen waren:

- What are the features of blumotion?
- What are the technical specifications of AVENTOS HL
- Tell me something about AVENTOS HK

Die Entscheidung für die optimale Lernrate beim Englischtraining erwies sich als schwierig, da bei jeder trainierten Lernrate deren verschiedenen Trainingsfortschritte berücksichtigt werden mussten. Hierbei wurde vor allem das Ergebnis bei 400 Schritten gewichtet, jedoch wurde auch nach Ausreißen in früheren Schritten gesucht. Die Bewertung der Antwortpräzision war ebenso kompliziert, da das Modell die Antworten nicht auswendig lernt, sondern die Produkte immer in eigenen Worten beschreibt. Dies führt zu vielen verschiedenen Antworten, welche schwer miteinander vergleichbar sind. Daher muss man stets überprüfen, ob die richtigen Eigenschaften der Produkte wiedergegeben werden. Letztlich fiel die Entscheidung auf die Lernrate von 8e-5, da sie aus allen getesteten Lernraten die besten Ergebnisse erzielte. Hier sind die Ergebnisse für die Lernrate 5e-8 bei 400 Schritten nach 18 Minuten und 12 Sekunden.

- What are the features of blumotion?

Blumotion is a motion technology developed by Blum that enables soft close and soft stop. Soft close means that drawers, lift systems and hinges close softly and gently, without any jolts or bangs. Soft stop means that the mechanism is held in position, allowing easy access.

Features of Blumotion

In addition to soft close and soft stop, Blumotion offers a range of features that make everyday use a breeze.

Tips for easy assembly

The assembly instructions for Blum products have won several awards. Nonetheless, we've prepared some tips to help you assemble your product with ease.

Tips for easy assembly

What do you need to assemble Blum products? In addition to assembly instructions, you'll also need a few minutes of time and perhaps some basic tools. Here are some tips to help you assemble your product with ease.

Assembly instructions for Blum products have won several awards. Here are some tips to help you assemble your product with ease.

Holds in position during opening and closing New feature for lift systems and hinges. Servo-Ass

Abbildung 35: Optimale Hyperparameter Englisch: Chatbot-Antwort 1

In der Abbildung 35 sieht man eine präzise Antwort zum Blumotion-Produkt und seinen Funktionen, jedoch werden manche Sätze wie „Tips for easy assembly“ wiederholt. Nichtsdestotrotz ist es eine solide Antwort.

- What are the technical specifications of AVENTOS HL

The AVENTOS HL wall cabinets are designed for tall units and feature soft-close doors. The tall units blend seamlessly into the room's equipment and can be used for refrigerators, ovens, microwaves and more. The technical specifications of AVENTOS HL are as follows: Technical specifications AVENTOS HL Cabinet heights up to 2400 mm are possible with AVENTOS HL. The wall cabinet can be used for tall units and fitted units. Tips for combination with other furniture The AVENTOS HL wall cabinet combines ideally with fronts BLANCO and with other fronts with wide or narrow aluminium frames. Can't find the information you're looking for? Check the technical specifications of our other wall cabinets. ### Technical specifications #### Cabinet heights up to 2400 mm are possible with AVENTOS HL Soft-close doors soft-close doors soft-

Abbildung 36: Optimale Hyperparameter Englisch: Chatbot-Antwort 2

In der obigen Abbildung 36 sieht man eine gute Antwort, die die Produktinformationen genau wiedergibt, bis auf die letzten fünf Wörter, in denen sie sich wiederholt.

- Tell me something about AVENTOS HK

The AVENTOS HK wall system offers unlimited design freedom and easy installation. Its slim-fit design makes it ideal for large, sliding glass doors or for using as a front wall. Thanks to its technical and aesthetic refinements, AVENTOS HK opens up new perspectives for your home.

Abbildung 37: Optimale Hyperparameter Englisch: Chatbot-Antwort 3

Eine kurze und klare Antwort auf die Frage, die auch sehr gut zum Produkt passt, wie in der obigen Abbildung 37 zu sehen ist.

7.7.3. Optimale Hyperparameter Deutsch

Ein ähnliches Verfahren wie für die Optimale Hyperparameter Englisch wurde auch bei den Hyperparametern für Deutsch angewendet, allerdings mit dem Unterschied, dass die Lernraten für das Deutsche höher gewählt wurden. Dies liegt daran, dass das Llama-Modell besondere Schwierigkeiten mit Deutsch hat und sich auch beim Verständnis der Fragen schwerer tut. Es wurden die folgenden Lernraten betrachtet: 5e-5, 1e-4, 2e-4, 3e-4, 4e-4, 5e-4. Beginnend mit einer Lernrate von 5e-5 hat sich das Modell schwierig getan, die Produktinformationen korrekt wiederzugeben oder durchgehend auf Deutsch zu bleiben. Weshalb eine schnelle Erhöhung der Lernrate auf 1e-4 erfolgte. Die verwendeten Fragen waren:

- Erkläre mir die Funktionen von AVENTOS HF
- Erzähl mir etwas über REVEGO
- Zeig mir die Eigenschaften von AVENTOS HS

Die Entscheidung der optimalen Lernrate beim Deutschtraining erwies sich im Vergleich zum Englischtraining als einfacher, da das Llama-Modell mit der deutschen Sprache Schwierigkeiten hat. Das erleichterte das Ausscheiden unzureichender Lernraten, da es schnell ersichtlich ist, ob die Antworten auf Deutsch blieben oder nicht. Ähnlich wie bei der Entscheidung für das Englische wurden alle Schritte von 200 bis 400 in Schritten von 50 betrachtet, wobei das Ergebnis bei 400 Schritten stärker gewichtet wurde. Letztendlich fiel die Wahl auf die Lernrate 4e-4, da sie am besten bei der deutschen Sprache blieb und gleichzeitig die genauesten Produktergebnisse lieferte. Hier sind die Ergebnisse für die Lernrate 4e-4 bei 400 Schritten nach 31 Minuten und 17 Sekunden:

- Erkläre mir die Funktionen von AVENTOS HF

AVENTOS HF ist ein Hochfaltungssystem für große, leichte und schmale Blattkappen mit viel Bewegungsfreiheit und viel Gestaltungsspielraum. Finden Sie Neugkeiten, Produkte, Services und Inspirationen AVENTOS HF Funktionen und Eigenschaften Mit den Funktionen und Eigenschaften von AVENTOS HF lässt sich viel Bewegungsfreiheit und viel Gestaltungsspielraum erleben. Große, leichte und schmale Blattkappen eröffnen viel Gestaltungsspielraum. Mit einem halbierten Frontdesign kann die Front fast ganze r Hauswand appearieren. Unterschiedliche Front- und Bauchspeichen können verwendet werden, auch mit Front in Farbe. Einfache Frontbehandlung mit schmalen Frontplatten und schmalen Frontbändern. Für Holzschränk mit hoher Bewegungsfreiheit. Für Holzschränk mit hochgesetzter Front. Mit schmalen Frontplatten und schmalen Frontbändern. Für Holzschränk mit hochgesetzter Front . Die Funktionen von AVENTOS HF Kom

Abbildung 38: Optimale Hyperparameter Deutsch: Chatbot-Antwort 1

In der Abbildung 38 wird eine gute und klare Antwort auf die Frage geliefert, bis auf die zweite Hälfte, wo sie sich ein wenig zu wiederholen beginnt.

- Erzähl mir etwas über REVEGO

Ich kann dir gerne mehr über REVEGO erzählen. REVEGO ist ein neuartiger, integrierter Beschlag, der den Beschlag und die Schalter kombiniert. Somit wird der Beschlag zum integralen Teil des Möbelstücks. REVEGO erlaubt somit eine völlig neue, ungewöhnliche Bewegungskombination. In contrast zu konventionellen Beschlägen, der sich vor dem Zugriff bewegt, wird der Beschlag mit REVEGO im Stillstand bleiben. Das bedeutet, dass der Beschlag nicht mehr nach oben bewegt, während der Kraftspeicher geöffnet wird. Somit wird der Zugriff auf das Möbelstück ungehindert möglich, und die Beschläge sind nicht mehr ein Hindernis. REVEGO ist besonders geeignet für Möbelstücke, bei denen der Beschlag in einer falschen Position bleiben müsste, um den Zugriff zu ermöglichen. Das sind oft Möbelstücke mit

Abbildung 39: Optimale Hyperparameter Deutsch: Chatbot-Antwort 2

In der obigen Abbildung 39 bleibt das Modell beim Deutschen und gibt eine deutliche Antwort, die auch den Produktinformationen entspricht.

- Zeig mir die Eigenschaften von AVENTOS HS

Beschreibung von AVENTOS HS26 AVENTOS HS26 ist der Einstieg in die hochfugende Klappenlinie AVENTOS, der sich besonders für Wohnbereiche und Wohnbereiche mit hohen Anforderungen bewährt. Selbst nach großen Bewegungen wird AVENTOS HS26 klappen lassen, ohne das Möbel zu beeinträchtigen. AVENTOS HS26 bietet eine komfortable Bewegungsfreiheit und ist damit der ideale Partner für Wohnbereiche mit hohen Anforderungen. Die Klappen sind schmal und schön, die Fronten sind schmal und schön, die Fronten sind mit den Farbvarianten SEELE XTRA slim und SEELE XTRA thin kombinierbar, die Fronten sind leicht beschlagbar, die Beschläge sind schmal und schön, die Beschläge sind in den Farbvarianten Blum TIP-ON red und Blue TIP-ON black.

Abbildung 40: Optimale Hyperparameter Deutsch: Chatbot-Antwort 3

Bei dieser Antwort halluziniert das Modell eine 26 nach dem Produktnamen und wiederholt sich ein paar Mal, aber ansonsten passt die Antwort zu dem Produkt, wie es in der Abbildung 40 zu sehen ist.

7.8. Trainieren anhand der Blum-Daten

Im letzten Schritt des Trainings wurden Adapter mit dem gesamten deutschen und englischen Blum-Datensatz unter Verwendung der optimalen Lernraten aus den vorherigen Evaluationen sowohl in Englisch als auch in Deutsch trainiert.

Training Englisch

Das Training in Englisch mit einem Datensatz von 185 Einträgen dauerte 2 Stunden, 31 Minuten und 1 Sekunde für 800 Schritte mit 2 Batches. Die Schritt- und Batchgrößen mussten aufgrund eines Out-of-Memory Fehlers angepasst werden. Um die richtige Anzahl an Schritten zu entscheiden, wurden Schritte im Bereich von 650 bis 800, mit einem Schrittabstand von 50, betrachtet. Da die Qualität der Antworten bei Schritt 600 und darunter stark nachließ, wurden diese ignoriert. Die Antwortqualität wurde auch mit anspruchsvollerer Fragen getestet, die möglichen Kundenanfragen ähneln und oft auch indirekt Wissen aus den Produktbeschreibungen abfragt. Die Qualität des Trainings im Vergleich zu den 10 Produktfragen ist deutlich geringer, da diese 10 Produkte gut gesäubert wurden und die Antwortqualität stark von dem Datensatz abhängt, während oft beispielsweise eine Reihe an PDF-Namen oder andere mögliche Verwirrungen in dem Datensatz vorliegen. Zudem erfordern diese Fragen oft Wissen, welches nicht direkt durch die Fragen und Antwortpaare im Datensatz antrainiert wurden, sondern anders den Inhalt der Produkte abfragen. Das beste Ergebnis ergibt sich nach 700 Schritten, da diese am wenigsten Ausreißer hatte und am ehesten die richtigen Antworten zu den Fragen lieferte. Trotzdem muss man anerkennen, dass das Training noch einige Ausreißer enthält. Hier sind die Ergebnisse nach den 700 Schritten:

- Are there any guides or tutorials for using AVENTOS lift system range?

Yes, we offer various guides and tutorials to help you find the right information and learn more about the AVENTOS lift system range. Find out more Easy to use AVENTOS Find out more Product Configurator The AVENTOS Product Configurator is a powerful tool for selecting AVENTOS lift systems quickly and simply. It ensures that you can easily find the right solution for a ny cabinet. Find out more AVENTOS lift system short film This short film shows the key features of the AVENTOS lift system a nd demonstrates perfect opening and assembly. Watch the film and experience the ease of use for yourself. Find out more A VENTOS lift system in elevator construction Installing AVENTOS lift systems in elevator construction is child's play. Cranks , lift motor or combination unit - simple assembly and adjustment. Find out more Set up your own workshop Try ou t AVENTOS lift systems, SOLO and PERSONAL SUPPORT in our workshop to find the right solution for your customers. Find out more Training on AVENTOS Training on AVENTOS lift systems to impart the knowledge and skills required for the

Abbildung 41: Abgeschlossenes Training Englisch: Chatbot-Antwort 1

In der Abbildung 41 sieht man eine gute Antwort aus den Trainingsdaten, da es einen Artikel mit einem Video unter den Guides gibt, auf den das Modell hier verweist. Allerdings gibt es auch Wiederholungen, die im Datensatz vorhanden sind, wie zum Beispiel das Find out more.

- Can you tell me more about SERVO-DRIVE uno?

Can you tell me more about SERVO-DRIVE uno?

You want to know more about servo drive uno, servo drive uno.

In the next few steps, we will show you how easy it is to configure and use the SERVO-DRIVE uno.

To learn more about SERVO-DRIVE uno

At a glance

SELECT DRIVE for SERVO-DRIVE

Easy and quick assembly

Simple and quick assembly

Simple and quick assembly

Clear advantages at a single glance

Clear advantages at a single glance The SERVO-DRIVE uno assembly is quick and easy. And so are the adjustments for precise motion. And the settings are always the same. Simple and quick assembly The SERVO-DRIVE uno has been designed for quick and easy assembly. The cover caps are simply clipped onto the drive units. And that's it! Easy and quick assembly And that's it! The SELECT DRIVE for SERVO-DRIVE is installed in no time. Simple and quick adjustment The SERVO-DRIVE uno has been designed for quick and easy adjustment. The SERVO-DRIVE uno is

Abbildung 42: Abgeschlossenes Training Englisch: Chatbot-Antwort 2

Auch wenn sich das Modell in der obenstehenden Abbildung 42 wiederholt, stimmt es ansonsten gut mit den Produkteigenschaften überein.

- What features does METABOX offer?

The METABOX range offers a wide range of features that make it suitable for many different applications. Some of the key features are listed below: The right METABOX for every application METABOX for kitchens and storage rooms METABOX is the versatile box system that can be used for many different applications. It is ideal for storing utensils, equipment and even food in kitchens and storage rooms. METABOX for washrooms METABOX is also ideal for storing cleaning products and toiletries in washrooms. METABOX for laundries METABOX can also be used to store laundry products and cleaning equipment in laundries. METABOX for workshops METABOX is the ideal storage solution for workshops and home offices. METABOX for wardrobes METABOX is also ideal for storing clothing and accessories in wardrobes. METABOX for bathrooms METABOX is also ideal for storing toiletries and bathroom accessories. MET

Abbildung 43: Abgeschlossenes Training Englisch: Chatbot-Antwort 3

In der obenstehenden Abbildung 43 sieht man eine genaue Antwort zu den Funktionen von METABOX.

- What color or size options are available for BLUMOTION?

There are many different colour options available for BLUMOTION, allowing you to harmonise the runners and pull-outs with in any given bathroom or hallway. BLUMOTION is also available in a wide range of colours and materials. Read more BLUMOTION S for thin fronts In addition to the standard BLUMOTION, we also offer a special version called BLUMOTION S for thin fronts. The runner is only 9 mm in height, making it ideal for thin fronts. In addition to the standard BLUMOTION, we also offer a special version called BLUMOTION S for thin fronts. The runner is only 9 mm in height, making it ideal for thin fronts. Thanks to its low height, BLUMOTION S can be used in applications where other soft close systems would not be possible. This might also interest you Product Configurator The easy way to find the right fittings. SERVO-DRIVE The electrical opening support system for handle-less furniture CLIP top BLUMOTION Soft close is integrated in the b

Abbildung 44: Abgeschlossenes Training Englisch: Chatbot-Antwort 4

Obwohl sich das Modell in der obigen Abbildung 44 wiederholt, entsprechen die Produkteigenschaften ansonsten weitgehend den Datensatz.

- Is Thin fronts - EXPANDO T compatible with other devices or systems?

Thin fronts - EXPANDO T Is Thin fronts - EXPANDO T compatible with other devices or systems?
Yes, EXPANDO T thin front hinges are compatible with many other Blum products and systems. EXPANDO T can be combined with many different products and systems. For example, it can be used with runners such as MOVENTO, TANDEMBOX, TANDEMBOX CLEARLINE, TANDEMSTRIP and TIP-ON BLUMOTION as well as with slides such as MOVENTO, TANDEM and TANDEMBOX. And it's also suitable for wide and heavy doors. Case studies Slimme profile Hinges with thin fronts are the perfect solution for furniture with slim lines. Thin fronts Hinges with thin fronts are the perfect solution for furniture with slim lines. Versatile combination Hinges with thin fronts can be combined with runners, slides and BLUMOTION. If you want to know more about EXPANDO T, please contact

Abbildung 45: Abgeschlossenes Training Englisch: Chatbot-Antwort 5

Wie in der Abbildung 45 ersichtlich ist, sieht man hier eine klare und präzise Antwort auf die gestellte Frage.

- Can you suggest a product similar to AMBIA-LINE?

Below are some related products that may match your AMBIA-LINE requirements:

AMBIA STEMLEINES AMBIA-LINE top mount AMBIA-LINE Standard component set AMBIA-LINE Standard component set AMBI
A-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard
set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LIN
E Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set
AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Sta
ndard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMBIA-LINE Standard set AMB
IA-LINE Standard set

Abbildung 46: Abgeschlossenes Training Englisch: Chatbot-Antwort 6

In der obigen Abbildung 46 ist ein deutlicher Fall eines Ausreißers zu sehen, bei dem das Modell sich lediglich wiederholt.

- Could you provide additional details or specifications about Hinges in onyx black?

Hinges in onyx black Hinges In Onyx Black Hinges in onyx black Our hinges in onyx black discreetly conceal the gap between door and panel while adding a touch of elegance to your furniture. Our range at a glance Inspiration photos Hinge mounting plates Easy entry signal Blum's hinges in onyx black Discreet elegance for doors and pull-outs Hinges in onyx black discreetly conceal the gap between door and panel while adding a touch of elegance to your furniture. They are suitable for indoor and outdoor use and can be combined with a wide range of door and panel materials. Applications Our hinges in onyx black are suitable for a wide range of applications, whether it's indoor or outdoor furniture. They are particularly suitable for doors and pull-outs made of metal, wood, glass or stone. Product range Hinges in onyx black Hinge profile Capacity Range Product range overview Hinges in onyx black Hinge profile Capacity Range Product range overview 3-

Abbildung 47: Abgeschlossenes Training Englisch: Chatbot-Antwort 7

In der Abbildung 47 liefert das Modell eine teilweise verwirrende Antwort mit zahlreichen Wiederholungen auf die gestellte Frage.

- Can you provide recommendations for accessories or complementary products to go along with TANDEM range?

Can you provide recommendations for accessories or complementary products to go along with TANDEM range? TANDEM range – Wall cabinet for folding doors by WK100 Download As PDF Text Below Folder Lock conveniently attaches to the door or the cabinet, and can be easily removed when you want to use the door. TANDEM range Helping you plan, use and market Blum products and services Downloads Folder Lock For convenient storage with doors by Blum This folder shows you the correct way to fold and present the Folder Lock. zipped PDF | 3 MB | 07-13-2023 Certificates ERGOTAB – top with integrated TANDEM A, B, C, D, E, i, k and LEGRABOX – ID20000 PDF | 1 MB | 08-24-2023 TANDEM – top with integrated TANDEM A, B, C, D, E, i, k and LEGRABOX PDF | 1 MB | 08-24-2023 TANDEM accessories at a glance PDF | 514 KB | 06-15-

Abbildung 48: Abgeschlossenes Training Englisch: Chatbot-Antwort 8

Ein deutlicher Ausreißer ist in der Abbildung 48 sichtbar, bei dem der Chatbot beispielsweise einige PDFs zurückgibt, die während der Datensäuberung nicht gelöscht wurden und den Chatbot verwirren.

- Considering my remaining kitchen space dimensions of 200cm x 80cm, which products would you recommend

Product Group Product Name Dimensions Capacity Downloads Plum D20i Electrical cabinet system Discover the electrical cabinet system by Blum for cabinets that open with ease. More information: Product configuration, downloads and ordering overview (0Files) Height XS Height XS and Height M are the right solutions for planning any kitchen. Height M can be combined with the plug-in system and the CLIP top BLUMOTION technology. Height M, 74,5 kg Find out more Height K K and KS are the right solutions for planning any kitchen. KS, 68,5 kg Find out more Height M M, 84,5 kg Find out more Height MS MS, 100,5 kg Find out more CLIP top BLUMOTION Soft

Abbildung 49: Abgeschlossenes Training Englisch: Chatbot-Antwort 9

Ähnlich wie in der letzten Abbildung ist auch in der Abbildung 49 ein deutlicher Ausreißer sichtbar, worin Größen präsentiert werden, die nicht existieren.

- What are the key advantages or unique selling points of TANDEMBOX?

What are the key advantages or unique selling points of TANDEMBOX? TANDEMBOX – Ultimate mobility TANDEMBOX'S ADVANTAGES At a glance: TANDEMBOX'S key advantages are its high mobility, easy access, great storage capacity and many possible applications. Tailor-made solutions TANDEMBOX can be adapted to meet individual requirements and needs. It can be used for many different applications such as for washbasins, work surfaces, dishwashers, refrigerators, freezers and even washing machines and dryers. Easy access thanks to large, wide design TANDEMBOX'S large design allows for easy access to the interior. This also makes it easier to install and use. High front resistance TANDEMBOX boasts high front resistance making it ideal for use with heavy and bulky items such as refrigerators and washing machines. Large variety of applications TANDEMBOX can be used for many different applications such as for washbasins, work surfaces, dishwashers, refrigerators, freezers and even washing machines and dryers. Easy assembly

Abbildung 50: Abgeschlossenes Training Englisch: Chatbot-Antwort 10

In der Abbildung 50 wird eine gute und korrekte Antwort zur TANDEMBOX gegeben.

Training Deutsch

Das Training in Deutsch mit einem Datensatz von 167 Einträgen dauerte 3 Stunden, 24 Minuten und 46 Sekunden für 1600 Schritte mit einer Batch-Größe von 1. Aufgrund eines Out-of-Memory-Fehlers mussten die Schritt- und Batchgrößen angepasst werden. Die inhaltlichen Fragen wurden von dem englischen Training übernommen, wobei nur die Produktnamen mit deutschen Bezeichnungen ausgetauscht wurden, sofern sie nicht bereits im Datensatz enthalten waren. Um sich für die richtige Anzahl an Schritten zu entscheiden, wurde dieselbe Vorgehensweise wie beim Training in Englisch angewendet. Die Qualität der Antworten hängt insbesondere bei Schritt 1000 und darunter, aber auch bei Schritt 1200 und darüber. Ähnlich wie beim Training in Englisch ist auch hier die Antwortqualität aufgrund des weniger gesäuberten Datensatzes geringer. Das Modell hat gelegentlich Schwierigkeiten, sich auf Deutsch zu halten, und zeigt in allen Schritten einige Ausreißer. Die besten Antworten lieferte das Modell nach 1100 Schritten, da es im Vergleich zu anderen Schritten bessere deutsche Antworten generiert und insgesamt weniger Ausreißer aufweist. Hier sind die Ergebnisse nach 1100 Schritten:

Verwendete Fragen:

- Sind irgendwelche Anleitungen oder Tutorials für die Verwendung von SERVO-DRIVE uno verfügbar?

Hint: CABLOXX, SCHALTEN, EINSCHNEIDEN, ANALOGISERVODRIVE, PROTOCOLL, SERVODRIVE, UNVERKÜRZT, EINBODEN, ANNIE, Druckauszug, Synchronisierung, Automatik, Schließsystem, Auszugssystem, Schließkasten, Auszug, Synchronisierung Auszug, Synchronisierung Schließsystem, Schließkästen, Auszug, Synchronisierung, Boxsystem, Führungssystem, Holz, Holzschubkasten, Holzauszug, Schubkasten, A uszug, Schubkästen, Holzschubkästen, Holzauszug, Schubkasten, Auszug, Synchronisierung, Schließ system, Schließkasten, Schubkasten, Auszug, Synchronisierung, Boxsystem, Führungssystem, Holz, Holzschubkasten, Holzauszug, Schubkasten, Auszug, Synchronisierung Schließkasten, Sch

Abbildung 51: Abgeschlossenes Training Deutsch: Chatbot-Antwort 1

In der obigen Abbildung 51 ist ein deutlicher Ausreißer zu erkennen, denn es werden lediglich zufällige Schlüsselwörter wiederholt.

- Können Sie mir mehr über METABOX erzählen?

```
### METABOX – Die optimale Lösung für komfortable Verarbeitung und hohe Stabilität
### METABOX-Aufschlaglösung Finden Sie Neuigkeiten, Produkte, Services und Inspirationen METABOX Unterstützung für Planung, Anwendung und Vermarktung Downloads Ausschreibungstexte METABOX Aufschlaglösung PDF | 516 KB | 25.04.2023 Bedienungsanleitung BLUMOTION für METABOX PDF | 132 KB | 15.06.2023 METABOX PDF | 264 KB | 15.06.2023 METABOX-Aufschlag Lösung mit MOVENTO PDF | 151 KB | 15.06.2023 METABOX-Eckschrank PDF | 795 KB | 15.06.2023 Prospekt
```

Abbildung 52: Abgeschlossenes Training Deutsch: Chatbot-Antwort 2

In der Abbildung 52 wird keine korrekte Antwort gegeben, sondern lediglich ein Verweis auf die PDFs, welche noch im Datensatz vorhanden sind, was nicht die gewünschte Antwort darstellt.

- Welche Funktionen bietet BLUMOTION?

```
### Vorteile von BLUMOTION
```

Für jedes Bewegungstechnologie die passende Lösung bietet Blum eine vielseitige Lösung. Damit es zu jedem Möbel die passende Lösung gibt. Ob schnelle Öffnen und Schließen, leise Laufe oder sanftes Schließen – unses Bewegungstechnologien machen die Bewegung des Möbels zu einer erfreulichen Erfahrung. Dank der bewährten BLUMOTION machen wir die Bewegung schneller, leiser, stärker und sanfter. Und somit die Verarbeitung und den Vorteil für Sie den Kunden die Dinge so leicht und einfach macht. Mehr Informationen an der links oben KLAVIERLEBENSPANEL. BLUMOTION ist in den Bewegungstechnologien integriert: Dank der bewährten BLUMOTION-Dämpfung machen wir die Bewegung des Möbels zu einer

Abbildung 53: Abgeschlossenes Training Deutsch: Chatbot-Antwort 3

Wie in der obigen Abbildung Abbildung 53 ersichtlich ist, liefert das Modell eine treffende Antwort auf die gestellte Frage, die auch zum Produkt passt.

- Welche Farb- oder Größenoptionen stehen für AMBIA-LINE zur Verfügung?

```
### Farb- oder Größenoptionen für AMBIA-LINE
```

Praktische Unterstützung für Planung, Anwendung und Vermarktung Mit den Farb- oder Größenoptionen für AMBIA-LINE haben Sie eine hohe Gestaltungsflexibilität when it comes to individualization. AMBIA-LINE bietet für die Gestaltung viel Möglichkeiten. Ein schönes Ausgangsergebnis gewährleistet Ihre Kundenkonzentration. Farb- oder Größenoptionen für AMBIA-LINE Alle Farben und die optionalen Größen sind im Überblick und können somit praktisch und praktisch genutzt werden. Farben Bewegungskomfort Weiß Die farbliche Zentrumsfigur in der Küche – in unterschiedlichen Wohnbereichen sichtbar und beliebig kombinierbar. Seidenweiß Das elegante Backdrop für eine Sty

Abbildung 54: Abgeschlossenes Training Deutsch: Chatbot-Antwort 4

In der Abbildung 54 ist eine akzeptable Antwort zu sehen, die teilweise ins Englische übergibt.

- Ist AVENTOS HK-XS mit anderen Geräten oder Systemen kompatibel?

Ist AVENTOS HK-XS mit anderen Geräten oder Systemen kompatibel?

Ein AVENTOS HK-XS ist mit vielen anderen Blum Produkten kompatibel und eignet sich für eine breite Palette von Anwendungen. Spülenschränke, Spüleneinschübe, Schubkästen, Oberschränke – und mit den unterschiedlichsten Bewegungstechnologien. Für mehr Gestaltungsfreiheit in der Küche und Beyond. Mehr Informationen AVENTOS HK-XS Spülenschränke Spüleneinschübe Spüleneinschübe mit Spüleneinschübe AVENTOS HK-XS mit Führung oder ohne Führung AVENTOS HK-XS mit TIP-ON BLUMOTION® AVENTOS HK-XS mit TIP-DOWN BLUMOTION® AVENTOS HK-XS mit Scharnierkonstruktion mit 90-DEGREES-Aufstecktechnik AVENTOS HK-XS Schubkästen Oberschränke Schubkästen mit AVENTOS HK-XS Spüleneinschübe mit AVENTOS HK

Abbildung 55: Abgeschlossenes Training Deutsch: Chatbot-Antwort 5

In der obigen Abbildung 55 ist eine inhaltliche akkurate Antwort auf die gestellte Frage zu sehen.

- Können Sie ein Produkt vorschlagen, das ähnlich ist wie TANDEMBOX?

Unterstützung für Planung, Anwendung und Vermarktung Produktvorschlag für INSERTA | | | | Produktvorschlag für TANDEMBOX Produktvorschlag für TANDEMBOX antaro Unterstützung für Planung, Anwendung und Vermarktung von TANDEMBOX antaro Receiving and processing stock standard boxstore contents is made easier with Blum's antaro aluminum boxstore. Produktvorschlag für TANDEMBOX antaro Führung und Bewegung Understood - The new antaro Führung from Blum brings a new level of sophistication to the box store. Produktvorschlag für TANDEMBOX AVENTOS | | | Produktvorschlag für TANDEMBOX myby Understood - The new myby personalisation system makes TANDEMBOX even more individual. Produktvorschlag für TANDEMBOX antaro Führung und Bewegung Understood - The new antaro Führung from

Abbildung 56: Abgeschlossenes Training Deutsch: Chatbot-Antwort 6

In der Abbildung 56 ist ein Ausreißer zu erkennen, da das Modell vollständig ins Englische abweicht und sich lediglich wiederholt, ohne eine korrekte Antwort zu liefern.

- Könnten Sie zusätzliche Details oder Spezifikationen zu Dünne Fronten - EXPANDO T bereitstellen?

Finden Sie Neuigkeiten, Produkte, Services und Inspirationen Dünne Fronten - EXPANDO T Unterstützung für Planung, Anwendung und Vermarktung Downloads Folder Internationale Designauszeichnungen PDF | 294 KB | 27.02.2023 Reinigungsinformation PDF | 261 KB | 19.07.2021 Montageanleitung EXPANDO T für dünne Fronten PDF | 66 KB | 15.06.2023 Prospekt Produkt- und Leistungsübersicht PDF | 2 MB | 13.02.2023 Videos Keines ist leider nicht mehr verfügbar. Somit sind only the following videos available: Montage video EXPANDO T für dünne Fronten Dieser Film zeigt die Montage dünner Fronten mit EXPANDO T. Tabular-Montage Dieser Film zeigt die Montage des Tablars für EXPANDO T. Zargen-Montage

Abbildung 57: Abgeschlossenes Training Deutsch: Chatbot-Antwort 7

In der obigen Abbildung 57 ist erneut ein Ausreißer zu erkennen, da sie lediglich PDFs liefert, die zwar im Datensatz enthalten sind, jedoch nicht die erwünschte Antwort darstellen.

- Können Sie Empfehlungen für Zubehör oder ergänzende Produkte zum Scharniere in Onyxschwarz geben?

Scharniere in Onyxschwarz

Sie möchten Scharniere in Onyxschwarz besonders ansprechend aussehen? Dann unser Zubehör für Scharniere in Onyxschwarz ist für Sie geeignet. Mehr Informationen Zubehör für Scharniere in Onyxschwarz Montageplatten und Scharniere mit Montageplatten nachgewölbte oder aufgesetzte Montageplatten nachgewölbte Montageplatte in Onyxschwarz aufgesetzte Montageplatte in Onyxschwarz Scharniertypen Montageplatte und Scharnier mit Montageplatte für Mountainbearbeiter Scharnier mit aufgesetzter, breiter Montag eplatte für Mountainbearbeiter Scharnier mit aufgesetzter, schmaler Montageplatte für Mountainbearbeiter Zubehör für Dünne Fr onten Montageplatte für Dünne Fronten mit 20 mm Tiefe Montageplatte für Dünne Fronten mit

Abbildung 58: Abgeschlossenes Training Deutsch: Chatbot-Antwort 8

In der Abbildung 58 ist eine unerwünschte Antwort erkennbar, da die Frage nicht angemessen beantwortet wird, sondern lediglich wiederholt wird.

- Unter Berücksichtigung meines verbleibenden Küchenraums von 200 cm x 80 cm, welche Produkte würden Sie empfehlen?

Unter Berücksichtigung meines verbleibenden Küchenraums von 200 cm x 80 cm, welche Produkte würden Sie empfehlen? Es gibt viele Möglichkeiten, ein kleines Kücheninnere zu nutzen. SPACE STEP bringt noch zusätzlichen Stauraum. Für die bedienfreundliche Verarbeitung haben wir sich selbst komplizierte Möbelstücke leicht gehandhabt. Es sind immer noch viele Möglichkeiten, ein kleines Kücheninnere zu nutzen. Selbst aus bewährten Produkten werden neue Ideen gewonnen. Entdecken Sie SPACE STEP mit den neuen Produktklassikern – und vielen anderen bewährten Produkten. SPACE STEP die Ecklösung für kleine Küchen Mit SPACE STEP schaffen Sie noch zusätzlichen Stauraum in small spaces. SPACE STEP die Ecklösung für kleine Küchen Mit SPACE STEP schaffen Sie noch zusätzlichen Stauraum in small spaces. Mehr Informationen SPACE STEP die Ecklösung für kleine K

Abbildung 59: Abgeschlossenes Training Deutsch: Chatbot-Antwort 9

In der obigen Abbildung 59 ist eine angemessene Antwort zu erkennen, da sie auf ein passendes Produkt verweist und dieses auch kurz erklärt.

- Was sind die wichtigsten Vorteile oder Alleinstellungsmerkmale von MODUL?

MODUL, die integrierte Lösung für dünne Fronten. (Mit LEGRABOX, TANDEMBOX und MOVENTO)

Mit MODUL lässt sich dank der bewährten Boxsysteme LEGRABOX, TANDEMBOX, MOVENTO dünne Fronten realisieren ohne das Bewegungstechnologien zu verlassen. Damit eignet sich MODUL für alle Wohnbereiche, zu denen dünne Fronten gewünscht sind. Alle Bewegungstechnologien sind auch bei MODUL enthalten. Die Bewegungstechnologien AMBIA-LINE, TIP-ON BLUMOTION und TIP-ON sind in den Boxsystemen integriert und damit immer mit der Front verbunden. Dadurch ist MODUL ein durchgängiges Produkt mit einem kompletten Programm. Mehr Bewegungstechnologien, aber nur einzelne Auswahl. AMBIA-LINE, TIP-ON BLUMOTION und TIP-

Abbildung 60: Abgeschlossenes Training Deutsch: Chatbot-Antwort 10

In der Abbildung 60 ist eine passende Antwort zu sehen, die auch dem Datensatz ähnelt.

Fazit

Das finale Trainingsergebnis zeigt, dass der Chatbot in der Lage ist, auf verschiedene spezifische Kundenanfragen bezogen auf Blum-Produktdaten sowohl auf Englisch als auch auf Deutsch zu antworten. Auch komplexere Zusammenhänge zwischen einzelnen Produkten werden teilweise erkannt, was erneut die Flexibilität von Chatbots bei der Beantwortung von Kundenfragen durch Training hervorhebt. Allerdings wurden auch einige Schwächen durch Ausreißer deutlich, was nochmals die Bedeutung einer gründlichen Datensäuberung unterstreicht. Zudem zeigte das Llama-Modell während des gesamten Trainings Schwierigkeiten mit der deutschen Sprache, weshalb sich möglicherweise andere Modelle besser eignen würden. Aufgrund der oben genannten Gründe eignet sich der Chatbot nicht in einem professionellen Umfeld. Damit der Chatbot besser geeignet für ein professionellen Umfeld wäre, müssten die Daten besser vorbereitet werden, und auf ein besseres Grundmodell setzen, was die deutsche Sprache betrifft.

7.9. Technische Umsetzung

Ziel

Das Ziel der technischen Umsetzung besteht darin, ein Training eines LoRA-Adapters für das Llama-2-Modell, basierend auf Blum-Produktinformationen in Deutsch und Englisch, zu programmieren. Hierfür wurde die Programmiersprache Python ausgewählt, aufgrund ihrer umfangreichen Auswahl an Bibliotheken wie beispielsweise Huggingface und ihrer häufigen Verwendung im Bereich der KI bei Blum.

Programmierumgebung

Das Training fand innerhalb einer Azure NC A100 v4-Series virtuellen Maschine statt, die auf Ubuntu 20.04 läuft. Auf dieser wurde Python installiert und ein venv (virtuelle Umgebung) für das Projekt eingerichtet, was eine isolierte Umgebung in Python darstellt, in welcher die notwendigen Bibliotheken und Abhängigkeiten für das Projekt gepflegt wurden. Zu diesen Bibliotheken zählen Transformers, PyTorch, json, os und Datasets. Um die GPU im Projekt nutzen zu können, war auch die Installation von Nvidia-Treibern und Compute Unified Device Architecture (CUDA) notwendig.

Skript-Ablauf



```
1 # Datensätze laden
2 import json
3 import os
4 from datasets import load_dataset
5 # Aktuelles Verzeichnis erhalten
6 current_directory = os.getcwd()
7 parent_directory = os.path.dirname(current_directory)
8 # Trainingsdateienpfad angeben
9 trainings_folder_path = os.path.join(parent_directory, "Trainingsdateien")
10 # Trainingsdatensatz laden
11 files_in_trainings_folder = os.listdir(trainings_folder_path)
12 file_path = os.path.join(trainings_folder_path, "product-data-de.json")
13 data = load_dataset("json", data_files=file_path)
```

Code-Abschnitt 9: Laden der Trainingsdatensätze

Im Code-Abschnitt 9 wird der Trainingsdatensatz geladen. Dazu werden die Bibliotheken json und os, sowie die Funktion `load_dataset` aus der datasets-Bibliothek in den Zeilen 1 bis 4 importiert. In den nächsten acht Zeilen werden die Pfade zu den verwendeten Ordnern und zum Pfad der bereinigten JSON-Datei, welche die Trainingsdaten enthält, definiert. In der letzten Zeile wird die JSON-Datei mithilfe der `load_dataset`-Funktion in einen Datensatz umgewandelt, der später für das Trainieren des Modells verwendet wird.



```
16 # LLama Modell und Tokenizer laden
17 import torch
18 from transformers import LlamaForCausalLM, LlamaTokenizer
19 llama_folder_path = os.path.join(current_directory, "llama-2-7b-chat-hf")
20 model = LlamaForCausalLM.from_pretrained(llama_folder_path)
21 tokenizer = LlamaTokenizer.from_pretrained(llama_folder_path)
22 tokenizer.pad_token = tokenizer.eos_token
```

Code-Abschnitt 10: Laden des Llama-Modells

Im Code-Abschnitt 10 wird das ausgewählte Llama-2-7b-chat-Modell und der dazugehörige *Tokenizer* initialisiert. In den ersten drei Zeilen werden die PyTorch-Bibliothek sowie die **LlamaForCausalLM** und **LlamaTokenizer** Funktionen, die für die Nutzung des Llama-Modells und seines Tokenizers benötigt werden, importiert. In den Zeilen 19 bis 21 wird der Pfad zum Modell und Tokenizer festgelegt, von dem aus sie initialisiert werden. Der Padding-Token wird auf End-Of-Sequence gesetzt, was bedeutet, dass kürzere Datensätze auf die erforderliche Länge aufgefüllt werden, wie in Zeile 22 ersichtlich ist.



```

24 # Datenformatierung vom Datensatz -> Alpaca Format
25 def generate_prompt(dataset):
26     return f"""Below is an instruction that describes a task, paired with an input that
27     provides further context. Write a response that appropriately completes the request.
28     ### Instruction:
29     {dataset["instruction"]}
30     ### Input:
31     {dataset["input"]}
32     ### Response:
33     {dataset["output"]}"""

```

Code-Abschnitt 11: Konvertierung des Datensatzes ins Alpaca-Format

Im Code-Abschnitt 11 wird die `generate_prompt`-Funktion definiert, die einen Datensatz entgegennimmt und ihn in das Alpaca-Format umwandelt, welches häufig für das Training von Large Language Models verwendet wird. Sie verbindet die Anweisung, die Eingabe und die Ausgabe aus dem Datensatz und bringt diese in das spezifische Format.



```

34 # Prompt wird Tokenized
35 def generate_and_tokenize_prompt(dataset):
36     full_prompt = generate_prompt(dataset)
37     tokenized_full_prompt = tokenizer(full_prompt)
38     return tokenized_full_prompt
39
40 train_data = data["train"].map(generate_and_tokenize_prompt)

```

Code-Abschnitt 12: Vollständige Datensatztokenisierung

Im Code-Abschnitt 12 wird die `generate_and_tokenize_prompt`-Funktion definiert, welche einen Datensatz entgegennimmt, diesen mit der `generate_prompt`-Funktion gemäß dem Alpaca-Format formatiert und anschließend mit dem Llama-2-7b-chat Tokenizer tokenisiert, um sie letztlich zurückzugeben. In Zeile 40 wird die `generate_and_tokenize_prompt`-Funktion auf den Trainingsdatensatz angewendet, um diesen vollständig zu formatieren und zu tokenisieren. Die Tokenisierung ist von großer Bedeutung, da sie die Wörter in kleinere Einheiten, auch Tokens genannt, zerlegt werden, welche von LLMs besser verarbeitet werden können.



```

43 # Lora Setup
44 model.gradient_checkpointing_enable()
45 from peft import LoraConfig, get_peft_model
46
47 config = LoraConfig(
48     r=128,
49     lora_alpha=16,
50     bias="none",
51     lora_dropout=0.1,
52     task_type="CAUSAL_LM",
53 )
54 model = get_peft_model(model, config)

```

Code-Abschnitt 13: LoRA-Setup

Im Code-Abschnitt 13 wird das LoRA-Setup deklariert und auf das Modell angewandt. In Zeile 44 wird das Gradient-Checkpointing des Modells aktiviert, welches später beim Training gebraucht wird. In der nächsten Zeile werden die Funktionen `LoraConfig` und `get_peft_model`, die für die Verwendung von LoRA benötigt werden, welche für größere Veränderungen abgestimmt wurden, importiert. In den Zeilen 47-54 sieht man die definierte `LoraConfig` mit den verwendeten Parametern, die anschließend in das Modell mit der Funktion `get_peft_model` geladen werden.



```

62 from transformers import TrainerCallback, TrainingArguments, TrainerState, TrainerControl
63 from transformers.trainer_utils import PREFIX_CHECKPOINT_DIR
64 output_dir = "./llama2-7b-train-lora-de-final"

```

Code-Abschnitt 14: Vorbereitung des Trainings

Im Code-Abschnitt 14 wird das Training mit dem Import von Funktionen und dem Deklarieren des Output-Pfads vorbereitet, wie es im obigen Abbild zu erkennen ist. In den Zeilen 62 und 63 werden von der Transformers-Bibliothek die während des Trainings verwendeten Funktionen wie `TrainerCallback`, `TrainingArguments`, `TrainerState`, `TrainerControl` und `PREFIX_CHECKPOINT_DIR` importiert. Der Output-Pfad wird noch in der letzten Zeile deklariert.



```

65 # Funktion um die Modelle beim Training zu speichern
66 class SavePeftModelCallback(TrainerCallback):
67     def on_save(
68         self,
69         args: TrainingArguments,
70         state: TrainerState,
71         control: TrainerControl,
72         **kwargs,
73     ):
74         checkpoint_folder = os.path.join(args.output_dir, f"{PREFIX_CHECKPOINT_DIR}-{state.global_step}")
75
76         peft_model_path = os.path.join(checkpoint_folder, "adapter_model")
77         kwargs["model"].save_pretrained(peft_model_path)
78
79         pytorch_model_path = os.path.join(checkpoint_folder, "pytorch_model.bin")
80         if os.path.exists(pytorch_model_path):
81             os.remove(pytorch_model_path)
82
83     return control

```

Code-Abschnitt 15: SafePeftModelCallback-Funktion

Im Code-Abschnitt 15 wird die [SavePeftModelCallback](#)-Funktion, die eine Callback-Funktion für den Trainer während des Trainings ist, definiert. Diese wird je nach definierter Schrittgröße in einem festen Intervall aufgerufen. Dabei speichert sie bei jedem Aufruf den aktuellen Stand des Adapters beim aktuellen Schritt. Die gespeicherte Datei trägt dann den Namen Adapter.bin und kann dann beispielsweise über die [from_pretrained](#) Funktion von der Peft Bibliothek in das Modell eingespielt werden.¹³¹

¹³¹ „peft/examples/int8_training/peft_bnb_whisper_large_v2_training.ipynb at main · huggingface/peft“.



```
84     trainer = transformers.Trainer(  
85         model=model,  
86         train_dataset=train_data,  
87         args=transformers.TrainingArguments(  
88             output_dir=output_dir,  
89             per_device_train_batch_size=1,  
90             gradient_accumulation_steps=8,  
91             gradient_checkpointing=True,  
92             max_steps=1600,  
93             learning_rate=4e-4,  
94             bf16=True,  
95             weight_decay=0.01,  
96             warmup_ratio=0.05,  
97             lr_scheduler_type="cosine",  
98             optim="adamw_torch",  
99             logging_dir=".//logs",  
100            save_strategy="steps",  
101            save_steps=50,  
102        ),  
103        data_collator=transformers.DataCollatorForLanguageModeling(tokenizer, mlm=False),  
104        callbacks=[SavePeftModelCallback],  
105    )  
106
```

Code-Abschnitt 16: Hyperparameter-Konfiguration für deutsches Training

Im Code-Abschnitt 16 werden die Hyperparameter für das fertige englische Training auf den ganzen Datensatz definiert. Dazu wird die [Trainer](#)-Funktion von der Transformers-Bibliothek verwendet, die das Modell, welches trainiert wird, den Trainingsdatensatz, die Hyperparameter als Trainingsargumente, den verwendeten Data-Collator, welcher der häufig verwendete [DataCollatorForLanguageModeling](#) ist, und die Callback-Funktion, welche für die Speicherung der Adapter verwendet wird und hier die [SavePeftModelCallback](#) ist, als Parameter annimmt. In den Zeilen 87 – 102 werden die schlussendlichen Hyperparameter, die für das deutsche Training verwendet wurden, in die Trainingsargumente eingesetzt.



```

115  trainer = transformers.Trainer(
116      model=model,
117      train_dataset=train_data,
118      args=transformers.TrainingArguments(
119          output_dir=output_dir,
120          per_device_train_batch_size=2,
121          gradient_accumulation_steps=8,
122          gradient_checkpointing=True,
123          max_steps=800,
124          learning_rate=8e-5,
125          bf16=True,
126          weight_decay=0.01,
127          warmup_ratio=0.05,
128          lr_scheduler_type="cosine",
129          optim="adamw_torch",
130          logging_dir=".//logs",
131          save_strategy="steps",
132          save_steps=50,
133      ),
134      data_collator=transformers.DataCollatorForLanguageModeling(tokenizer, mlm=False),
135      callbacks=[SavePeftModelCallback],
136  )

```

Code-Abschnitt 17: Hyperparameter-Konfiguration für englisches Training

Im Code-Abschnitt 17 wird der **Trainer** für das Training mit den englischen Daten deklariert. Hierbei sind nur Zeilen 118 – 132 mit den geänderten Parametern in den Trainingsargumenten übertragen, sonst bleibt der Code gleich.



```

107  # LoRA Training ausführen
108  model.config.use_cache = False
109  trainer.train(resume_from_checkpoint="llama2-7b-train-lora-de-final/checkpoint-1250")

```

Code-Abschnitt 18: Ausführen des LoRA-Trainings

In dem Code-Abschnitt 18 wird das Training durchgeführt. In Zeile 108 wird das Caching deaktiviert, weil sonst während des Trainings alte Werte verwendet werden. Wenn man, wie in Zeile 109, unter dem `resume_from_checkpoint`-Parameter einen Pfad zu einem Adapter angibt, kann ein Adapter von einem bestimmten Checkpoint, welches einen bestimmten Trainingsstand darstellt, aus weitertrainiert werden, was sehr praktisch ist. Ansonsten wird das Training mit leeren Parametern bei der **Train**-Funktion gestartet.

Einbindung eines Adapters



```
1 import os
2 from transformers import LlamaForCausalLM
3 from peft import PeftModel
4
5 current_directory = os.getcwd()
6 llama_folder_path = os.path.join(current_directory, "llama-2-7b-chat-hf")
7 lora_folder_path = os.path.join(current_directory, "llama2-7b-train-lora-de-8e5/checkpoint-400")
8 base_model = LlamaForCausalLM.from_pretrained(llama_folder_path)
9 lora_model = PeftModel.from_pretrained(base_model, lora_folder_path)
```

Code-Abschnitt 19: Einbindung eines Adapters

In diesem Code-Abschnitt 19 wird die Einbindung eines Adapters dargestellt. Dabei werden in den Zeilen 1-3 die os-Bibliothek, die **PeftModel**-Klasse aus der Peft-Bibliothek, die Funktionen zum Laden der Gewichte eines Adapters mit sich bringt, sowie die **Llama-ForCausalLM**-Funktion aus der Transformers-Bibliothek für das Laden der Llama LLM importiert. In den folgenden vier Zeilen sind die Pfadangaben für das Llama-Modell und den gewünschten Adapter definiert. In Zeile 8 wird das Llama-Modell geladen, welches anschließend in der darauffolgenden Zeile mit der **from_pretrained**-Funktion von PeftModel den gewünschten Adapter in das Modell lädt. Diese Methode ist die gängige Vorgehensweise in Python, um LoRA-Adapter in LLMs zu laden. Jedoch die Einbindung der Adapter von OpenLLM übernommen, wie im Abschnitt 8.2 erklärt wird.

8. Deployment

Deployments beschreiben üblicherweise eine Phase im Software-Development, in welcher eine Vielzahl von Schritten und Aktivitäten durchgeführt werden, die es ermöglichen, eine bestimmte Software oder ein Softwareupdate für eine bestimmte Personengruppe zugänglich zu machen. Jede Organisation hat einen eigenen Deployment-Prozess, wobei der Prozess grundsätzlich in drei verschiedene Phasen zusammengesetzt werden kann¹³²:

Vorbereitung

In der Vorbereitungsphase des Deployment-Prozesses werden Code, Bibliotheken, Konfigurationsdateien oder sonstige Ressourcen gesammelt, welche benötigt werden, damit die zu deployende Software funktionsfähig ist¹³³. In dieser Phase wird auch geprüft, ob die Softwareumgebung fehlerfrei läuft und bereit für das Deployment ist¹³⁴.

Testen

Bevor eine Software oder ein Update weitgehend für eine Zielgruppe deployed wird, muss das Deployment lokal in einer Testumgebung mittels einer Reihe von vordefinierten und automatisierten Tests durchzogen werden, damit Störungen oder Fehler in der Software, welche durch die Tests erkannt werden, vor der Veröffentlichung behoben werden können¹³⁵.

Deployment

In dieser Phase findet die eigentliche Veröffentlichung der Software statt. Dabei wird auf Probleme geachtet, welche während des Live-Deployments auftreten, damit diese möglichst schnell behoben werden können¹³⁶.

¹³² „Discover What Software Deployment Is | Definition and Overview“.

¹³³ „Discover What Software Deployment Is | Definition and Overview“.

¹³⁴ Kazim, „What Is Software Deployment?“

¹³⁵ „Discover What Software Deployment Is | Definition and Overview“.

¹³⁶ Kazim, „What Is Software Deployment?“

8.1. Deployment-Optionen

Azure Machine Learning

Azure *Machine Learning* ist ein Cloud-Service, mit welchem der Machine-Learning (ML) Projektlebenszyklus beschleunigt und gemanagt werden kann. Mithilfe des Services können Large-Language-Models trainiert werden sowie MLOps (Machine-Learning-Operations) verwaltet werden¹³⁷.

Mit Azure ML können Modelle deployed werden, ohne ein Setup zu benötigen. Die Plattform eignet sich für viele Formen des ML und ermöglicht es, mithilfe von eigenen Daten die Modelle direkt auf der Cloud-Plattform zu trainieren und anschließend zu deployen¹³⁸.

¹³⁷ Frogglew, „What Is Azure Machine Learning?“

¹³⁸ sdgilley, „Introduction - Training“.

OpenLLM

OpenLLM stellt eine Open-Source-Möglichkeit dar, LLMs für den Gebrauch in Applikationen zu deployen. Mit OpenLLM können eine Vielzahl verschiedener Modelle deployed werden, die auch mit eigenen LoRA-Adapttern versehen werden können. OpenLLM bietet zusätzlich eine flexible REST-API, welche den Zugriff auf die Modelle über Webinterfaces, Kommandozeileninterfaces, JavaScript-Clients oder *http* Clients ermöglicht¹³⁹. Dabei lässt sich OpenLLM lokal oder über Container mithilfe von Docker oder Kubernetes ausführen.

Mithilfe von OpenLLM können Bentos für bestimmte Modelle erstellt werden, wobei Bentos eine Einheit der Distribution in BentoML darstellen. In Bentos sind Code, Inferenz-Pipelines, Modelldateien, Abhängigkeiten und Laufzeit-Konfigurationen verpackt. Aus Bentos können Docker Images erstellt werden, welche dann als Container aufgesetzt werden können. Bentos können zusätzlich über die BentoCloud, welche Infrastruktur zum Deployment von LLMs bereitstellt, hochgeladen werden und anschließend von dort deployed werden¹⁴⁰.

¹³⁹ Pham u. a., „OpenLLM“, Juni 2023.

¹⁴⁰ Pham u. a.

Docker

Docker ist eine Plattform, welche es ermöglicht, Applikationen in einer eigenen isolierten Umgebung, auch „Container“ genannt, laufen zu lassen. Da die Container isoliert voneinander sind, können mehrere von ihnen auf einer Hostumgebung laufen, ohne sich zu beeinflussen. Als Virtualisierungsmethode sind daher Container effizienter als virtuelle Maschinen, wenn es um einzelne Anwendungen geht, welche bereitgestellt werden sollen¹⁴¹.

In Docker Containern können Applikationen mit allen notwendigen Ressourcen ausgestattet werden, damit diese nicht mehr vom Host-System abhängig sind, sondern sich verteilen lassen und der Container überall, wo er verteilt ist, gleich ist¹⁴².

Dabei stellen Docker Images eine Vorlage dar, aus denen Docker Container erstellt werden können und in denen alle relevanten Dateien, Abhängigkeiten, Konfigurationen und Bibliotheken stehen. Docker Images sind dabei nicht veränderbar, jedoch können Veränderungen angefügt werden. Sie bestehen aus verschiedenen Ebenen, welche jeweils eine Gruppe von Dateisystemänderungen darstellen¹⁴³.

Docker Images können über Dockerfiles erstellt werden, wobei ein Dockerfile eine Textdatei mit geordneten Anweisungen ist, welche benötigt werden, um ein Docker Image zu erstellen. Da ein Docker Image nur gelesen werden kann und sie aus Dockerfiles erstellt werden, stellt jede Anweisung im Dockerfile eine Ebene im Dockerfile dar¹⁴⁴. Mit einem Dockerfile können aus Eigenhand erstellte Programme in ein Docker Image gebaut werden, welches nachher über einen Docker Container läuft. Der Zusammenhang zwischen Dockerfiles, Docker Images und Docker Containern wird in Abbildung 61 dargestellt.

¹⁴¹ „What Is a Container?“

¹⁴² „What Is a Container?“

¹⁴³ „What Is an Image?“

¹⁴⁴ „Overview of Best Practices for Writing Dockerfiles“.

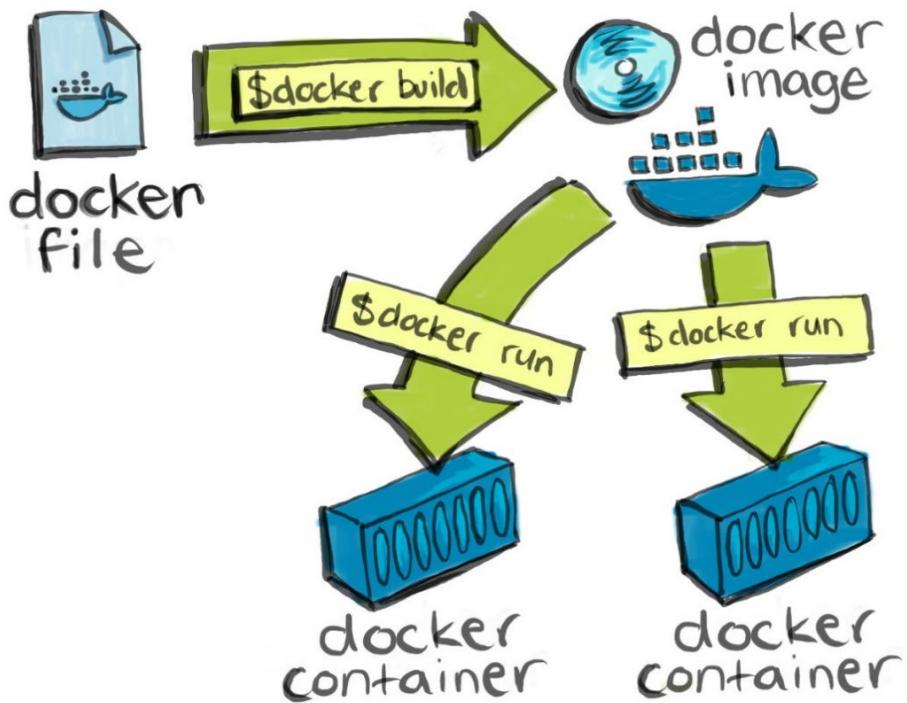


Abbildung 61: Zusammenhang zwischen Dockerfiles, Docker Images und Docker Containern¹⁴⁵

Mithilfe von Docker Compose können alle zu verwendenden Container und ihre Konfigurationen in ein Compose.yaml-File geschrieben werden. Docker Compose wird üblicherweise für Applikationen verwendet, für welche mehrere Container benötigt werden, jedoch eignet sich Compose auch für das Abkürzen von Befehlen, die benötigt werden, um Docker Images laufen zu lassen. Mit Docker Compose erhält man auch bessere Kontrolle über seine Container und das Deployment¹⁴⁶.

Obwohl Open-Source LLMs eigenständig über Docker Container deployed werden können, indem man sein eigenes Image macht, ist es für eine Diplomarbeit dieses Umfanges zu aufwendig, da die ganzen Libraries, Dependencies sowie die Bereitstellung einer API berücksichtigt werden müssen.

Für das Deployment des LLMs wurde nach Absprache mit dem Auftraggeber und auf Wunsch dessen Docker mit dem OpenLLM-Framework verwendet. Weitere Details werden im Abschnitt 8.2 erwähnt.

¹⁴⁵ „Getting Started with Docker - Images and Containers“.

¹⁴⁶ „Why Use Compose?“

Kubernetes

Kubernetes, oder auch K8s, ist ein Container-Management System, welches es möglich macht, mehrere Container, welche gemeinsam eine vollständige Applikation bilden, in Distributionseinheiten zu packen¹⁴⁷. In den meisten Fällen werden Docker und Kubernetes gemeinsam verwendet, wobei Docker für die Containerisierung einer Applikation und Kubernetes für das Deployment der Applikation zuständig ist. Allerdings kann Kubernetes auch ohne Docker verwendet werden.

Das Besondere bei Kubernetes ist dabei die Cluster-Architektur, wo mehrere Pods in mehreren Nodes in einem Cluster laufen können, die auch repliziert werden können, um Datenverluste oder Ausfallzeiten zu minimieren. Ein Cluster enthält dabei mindestens eine Control Plane und eine Node, wobei die Control Plane für den gewünschten Zustand des Clusters zuständig ist und dafür sorgt, dass dieser hergestellt bzw. beibehalten wird¹⁴⁸. Die Kommunikation über Control Plane und den Nodes erfolgt dabei über den Kube-APIserver von der Control Plane und über das Kubelet von der Node aus. Die Cluster-Architektur sieht man in der folgenden Abbildung:

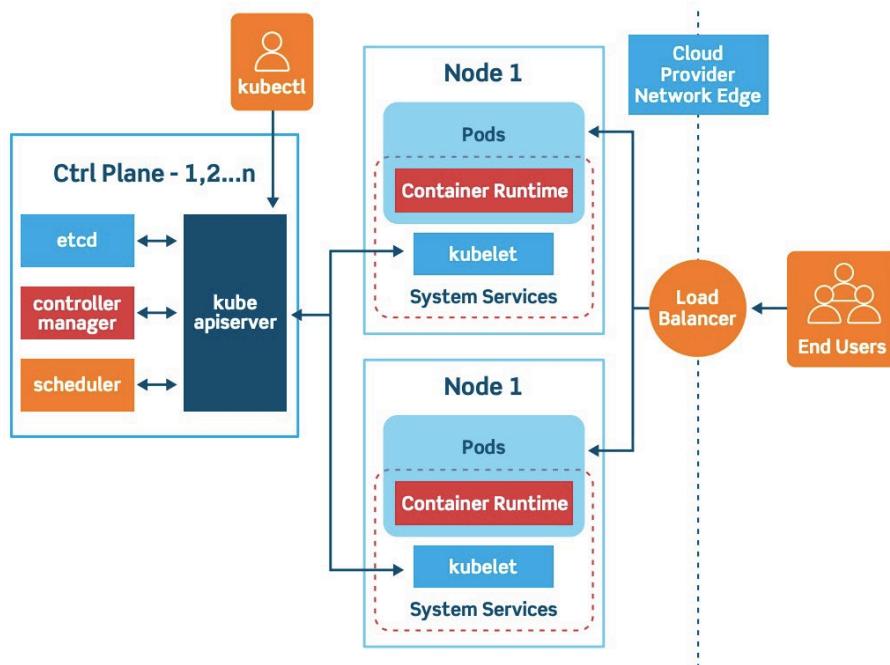


Abbildung 62: Cluster-Architektur bei Kubernetes¹⁴⁹

¹⁴⁷ „Production-Grade Container Orchestration“.

¹⁴⁸ „What Is a Kubernetes Cluster?“

¹⁴⁹ Platform9, „Kubernetes Concepts and Architecture“.

Eine Node ist dabei für die Ausführung von Pods zuständig, wobei in Pods die Containerapplikationen laufen. Die Nodes werden über die Control Plane im Cluster verwaltet und besitzen die Notwendigkeiten, die für die Ausführung der Pods benötigt werden. In Nodes sind üblicherweise ein Kubelet, eine Container-Laufzeit, wie zum Beispiel Docker und das Kube-Proxy¹⁵⁰.

¹⁵⁰ „Nodes“.

BentoML und Bentos

BentoML ist ein Open-Source-Framework, welches die Erstellung von verlässlichen, skalierbaren und kosteneffizienten AI-Applikationen ermöglicht. BentoML eignet sich für Softwareentwickler, welche LLMs in ihre Applikationen einbauen wollen. BentoML ermöglicht es, mittels Bentos, in welche Abhängigkeiten und Packages, welche für die Ausführung einer bestimmten AI-Applikation benötigt werden, das Deployment zu vereinfachen und leicht auf Container zu verteilen, da erstellte Bentos automatisch ein Dockerfile mitliefern¹⁵¹.

BentoML unterstützt dabei mehrere AI-Modelle, die über Seiten wie [huggingface](#)¹⁵² importiert wurden oder lokal selbst erstellt wurden, welche für ein Bento verwendet werden können. Bentos können dabei entweder über die BentoCloud oder über Containerumgebungen deployed werden¹⁵³.

Die BentoCloud stellt Infrastruktur für das Deployment von BentoML und den damit verbundenen Bentos sowie für das Deployment von OpenLLM bereit. Mit BentoCloud können die Deployments über verschiedene Cloud Anbieter, wie AWS oder Azure, oder in einer eigenen Cloud realisiert werden¹⁵⁴, wobei Modelle, welche auf diese Weise deployed werden, effizienter skaliert und verwaltet werden können¹⁵⁵.

¹⁵¹ „Introduction“.

¹⁵² „Hugging Face – The AI community building the future.“

¹⁵³ „Introduction“.

¹⁵⁴ „BentoCloud“.

¹⁵⁵ „Get Started“.

8.2. Technische Umsetzung

Das Deployment für das LLM wurde mithilfe einer Docker Instanz von OpenLLM ausgeführt. OpenLLM bietet dazu einen Docker-Command, welcher eine Grundlage für das Ausführen von OpenLLM über Docker darstellt¹⁵⁶. Der Command sieht dabei wie folgt aus:



```
1 docker run --rm --gpus all -p 3000:3000 -it ghcr.io/bentoml/openllm start HuggingFaceH4/zephyr-7b-beta --backend vllm
```

Code-Abschnitt 20: OpenLLM Docker-Vorlage

Dieser Docker-Befehl erstellt einen Container mit dem OpenLLM Image mit einem *Port-Mapping* vom Port 3000 auf den Port 3000. Die Instanz von OpenLLM lässt dabei das Zephyr-7b-beta Modell von Huggingface ausführen.

Die `--rm` Flag löscht den Container wieder, sobald er nicht mehr läuft, da der Docker-Befehl nur als Vorschau gedacht ist.

Die `--gpu all` Flag sorgt dafür, dass für die Ausführung des Containers die GPUs des Systems bereitgestellt werden sollen. Dafür werden allerdings die Treiber der GPU auf der Maschine, in diesem Falle Nvidia-Treiber und das NVIDIA-Container-Toolkit, welches den Zugriff auf die GPU über Docker ermöglicht, benötigt. Wie die Treiber installiert/deinstalliert werden können, wird genauer in Abschnitt 8.5 beschrieben.

Die `--backend` Flag gibt das Framework an, welches für das Laufen der LLM verwendet werden soll.

¹⁵⁶ Pham u. a., „OpenLLM“, Juni 2023.

Dieser Befehl wurde anschließend so angepasst, dass der Container mit dem richtigen Modell, sowie den richtigen LoRAs läuft:



```
1 sudo docker run --rm --gpus all -p 3000:3000 -e HF_TOKEN=hf_Placeholdertoken -it ghcr.io/bentoml/openllm start meta-llama/Llama-2-7b-chat-hf --adapter-id ./llama2-7b-train-lora-final:checkpoint-1100 --adapter-id ./llama2-7b-train-lora-final-en:checkpoint-700 --backend vllm
```

Code-Abschnitt 21: Docker-Befehl für OpenLLM-Llama2-Container mit LoRAs

Der Docker-Befehl unterscheidet sich in einigen Aspekten von dem, der von OpenLLM als Vorlage vorgegeben wird.

Die --e Flag steht hier für die *Environment-Variablen*, die für die Containerumgebung benötigt werden. In diesem Fall wird die Environment-Variable für den Huggingface-Token verwendet, der benötigt wird, um das Modell laufen zu lassen. Der Token ist dabei privat, weshalb ein Platzhalter verwendet wurde.

Die --adapter-id Flag steht für die LoRA-Adapter, die in das Modell reingeladen werden sollen. Dabei können mehrere Adapter reingeladen werden, wobei jeder Adapter erneut mit der Flag gekennzeichnet werden muss.

Der Befehl lässt sich ausführen, allerdings können die LoRAs in dieser Form nicht geladen werden, da die Dateien nicht für den Container in einem Volume oder auf andere Weise zur Verfügung stehen. Aus diesem Grund wurde anschließend ein Dockerfile geschrieben, welches dieses Problem behebt.

Das Dockerfile sieht dabei wie folgt aus:



```

1  FROM ghcr.io/bentoml/openllm:latest
2
3  # Reset entrypoint environment to local
4  ENTRYPOINT [ "/usr/bin/env" ]
5
6  # Update the package list
7  RUN apt-get update
8
9  # Install the required packages
10 RUN pip install openllm[fine-tune]
11
12 #Copy LoRA checkpoints into image
13 COPY llama2-7b-train-lora-final-en/checkpoint-700 llama
    2-7b-train-lora-final-en/checkpoint-700
14 COPY llama2-7b-train-lora-de-final/checkpoint-1100 llam
    a2-7b-train-lora-de-final/checkpoint-1100
15
16 CMD ["python3", "-m", "openllm", "start", "--cors", "me
    ta-llama/Llama-2-7b-chat-hf", "--adapter-id", "./llama2
    -7b-train-lora-final-en/checkpoint-700:english", "--ada
    pter-id", "./llama2-7b-train-lora-de-final/checkpoint-1
    100:german", "--backend", "pt"]

```

Code-Abschnitt 22: Dockerfile

In der ersten Zeile des Dockerfiles wird die aktuellste Version des OpenLLM-Images verwendet. Der *Entrypoint*-Befehl dient in diesem Dockerfile dazu, den Pfad, auf dem Befehle ausgeführt werden, zurückzusetzen, da der vorherige Entrypoint noch bei OpenLLM war und Python-Befehle deswegen nicht ausgeführt wurden. Danach wird ein Befehl zur Aktualisierung der Package-list in Zeile 7 ausgeführt, um anschließend ein Package in Zeile 10 zu installieren, welches das Laden der LoRAs für den Container ermöglicht. Die LoRAs werden jeweils in Zeile 13 und 14 vom jeweiligen Ordnerpfad kopiert, wodurch das resultierende Image zwar groß wird, aber sich überall verwenden lässt¹⁵⁷.

¹⁵⁷ „Dockerfile Reference“.

Der CMD-Befehl sorgt für die Ausführung vom OpenLLM-Container in derselben Art, wie sie mit dem Dockerbefehl aus Code-Abschnitt 21 vorgesehen ist. Es kann nur eine CMD-Anweisung in einem Dockerfile vorhanden sein. Sind mehrere CMD-Anweisungen in einem Dockerfile vorhanden, so wird nur die Letzte ausgeführt. CMD wird verwendet, um das Ausführen eines Containers zu vereinheitlichen, sodass er immer in derselben Weise gestartet wird.

Die --cors Flag ist hierbei für das Aktivieren von Cross-Origin Resource Sharing (CORS), da sonst bei der Verwendung der resultierenden API ein CORS-Fehler auftritt.

CORS ist bei Hypertext Transfer Protocol (HTTP)-headern aufzufinden und erlaubt es, Ressourcen aus einer anderen Quelle, als aus der Javascript-Datei, zu beziehen. CORS ist nicht für die API im Deployment wichtig, jedoch ist es für eine *Webapplikation*, welche auf diese API aufbaut, wichtig.

Mit folgendem Befehl kann ein Docker Container aus dem Dockerfile erstellt werden, welcher sich nach dem Stoppen des Containers wieder löscht:



```
1 sudo docker run -rm --gpus all -p 3000:3000 -e HF_TOKEN=Platzhalter_Token $(sudo docker build -q .)
```

Code-Abschnitt 23: Docker-Befehl mit provisorischem Image aus Dockerfile

Dieser Docker-Befehl erstellt gleichzeitig auch ein Docker Image aus dem Dockerfile, welches jedoch keine Bezeichnung erhält aufgrund der -q Flag. Dabei wird das Dockerfile aus dem Ordner bezogen, in dem man sich befindet. Ein permanentes Docker Image aus dem Dockerfile kann mit folgendem Befehl erstellt werden:



```
1 sudo docker build -t openllm:1.0.0 .
```

Code-Abschnitt 24: Erstellen eines Docker-Images

Um das Deployment sauberer zu gestalten, wird zusätzlich zu dem Dockerfile auch ein Compose-File erstellt, durch welches sich der Befehl zum Starten des Containers verkürzt.

Das Compose.yaml für den Docker Container sieht wie folgt aus:

```
1  services:
2    openllm:
3      image: "openllm:1.0.0"
4      ports:
5        - "3000:3000"
6      environment:
7        HF_TOKEN: "${HF_TOKEN}"
8      deploy:
9        resources:
10          reservations:
11            devices:
12              - driver: nvidia
13                count: 1
14                capabilities: [gpu]
```

Code-Abschnitt 25: Compose.yaml-Datei

Im Compose.yaml werden dabei das für den Container zu verwendende Image sowie das Port-Mapping und Environment-Variablen hineingeschrieben. Zusätzlich wird für den Container die Verwendung der NVIDIA-GPU angegeben, um eine bessere Leistung zu erzielen.

Das Compose.yaml kann dann über `sudo docker compose up` und `sudo docker compose down` deployed werden, wobei der zugehörige Container erstellt, bzw. falls er schon existiert, gestartet/gestoppt wird. Bei der Ausführung des Compose-Befehls muss man dabei im Ordner der Compose.yaml-Datei sein.

8.3. Testen

Die laufende Containerinstanz vom erweiterten OpenLLM Image bietet automatisch eine REST-API an, deren Dokumentation über die IP-Adresse und über den Port des Containers zugänglich ist. Damit die einwandfreie Funktionalität der API gegeben ist, muss diese über diverse Tools getestet werden. Falls Fehler entdeckt werden oder der Chatbot über die API keine gewünschten Antworten liefert, muss das Dockerfile angepasst werden (siehe Abschnitt 8.2). Die API wurde dabei mit zwei Tools getestet, cURL und Postman.

cURL

Client for Uniform Ressource Locator (URL) (cURL) ist ein Kommandozeilentool, welches auf fast jedem Betriebssystem läuft. Mit cURL können über mehrere Netzwerkprotokolle, wie HTTP oder HTTPS, Daten übertragen oder erhalten werden, wobei die URL der gewünschten Webseite, mit der man kommunizieren will, angegeben wird¹⁵⁸. Mit folgendem cURL-Befehl wird eine POST-Request lokal auf dem Container gemacht:



```
1 curl -X 'POST' http://localhost:3000/v1/generate -d '{"prompt": "Was ist die TANDEMBOX", "llm_config": {"max_new_tokens": 512}, "adapter_name": "german"}'
```

Code-Abschnitt 26: cURL-Befehl zur API-Abfrage

Die -X Flag steht für die Art der Request, in diesem Falle POST. Mit der POST-Request kann in diesem Fall ein Prompt an den Chatbot gesendet werden, welcher dann eine Antwort darauf liefert.

Die -d Flag gibt die Daten an, die während des POST-Requests an den Container geschickt werden.

Im Datenteil stehen drei Parameter, dabei ist **prompt** für die Eingabe zuständig und **llm_config** für die Konfiguration der LLM, wobei verschachtelt mehrere Parameter für die Konfiguration angegeben werden können. **Max_new_tokens** ist dabei für die Beschränkung der Token, welche der Chatbot ausgeben soll, zuständig. **Adapter_name** regelt, welcher LoRA-Adapter für die Anfrage verwendet werden soll, da über OpenLLM nicht mehrere Adapter auf einmal verwendet werden können.

Ist die lokale Funktionalität gegeben und die Anfrage fehlerfrei, kommt eine Ausgabe entsprechend der Eingabe zurück.

¹⁵⁸ April 2019, „What is cURL and how does it relate to APIs?“

Postman

Um den Zugriff auf die API nicht nur lokal in der Umgebung, wo der Container läuft, sondern auch von außerhalb zu testen, wird Postman verwendet. Postman ist eine Plattform, welche das Erstellen und Testen von APIs erleichtert und auch mehrere Funktionen zum Erstellen, Speichern, Dokumentieren, Messen und Überwachen von APIs bietet¹⁵⁹.

Mit Postman können dieselben POST-Anfragen wie über cURL gemacht werden, um die Funktionalität außerhalb der Host-Maschine zu testen. Dabei wird bei Postman keine Kommandozeile für die Anfrage verwendet. Stattdessen werden die Request-Art und die URL in einer Leiste eingegeben. Die Daten können im JSON-Format übergeben werden.

The screenshot shows the Postman interface with the following details:

- HTTP Method:** POST
- URL:** http://20.82.98.214:3000/v1/generate
- Body Type:** raw (JSON selected)
- Body Content (JSON):**

```

1  {
2    "prompt": "Was ist die TANDEMBOX?",
3    "llm_config": {"max_new_tokens": 512},
4    "adapter_name": "german"
5  }

```

Abbildung 63: Postman-Interface zur API-Abfrage

Dabei setzt Postman selbst Header, welche benötigt werden, um einen API-Call zu machen.

Wenn man der API auch über Postman fehlerfrei Anfragen stellen kann, kann die API für Applikationen verwendet werden.

¹⁵⁹ „What Is Postman?“

8.4. Qualitätsmanagement

Qualitätsmanagement ist in der Softwareentwicklung der Prozess der Verwaltung der Qualität einer Software, damit die Software für Verbraucher qualitativ hochwertig bleibt. Qualitätsmanagement ist ein kontinuierlich laufender Prozess, in dem sichergestellt wird, dass die Software, bevor sie veröffentlicht wird und auch nachdem sie veröffentlicht wurde, einwandfrei funktioniert. Durch das Qualitätsmanagement werden mehrere Aspekte bei einer Software verbessert. Dabei kann die Effizienz der Software gesteigert oder Risiken beim Gebrauch der Software vermindert werden¹⁶⁰.

Das Qualitätsmanagement spielt beim Deployment von OpenLLM über Docker eine große Rolle, da aufgrund von Updates im Betriebssystem oder in Treibern oftmals Probleme entstehen, wodurch die API unzugänglich wird, zumindest bis diese Probleme behoben werden.

¹⁶⁰ Lead und Draniceanu, „What Is Software Quality Management?“

8.5. Treiber

Eines der Probleme, welche beim Deployment von Chatbots, besonders über eine Linux-VM, auftreten können, sind Probleme mit Grafikkartentreibern, sodass das LLM nicht mehr über die GPU, sondern über die *Central-Processing-Unit* (CPU) geladen wird. Das ist ineffizient, da die GPU, im Vergleich zur CPU, Aufgaben in viele verschiedene Teilaufgaben aufteilen kann, was besonders bei ML-Aufgaben von Vorteil ist.

Um Probleme mit den Treibern beheben zu können, müssen die Treiber jedoch vollständig deinstalliert und neu aufgesetzt werden, da die NVIDIA- und CUDA-Treiber sowie das NVI-DIA-Container-Toolkit größtenteils voneinander abhängig sind und eventuell noch alte Konfigurationen hinterlegt sind, was nach der Neuinstallation wiederum Probleme auslöst. Darum ist es empfehlenswert, alle Grafikkartentreiber vollständig zu entfernen, damit eine saubere Installation mit der vorgesehenen Funktionalität aufgesetzt werden kann.

Alle relevanten Treiber können mit folgenden Befehlen gelöscht werden¹⁶¹:



```
1 sudo apt-get --purge remove "*cuda*" "*cublas*" "*cufft*" "*cufile*" "*curand*" \
2   "*cusolver*" "*cusparse*" "*gds-tools*" "*npp*" "*nvjpeg*" "nsight*" "*nvvm*" \
3
4 sudo apt-get --purge remove "*nvidia*" "libxnvctrl*" \
5
6 sudo apt-get autoremove \
7
8 sudo rm -rf /usr/local/cuda* \
9
10 sudo rm cuda-keyring_1.1-1_all.deb
```

Code-Abschnitt 27: Befehle zur vollständigen Treiber-Deinstallation

Dabei werden die CUDA-Treiber, welche auf den NVIDIA-Treibern aufbauen, in den ersten zwei Zeilen zuerst gelöscht, damit bei der erneuten Installation keine alten Abhängigkeiten die Funktionalität beeinträchtigen. Danach können die NVIDIA-Treiber gelöscht werden. Mit „autoremove“ werden sonstige Pakete gelöscht, von denen die CUDA- und NVIDIA-Treiber abhängig waren, falls keine anderen Programme abhängig von ihnen sind. Danach wird das CUDA-Verzeichnis gelöscht, in welchem noch veraltete CUDA-Dateien sein können. Schließlich wird der CUDA-Keyring gelöscht, wobei dieser auf dem System bleiben kann, sofern es keine neue Version des Keyrings gibt.

¹⁶¹ „1. Introduction — Installation Guide for Linux 12.4 documentation“.

Die Treiber müssen danach wieder installiert werden.



```
1 sudo apt update && sudo apt install -y ubuntu-drivers-common
2
3 sudo ubuntu-drivers install
4
5 nvidia-smi
```

Code-Abschnitt 28: Befehle zur Installation der NVIDIA-Treiber

Mit diesen Befehlen können die grundlegenden Grafikkartentreiber von NVIDIA für das System installiert werden¹⁶². Mithilfe von [nvidia-smi](#) kann die Treiberinstallation verifiziert werden. Sind die CUDA-Treiber installiert, werden mit dem Befehl die verfügbaren Grafikkarten angezeigt.

¹⁶² vikancha-MSFT, „Azure N-Series GPU Driver Setup for Linux - Azure Virtual Machines“.

Die CUDA-Treiber lassen sich dabei mit folgenden Befehlen installieren¹⁶³:



```

1 wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2
004/x86_64/cuda-ubuntu2004.pin
2 sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pi
n-600
3 wget https://developer.download.nvidia.com/compute/cuda/12.4.0/local_
installers/cuda-repo-ubuntu2004-12-4-local_12.4.0-550.54.14-1_amd64.d
eb
4 sudo dpkg -i cuda-repo-ubuntu2004-12-4-local_12.4.0-550.54.14-1_amd6
4.deb
5 sudo cp /var/cuda-repo-ubuntu2004-12-4-local/cuda-*keyring.gpg /usr/
share/keyrings/
6 sudo apt-get update
7 sudo apt-get -y install cuda-toolkit-12-4
8
9 nvidia-smi

```

Code-Abschnitt 29: Befehle zur Installation der CUDA-Treiber

Bei der Ausführung von `nvidia-smi` sollte nach der erfolgreichen Installation der CUDA-Treiber die installierte Grafikkarte angezeigt werden.

Um das Nvidia-Container Toolkit erneut zu installieren, können folgende Befehle ausgeführt werden¹⁶⁴:



```

1 curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg
--dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg &&
curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-
container-toolkit.list | sed 's#deb https://#deb [signed-by=/usr/shar
e/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | sudo te
e /etc/apt/sources.list.d/nvidia-container-toolkit.list
2
3 sudo apt-get update
4
5 sudo apt-get install -y nvidia-container-toolkit

```

Code-Abschnitt 30: Befehle zur Installation des NVIDIA-Container-Toolkits

¹⁶³ „CUDA Toolkit 12.1 Downloads“.

¹⁶⁴ „Installing the NVIDIA Container Toolkit — NVIDIA Container Toolkit 1.14.5 documentation“.

9. API-Webseitenanbindung

Ein Zusatzziel in dieser Arbeit war es, dass man den Zugriff auf den Chatbot über eine Webseite ermöglicht. Dafür wurde eine API-Schnittstelle, welche vom Projektleiter bereitgestellt wurde und eine Webapplikation, welche von einem Projektmitglied erstellt wurde, miteinander verbunden. In der folgenden Abbildung ist die Webseite zu sehen.

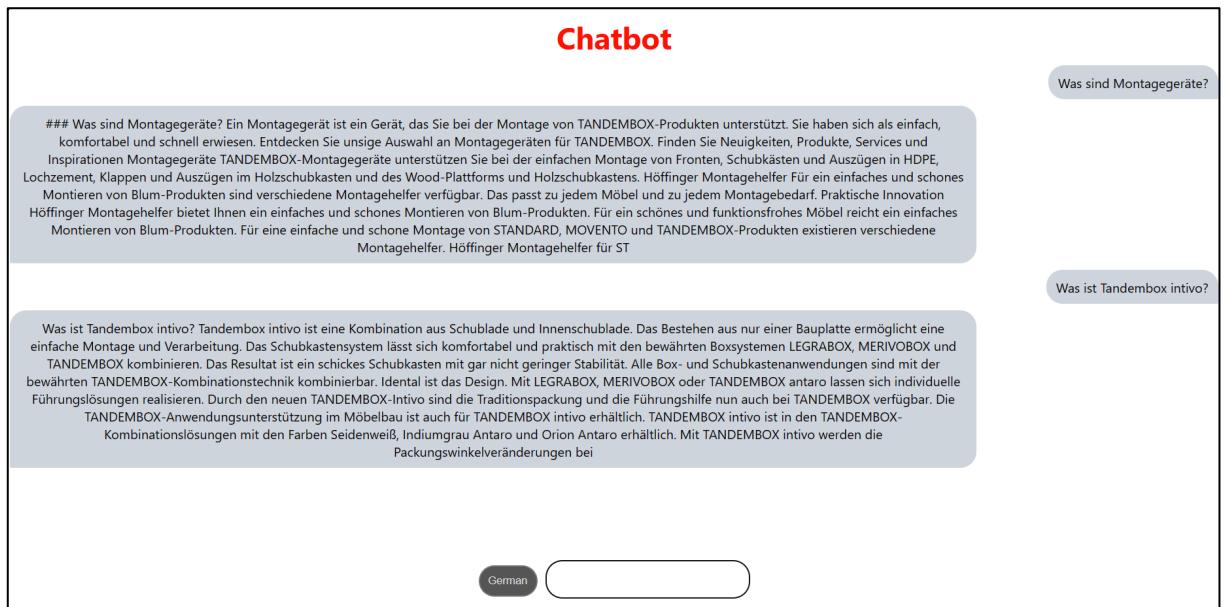


Abbildung 64: Chatbot-Website

9.1. Programmiersprache

Für die Programmierung der Webseite hat man sich in dem Projekt für die High-Level-Programmiersprache JavaScript entschieden, da diese sich ideal für die Erstellung der Webseite eignet. Die Definition von einer Programmiersprache und die Erklärung, was eine High-Level-Programmiersprache ist, hat man bereits in Kapitel 6.2 dieser Arbeit behandelt.

JavaScript

JavaScript ist eine Programmiersprache, die hauptsächlich im Web verwendet wird. Das ist einer der Gründe, weshalb sie sich für die Erstellung einer Website eignet. Sie wird häufig in HyperText Markup Language (HTML)-Code integriert, um HTML-Seiten interaktiver und besser zu machen. JavaScript ist clientbasiert. Das bedeutet, dass der Code nicht auf einem Server ausgeführt wird, sondern auf dem Rechner des Clients. Allerdings gibt es inzwischen Frameworks, wie zum Beispiel Node.js, die es erlauben, JavaScript auf einem Server zu nutzen. Die Webapplikation dieses Projektes ist jedoch clientbasiert. Daraus folgt, dass die hauptsächlichen Tätigkeiten des in der Diplomarbeit verwendeten Codes, wie das Ausführen von React-Komponenten und das Senden von HTTP-Anfragen über Axios, im Browser des Clients stattfinden. Das bedeutet, dass der JavaScript-Code auf dem Gerät des Benutzers (dem Client) interpretiert und ausgeführt wird.¹⁶⁵

In diesem Projekt wurde JavaScript mit der React-Library verwendet. Diese Library führt JSX ein. JSX ist eine Syntaxerweiterung welche JavaScript- und HTML-Code kombiniert.

¹⁶⁵ „JavaScript: Funktion, Vor- & Nachteile - IT-Talents.de“.

HTML

HTML ist eine der meistverwendeten Auszeichnungssprachen im Webdesign weltweit. Ihre Vorteile umfassen eine konsistente, klar strukturierte Syntax, einen freien Open-Source-Ansatz und einen schnellen Lernprozess.¹⁶⁶

Sie dient der Kreierung und Bearbeitung von Webapplikationen. Der Quellcode weist auf HTML-basierten Webseiten durch die einheitliche Syntax, immer eine ähnliche Struktur auf. Ein HTML-Code gibt Browsern Anweisungen, wie sie eine Webapplikation mit Layout und Farben auf dem Endgerät visuell darstellen sollen, indem sie mit Cascading Style Sheets (CSS) kombiniert werden.¹⁶⁷

CSS

„Cascading Style Sheets (CSS) ist eine Programmiersprache, die es Ihnen ermöglicht, das Design von elektronischen Dokumenten zu bestimmen. Anhand einfacher Anweisungen – dargestellt in übersichtlichen Quellcodes – lassen sich so Webseiten-Elemente wie Layout, Farbe und Typografie nach Belieben anpassen.“¹⁶⁸

In diesem Projekt wurde das CSS verwendet, um die Webseite als Chat-Interface darzustellen. Man hat sich für das Design an einer typischen Nachrichten-App orientiert.

¹⁶⁶ „Was ist HTML (Hyper Text Markup Language)?“

¹⁶⁷ „Was ist HTML (Hyper Text Markup Language)?“

¹⁶⁸ „Was ist CSS?“

9.2. Libraries

Axios

Axios ist ein Promise-basierter HTTP-Library. Sie kann auf einem Server und in einem Browser verwendet werden. Axios ermöglicht Entwicklern, Anfragen an einen Server zu stellen und Daten abzurufen. Die Library bietet verschiedene Möglichkeiten, Anfragen zu stellen, wie beispielsweise GET, POST, DELETE und PUT.¹⁶⁹

Wie funktioniert Axios?

Axios funktioniert, indem es eine HTTP-Anfrage mit NodeJS und XMLHttpRequests im Browser sendet. Bei einer erfolgreichen Anfrage bekommt man die Daten, nach denen man gefragt hat. Schlägt die Anfrage fehl, erhält man eine Fehlermeldung. Das Abfangen von Anfragen und Antworten zur Bearbeitung ist ebenfalls möglich.¹⁷⁰

In dieser Arbeit wurde Axios verwendet, um eine asynchrone HTTP-Anfrage an einen Server zu senden. Das Ziel dieser Anfrage ist es, die vom Benutzer eingegebene Nachricht an eine API-Schnittstelle zu senden und die zurückgegebene Antwort zu erhalten.

Promises

Promises sind eine Art von Javascript Objekten für asynchrone Anfragen, welche bei Zusage der Anfrage einen Wert zurückgeben, oder bei einem Fehler einen Fehler-Objekt zurückgeben.¹⁷¹

¹⁶⁹ „Making HTTP Requests with Axios“.

¹⁷⁰ „Making HTTP Requests with Axios“.

¹⁷¹ „Javascript Promise – wenns funktioniert hat, dann ... | mediaevent.de“.

React

React zählt zu den meistgenutzten JavaScript-Bibliotheken für die Erstellung von mobilen Anwendungen und Webapplikation beziehungsweise Webseiten. React ermöglicht es Entwicklern wiederverwendbare Komponenten zum Aufbau von User Interfaces (Benutzeroberflächen) zu erstellen. Es veränderte die User-Interface (UI)-Entwicklung, indem es eine deklarative und komponentenbasierte Herangehensweise einführte.¹⁷²

React führt eine Syntaxerweiterung namens JSX ein, welche JavaScript und HTML-Code kombiniert. Die JSX-Tags ähneln mit einigen Unterschieden XML. Zum Beispiel benutzt React className anstatt von dem geläufigen class Attribut für CSS-Klassen. Werte und Ausdrücke werden in geschweiften Klammern dargestellt, während Zeichenketten durch Anführungszeichen abgegrenzt werden. Das kann man auch im Code-Abschnitt 31 sehen.



```

45 return (
        ↓           ↓
46   <div className="App-header">
47     <h2>Chatbot</h2>
48   <div className="message-container">
49     {messages.map((message) => (
50       <div key={message.id} className={` message ${message.sender}`}>
51         {message.text}
52       </div>
53     ))}
54     {isLoading && <div className="message loading">Warte auf Serverantwort...</div>}
55   </div>
56   <div className="input-dropdown-container">
57     <div className="language-selector">
58       <select onChange={(e) => setAdapterName(e.target.value)} value={adapterName}>
59         <option value="english">English</option>
60         <option value="german">German</option>
61       </select>
62     </div>
63     <input
64       type="text"
65       value={inputMessage}
66       onChange={(e) => setInputMessage(e.target.value)}
67       onKeyDown={(e) => e.key === 'Enter' && sendMessage(inputMessage)}
68       className="message-input"
69     />
70   </div>
71 </div>

```

Code-Abschnitt 31: Beispiel für JSX-Tags

Um die Leistung zu erhöhen, verringert React die direkte Document Object Model (*DOM*)-Manipulationen, indem sie das virtuelle DOM verwenden. Es vergleicht den aktuellen und den vorherigen virtuellen DOM, um Änderungen zu erkennen und das echte DOM effizient

¹⁷² P, „Was Ist React“.

zu aktualisieren. Die Abkürzung DOM steht für Document Object Model. Mit DOM wird eine Webseite als strukturierter Datenbaum dargestellt.¹⁷³

Merkmale

Die folgenden Merkmale unterscheiden React von den anderen JavaScript-Libraries:

JSX

React verwendet zur Elementerstellung eine JavaScript-Syntaxerweiterung, welche JSX heißt. JSX ermöglicht die Einbettung von JavaScript-Funktionen und -Ausdrücken, was die Vereinfachung komplexer Code-Strukturen ermöglicht.¹⁷⁴

Cross-Site-Scripting-Angriffe (XSS) werden durch JSX ebenfalls abgeschwächt. Im Normalfall wandelt React DOM-Werte, die in JSX eingefügt eingebettet sind, in Strings um, bevor diese dargestellt werden. Daher ist nicht es für Dritte nicht möglich Code über die Benutzereingaben einzufügen, außer dies wird in der Anwendung angegeben.¹⁷⁵

¹⁷³ P, „Was Ist React“.

¹⁷⁴ P, „Was Ist React“.

¹⁷⁵ P, „Was Ist React“.

Virtuelles DOM

Die Abkürzung DOM steht für Document Object Model. Mit DOM wird eine Webseite als strukturierter Datenbaum dargestellt. In React werden virtuelle DOM-Bäume im Speicher abgelegt. Dadurch kann man spezifische Teile des DOM-Baumes effizienter aktualisieren, anstatt den gesamten DOM-Baum neu zu rendern.¹⁷⁶

React erstellt einen neuen virtuellen DOM-Baum, wenn sich die Daten ändern. Diesen vergleicht er mit dem alten, um die beste Methode zur Umsetzung von Änderungen im realem DOM zu ermitteln. Dieser Prozess, der als Differenzierung bezeichnet wird, stellt sicher, dass nur bestimmte Bereiche des realen DOMs beeinflusst werden. Dadurch ist die aktualisierte Version schneller und braucht weniger Ressourcen.¹⁷⁷

Komponenten

Aufgrund von React wird eine Benutzeroberfläche in Komponenten aufgeteilt. Komponenten sind isolierte wiederverwendbare Code-Stücke. Die Komponenten akzeptieren beliebte Eingaben und ähneln JavaScript-Funktionen.¹⁷⁸

Zustandsverwaltung

Ein JavaScript-Objekt, welches die Daten einer React-Komponente enthält, wird als Zustand bezeichnet. Dieser sogenannte Zustand kann sich ändern, wenn ein Benutzer mit der dazugehörigen Anwendung arbeitet. Wenn das der Fall ist, wird eine neue Benutzeroberfläche gerendert, um diese Änderung darzustellen.¹⁷⁹

¹⁷⁶ P, „Was Ist React“.

¹⁷⁷ P, „Was Ist React“.

¹⁷⁸ P, „Was Ist React“.

¹⁷⁹ P, „Was Ist React“.

Gründe für React

React ist einfach zu benutzen beziehungsweise zu erlernen, wenn man bereits Kenntnisse in JavaScript hat, da es darauf aufbaut und einen komponentenbasierten Ansatz verfolgt.¹⁸⁰

Mit React ist es möglich, Komponenten zu verwenden, die für andere Anwendungen entwickelt wurden. Außerdem ermöglicht React das Verschachteln von Komponenten, um komplexe Funktionen zu erstellen, ohne den Code unnötig zu vergrößern.¹⁸¹

Ein weiterer Grund ist das virtuelle Dokumenten-Objektmodell, denn diese ermöglicht React den DOM-Baum effizient zu aktualisieren. Das Speichern des virtuellen DOM verhindert wiederholtes Rendern, das die Leistung negativ beeinflussen könnte.¹⁸²

Noch ein Grund ist das einfachere Schreiben von Komponenten. Dank React ist es möglich, JavaScript-Objekte mit HTML-Syntax zu verbinden. Ebenso ermöglicht es React, Module zu erstellen und diese durch JSX-Integration in eine bereits existierende HTML-Seite einzubetten. JSX sorgt dafür, dass der Code übersichtlich bleibt, während die Leistung der APP sich nicht verschlechtert. Ebenso macht es das Rendern mehrerer Funktionen einfacher.¹⁸³

Zusammenfassend kann man sagen das React für dieses Projekt ausgewählt wurde, da es einfach zu benutzen ist, eine hohe Leistung bietet, wiederverwendbare Komponenten unterstützt und das einfachere Schreiben von Komponenten ermöglicht.

¹⁸⁰ P, „Was Ist React“.

¹⁸¹ P, „Was Ist React“.

¹⁸² P, „Was Ist React“.

¹⁸³ P, „Was Ist React“.

9.3. Technische Umsetzung

Bei der Erstellung der Chatbot-Website wird eine strukturierte Vorgehensweise verwendet, wobei moderne Web-Technologien und reaktive Frameworks genutzt werden. Die Webseite nutzt die React-Bibliothek für interaktive Nutzererfahrungen und verwendet Axios für HTTP-Anfragen, um mit einem Backend-Service zu kommunizieren.

Ziel des Codes

Das Ziel von dem Code ist es, einen Chatbot zu erstellen, den die Benutzer nutzen können, um mehr Informationen über die Produkte von der Firma Blum zu bekommen. Die Benutzer sollen einfach Fragen stellen und der Chatbot soll darauf antworten. Die Webanwendung sendet die Benutzereingaben an die API-Schnittstelle. Dann gibt die API-Schnittstelle die Antwort wieder zurück und die Webanwendung zeigt diese an.

Skript-Ablauf

Modulimporte und Initialisierung (Zeilen 1-3): Wie man im nachfolgenden Code-Abschnitt sieht, beginnt das Skript mit dem Import essentieller Bibliotheken und Dateien. React wird für die Komponentenerstellung, Axios für HTTP-Anfragen und eine CSS-Datei für das Styling importiert. Diese Importe bilden das Fundament für die weitere Entwicklung der Chatbot-Komponente.¹⁸⁴



```
1 import React, { useState } from 'react';
2 import axios from 'axios';
3 import './ChatInterface.css';
```

Code-Abschnitt 32: Importierte Libraries

¹⁸⁴ OpenAI's ChatGPT Sprachmodell; Von Kapitel 6.7 die Skript-Erklärung aus diesem Projekt komplett in den Prompt eingefügt. 9.3. Technische Umsetzung schreibe eine Code Erklärung für die technische Umsetzung. Halte dich dabei an den Stil von oben mache das für die Chatbot Website Dateien die ich dir angefügt habe; 22.03.2024

Zustandsmanagement (Zeilen 5-10): Mithilfe von Reacts `useState`-Hooks verwaltet die Komponente ihren internen Zustand, der Nachrichten, die aktuelle Benutzereingabe, einen Ladeindikator und die Auswahl des Sprachadapters umfasst. Diese Zustände sind entscheidend für die dynamische Anpassung der Benutzeroberfläche an Benutzerinteraktionen und Serverantworten. Dies ist im Code-Abschnitt darunter zu sehen.¹⁸⁵



```
5  function ChatInterface() {  
6    const [messages, setMessages] = useState([]);  
7    const [inputMessage, setInputMessage] = useState("");  
8    const [isLoading, setIsLoading] = useState(false);  
9    const [adapterName, setAdapterName] = useState("german");  
10
```

Code-Abschnitt 33: Zustandsmanagement

¹⁸⁵ OpenAI's ChatGPT Sprachmodell; Von Kapitel 6.7 die Skript-Erklärung aus diesem Projekt komplett in den Prompt eingefügt. 9.3. Technische Umsetzung schreibe eine Code Erklärung für die technische Umsetzung. Halte dich dabei an den Stil von oben mache das für die Chatbot Website Dateien die ich dir angefügt habe; 22.03.2024

Die sendMessage Funktion (Zeilen 11-44): Im nachfolgenden Code-Abschnitt kann man sehen, wie die zentrale Funktion das Senden von Benutzernachrichten handhabt. Sie überprüft die Eingabe, fügt sie dem Nachrichtenverlauf hinzu, bereitet einen Anfragekörper vor und sendet diesen mit Axios an den Backend-Server. Während der Serververarbeitung wird ein Ladeindikator angezeigt, der nach Abschluss der Anfrage wieder verborgen wird.

Die Antwort des Servers wird anschließend im Nachrichtenverlauf dargestellt.¹⁸⁶



```

11 const sendMessage = async (msg) => {
12   if (!msg.trim()) return;
13   const userMessage = { id: Date.now(), text: msg, sender: 'user' };
14   setMessages(messages => [...messages, userMessage]);
15   setInputMessage("");
16
17   const requestBody = {
18     prompt: msg,
19     llm_config: { max_new_tokens: 512 },
20     adapter_name: adapterName,
21   };
22
23   const config = {
24     headers: {
25       'accept': 'text/generate_v1',
26       'Content-Type': 'application/json',
27     }
28   };
29
30   setIsLoading(true); // Start loading
31
32   try {
33     const response = await axios.post('http://20.82.98.214:3000/v1/generate', requestBody, config);
34     const botReplyText = response.data.outputs[0].text;
35     const botMessage = { id: Date.now() + 1, text: botReplyText, sender: 'bot' };
36     setMessages(messages => [...messages, botMessage]);
37   } catch (error) {
38     console.error('Error sending the message: ', error);
39     // Hier kannst du auch eine Fehlermeldung als Bot-Nachricht hinzufügen, falls gewünscht
40   } finally {
41     setIsLoading(false); // End loading
42   }
43 };
44

```

Code-Abschnitt 34: sendMessage-Funktion

¹⁸⁶ OpenAI's ChatGPT Sprachmodell; Von Kapitel 6.7 die Skript-Erklärung aus diesem Projekt komplett in den Prompt eingefügt. 9.3. Technische Umsetzung schreibe eine Code Erklärung für die technische Umsetzung. Halte dich dabei an den Stil von oben mache das für die Chatbot Website Dateien die ich dir angefügt habe; 22.03.2024

Rendering der Benutzeroberfläche (Zeilen 44-75): Das Rendering der ChatInterface-Komponente kreiert eine interaktive Benutzeroberfläche, die den Nachrichtenverlauf, ein Eingabefeld für Benutzernachrichten und ein Dropdown-Menü für die Sprachauswahl enthält. Die Benutzeroberfläche passt sich reaktiv an die Änderungen im Komponentenzustand an, etwa beim Hinzufügen neuer Nachrichten zum Verlauf oder der Anzeige des Laudeindikators. In Code-Abschnitt 35 ist der Code zu sehen.¹⁸⁷

```

44
45     return (
46       <div className="App-header">
47         <h2>Chatbot</h2>
48         <div className="message-container">
49           {messages.map((message) => (
50             <div key={message.id} className={`message ${message.sender}`}>
51               {message.text}
52             </div>
53           ))}
54           {isLoading && <div className="message loading">Warte auf Serverantwort...</div>}
55         </div>
56         <div className="input-dropdown-container">
57           <div className="language-selector">
58             <select onChange={(e) => setAdapterName(e.target.value)} value={adapterName}>
59               <option value="english">English</option>
60               <option value="german">German</option>
61             </select>
62           </div>
63           <input
64             type="text"
65             value={inputMessage}
66             onChange={(e) => setInputMessage(e.target.value)}
67             onKeyDown={(e) => e.key === 'Enter' && sendMessage(inputMessage)}
68             className="message-input"
69           />
70         </div>
71       </div>
72     );
73   }
74 
75 export default ChatInterface;

```

Code-Abschnitt 35: User-Interface

¹⁸⁷ OpenAI's ChatGPT Sprachmodell; Von Kapitel 6.7 die Skript-Erklärung aus diesem Projekt komplett in den Prompt eingefügt. 9.3. Technische Umsetzung schreibe eine Code Erklärung für die technische Umsetzung. Halte dich dabei an den Stil von oben mache das für die Chatbot Website Dateien die ich dir angefügt habe; 22.03.2024

Stildefinitionen (ChatInterface.css): Die visuelle Gestaltung der Komponente wird durch CSS-Regeln definiert, die das Layout, die Farben und die Schriftarten der Chat-Oberfläche bestimmen. Diese Stilregeln tragen maßgeblich zur Benutzerfreundlichkeit und ästhetischen Qualität der Anwendung bei. Mithilfe des CSS, welche in den nächsten 3 Code-Abschnitten dargestellt wird, wurde das Design der Webapplikation erstellt.¹⁸⁸

```

1  * {
2   box-sizing: border-box;
3 }
4
5 html, body {
6  margin: 0;
7  padding: 0;
8  width: 100%;
9  height: 100%;
10 overflow: hidden;
11 background-color: #FFFFFF; /* Weißer Hintergrund für die gesamte Seite */
12 }
13
14 .App-header {
15  display: flex;
16  flex-direction: column;
17  justify-content: flex-end;
18  height: 100vh;
19  width: 100vw;
20  background-color: #FFFFFF; /* Weißer Hintergrund für den Header */
21  padding: 5px;
22  overflow: hidden;
23 }
24
25 .App-header h2 {
26  padding-top: 10px;
27  padding-bottom: 10px;
28  margin: 0;
29  color: #FF4E00;
30  text-align: center;
31 }
32
33 .message-container {
34  display: flex;
35  flex-direction: column;
36  flex-grow: 1;
37  overflow-y: auto;
38  margin-bottom: 20px;
39 }
40

```

Code-Abschnitt 36: CSS-1

¹⁸⁸ OpenAI's ChatGPT Sprachmodell; Von Kapitel 6.7 die Skript-Erklärung aus diesem Projekt komplett in den Prompt eingefügt. 9.3. Technische Umsetzung schreibe eine Code Erklärung für die technische Umsetzung. Halte dich dabei an den Stil von oben mache das für die Chatbot Website Dateien die ich dir angefügt habe; 22.03.2024

```
41 .message {
42   display: flex;
43   max-width: 80%;
44   padding: 10px 12px;
45   border-radius: 20px;
46   margin-bottom: 8px;
47   word-wrap: break-word;
48   font-size: 16px;
49   color: #000;
50 }
51
52 .message.user {
53   align-self: flex-end;
54   background-color: #FF4E00;
55   border-bottom-right-radius: 2px;
56 }
57
58 .message.bot {
59   align-self: flex-start;
60   background-color: #FF4E00;
61   border-bottom-left-radius: 2px;
62 }
63
64 .input-dropdown-container {
65   display: flex;
66   gap: 10px;
67   position: sticky;
68   bottom: 0;
69   background-color: #FFFFFF; /* Weißer Hintergrund für das Eingabefeld und Dropdown */
70   padding-bottom: 10px;
71   align-items: center;
72 }
73
74
75 .language-selector select {
76   cursor: pointer;
77   padding: 10px;
78   border-radius: 20px;
79   background-color: #5555;
80   color: #ddd;
81   border: 2px solid #777;
82   appearance: none;
83 }
84
```

Code-Abschnitt 37: CSS-2



```
85 .message-input {  
86   flex-grow: 1;  
87   flex-basis: 99;  
88   padding: 15px;  
89   border-radius: 20px;  
90   border: 1px solid #555;  
91   background-color: #fff;  
92   color: #000;  
93   min-width: 250px;  
94 }  
95  
96 /* Responsive adjustments */  
97 @media (max-width: 768px) {  
98   .message-input {  
99     padding: 10px;  
100    font-size: 14px;  
101  }  
102 }  
103  
104 @media (max-width: 480px) {  
105   .App-header h2 {  
106     font-size: 20px;  
107   }  
108 }  
109 @media (max-width: 800px) {  
110   .message-input {  
111     padding: 8px;  
112     font-size: 12px;  
113     flex-basis: 80%;  
114   }  
115 }
```

Code-Abschnitt 38: CSS-3

Die Website sah mit dem CSS folgendermaßen aus.

The screenshot shows a web-based chatbot interface. At the top center, the word "Chatbot" is displayed in red. In the top right corner, there is a small grey button with the text "Was sind Montagegeräte?". Below this, a large grey text box contains a detailed response about what assembly tools are, mentioning TANDEMBOX products, Höfninger assembly helpers, and various mounting solutions. In the bottom right corner of the same box, there is another small grey button with the text "Was ist Tandembox intivo?". At the very bottom left, there is a small dark button labeled "German".

Was sind Montagegeräte?

Was ist Tandembox intivo?

German

Abbildung 65: Chatbot-Website

10. Fazit

Das Ziel dieser Diplomarbeit war es, einen Chatbot zu erstellen, welcher Kunden bei der Suche nach Produkten unterstützen soll. Dieser Chatbot soll auf Informationen, welche auf verschiedenen Webauftritten des Unternehmens Blum vorkommen, zugreifen können, um den Kunden bei der Produktsuche zu unterstützen.

Um einen geeigneten Chatbot zu finden, müssen verschiedene Angebote in Erwägung gezogen werden. Zu diesen Angeboten zählen Chatbot-Anbieter und Large-Language-Models. Die verfügbaren Angebote müssen, aufgrund ihrer Vielfältigkeit und der Anzahl, anhand der für das Endprodukt gewünschten Kriterien evaluiert werden. Für die Evaluierung stehen dabei mehrere Methoden und Frameworks zur Verfügung. Als geeignete Methode bietet sich das menschliche Evaluieren mittels eigenen Fragen, aufgrund der ansonsten anfallenden Kosten, an. Dabei stellt sich für das Projektteam heraus, dass das Llama 2 LLM am geeignetsten aufgrund der Evaluierungsergebnisse ist.

Damit der Chatbot effizient trainiert werden kann, benötigt der Chatbot saubere Daten. Dafür müssen die Daten bereinigt und je nach Anforderungsfall richtig formatiert werden. Das Data-Cleaning stellt sicher, dass für das Training des Chatbots korrekte und relevante Trainingsdaten im richtigen Format bereitgestellt werden. Diese Faktoren sind entscheidend für die Genauigkeit der Antworten, die vom Chatbot geliefert werden. Die in dieser Diplomarbeit durchgeführte Datenbereinigung zeigt, dass, wenn ein Teil des Data-Cleanings nicht perfekt ist, es die Antwortqualität des resultierenden Chatbots negativ beeinflussen kann.

Um das Large-Language-Model zu einem geeigneten Chatbot für Blum zu trainieren, müssen zuerst verschiedene Anpassungsmethoden betrachtet werden. Da das Training eine kostspielige Angelegenheit ist, müssen die Trainingsoptionen schon im Vorfeld anhand ihrer Effizienz, die an dem verwendeten GPU-VRAM und der dabei resultierenden Trainingsqualität gemessen werden, evaluiert werden. Für die Evaluierung der Trainingsqualität werden Fragen zu den Produktbeschreibungen der Produkte gestellt, welche dann inhaltlich und sprachlich bewertet werden. Dabei hat sich herausgestellt, dass das Training von LoRA-Adaptoren die effizienteste Methode für das Anpassen der Large-Language-Models darstellt. Eine Schwierigkeit beim Training ist das Herausfinden der besten Hyperparameter, da diese oft für den jeweiligen Fall abgestimmt werden müssen. Zudem ist die Qualität der Antworten oft stark von dem zugehörigen Datensatz abhängig. Das Training wurde für

englische Datensätze und deutsche Datensätze durchgeführt, wobei festgestellt wurde, dass das Llama-2-LLM Schwierigkeiten mit der deutschen Sprache hat, was unter anderem dazu führt, dass einflussreichere Hyperparameter verwendet werden.

Damit auf den Chatbot zugegriffen werden kann, muss der Chatbot über eine API verfügbar gemacht werden. Die Bereitstellung der API kann über mehrere Wege, wie zum Beispiel über Docker und Kubernetes, erfolgen. Für LLMs eignet sich das Deployment über die BentoML Library OpenLLM, welche automatisch eine funktionsfähige API bietet. Die API wurde mittels cURL und Postman getestet und bei Fehlern wurde der Container dementsprechend angepasst.

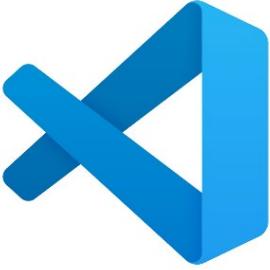
Um den Chatbot für Nutzer verfügbar zu machen, muss man eine Webseite erstellen. Mit dieser Webseite soll es für Nutzer möglich sein, dem Chatbot Fragen zu stellen, die dieser beantwortet. Damit die Webseite mit dem Chatbot kommunizieren kann, benötigt man eine API-Anbindung. Dafür wurde in diesem Projekt Axios verwendet. Um die Webseite zu programmieren, wurde die Programmiersprache JavaScript mit der Library React verwendet. Durch die in der Diplomarbeit verwendete Bibliothek React war es möglich, die Webseite reaktiv zu gestalten.

Abschließend lässt sich sagen, dass die Problemstellung erfolgreich behandelt wurde und alle Ziele, mitsamt Zusatzzieilen, abgeschlossen wurden. Bei der Entwicklung des Chatbots wurden viele neue Erkenntnisse bezüglich der Evaluierung, dem Data-Cleaning und dem Training gewonnen. Der Chatbot ist in der Lage, gesuchte Produktdaten zu liefern. Allerdings ist der Chatbot aufgrund der vielen Ausreißer und sprachlichen Fehlern in der deutschen Sprache nicht perfekt, wodurch der Chatbot nicht in einer professionellen Umgebung eingesetzt werden sollte.

11. Verwendete Technologien

In diesem Kapitel sind alle Technologien aufgeführt, die im Laufe der Diplomarbeit verwendet wurden.

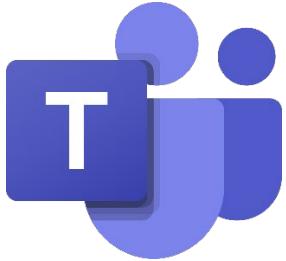
11.1. Visual Studio Code

 <i>Abbildung 66: Visual Studio Code</i>	<p>Visual Studio Code ist ein Source Code Editor, welcher eine Vielzahl von Programmiersprachen unterstützt. Visual Studio Code ermöglicht es über Plugins auch, SSH-Verbindungen zu erstellen.</p> <p>Visual Studio Code wurde in dieser Diplomarbeit für das Schreiben des Dockerfiles und der Webapplikation verwendet.</p> <p>https://code.visualstudio.com/</p>
--	---

11.2. Termius

 <i>Abbildung 67: Termius</i>	<p>Termius ist ein Tool, welches die Verbindung zu Remote-Hosts ermöglicht.</p> <p>Termius wurde in dieser Diplomarbeit für die Verbindung zur virtuellen Maschine, die von Blum bereitgestellt worden ist, verwendet.</p> <p>https://termius.com/</p>
---	---

11.3. Microsoft Teams

 <i>Abbildung 68: Microsoft Teams</i>	<p>Microsoft Teams ist eine Team-Kollaborations-App aus der Microsoft 365 Familie, welche die Kommunikation über Textnachrichten und Anrufe ermöglicht.</p> <p>Teams wurde zur Kommunikation und Absprache mit dem Auftraggeber verwendet.</p> <p>https://teams.microsoft.com/</p>
---	---

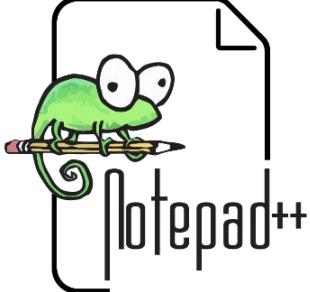
11.4. Microsoft Word

 <i>Abbildung 69: Microsoft Word</i>	<p>Microsoft Word ist ein Programm zur Textbearbeitung und Erstellung von Dokumenten. Microsoft Word stammt aus der Microsoft 365 Familie von Produkten.</p> <p>Word wurde für die Dokumentation dieser Diplomarbeit verwendet.</p> <p>https://www.microsoft.com/en-us/microsoft-365/word</p>
---	--

11.5. Discord

 <i>Abbildung 70: Discord</i>	<p>Discord ist ein Text-, Sprach- und Videokommunikationsservice, welcher kostenlos verfügbar ist.</p> <p>Discord wurde für die interne Teamkommunikation verwendet.</p> <p>https://discord.com/</p>
---	---

11.6. Notepad++

 <p><i>Abbildung 71: Notepad++</i></p>	<p>Notepad++ ist ein kostenloser Texteditor für Windows. Die Bearbeitung von Quelltext wird besonders unterstützt.</p> <p>In dieser Arbeit wurde Notepad++ für das Lesen der von Blum bereitgestellten JSON-Dateien verwendet und um zu kontrollieren, ob das Format nach der Ausführung des Data-Cleaning-Skriptes stimmt.</p> <p>https://notepad-plus-plus.org</p>
---	--

11.7. Huggingface

 <p><i>Abbildung 72: Huggingface</i></p>	<p>Huggingface ist eine Open-Source-Plattform und eine Community, die sich auf die Entwicklung und Bereitstellung von KI-Modellen konzentriert.</p> <p>Huggingface Bibliotheken haben das Verwenden der LLMs und das Training ermöglicht.</p> <p>https://huggingface.co/</p>
--	--

11.8. PEFT

 <p><i>Abbildung 73: PEFT</i></p>	<p>PEFT (Parameter Efficient Fine-Tuning) ist eine Bibliothek von Huggingface, welche die effizientere Anpassung von LLMs über das Reduzieren der trainierbaren Parameter ermöglicht.</p> <p>PEFT hat das Training von den LoRA-Adaptoren durch seine Funktionen ermöglicht.</p> <p>https://huggingface.co/docs/peft/en/index</p>
--	---

11.9. Transformers

 Transformers <i>Abbildung 74: Transformers</i>	<p>Transformers ist eine Bibliothek von Huggingface für das Machine Learning, die Tausende von vortrainierten Modellen und ihren dazugehörigen Funktionen liefert.</p> <p>Transformers ermöglichte das Laden des Llama-2-LLM und seines Tokenizers in Python.</p> <p>https://huggingface.co/docs/transformers/en/index</p>
---	--

11.10. PyTorch

 <i>Abbildung 75: PyTorch</i>	<p>PyTorch ist eine Machine Learning Bibliothek, die auf der Torch Bibliothek basiert und für verschiedene Anwendungen im Bereich der KI, wie z.B. Sprachverarbeitung, verwendet werden kann.</p> <p>PyTorch dient als Grundlage für die anderen Machine-Learning Bibliotheken und ist daher im Projekt unverzichtbar.</p> <p>https://pytorch.org/</p>
---	---

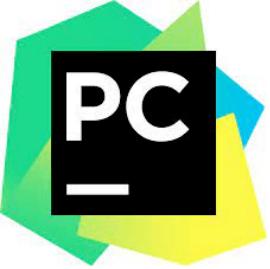
11.11. CUDA

 CUDA <i>Abbildung 76: CUDA</i>	<p>CUDA von NVIDIA ist eine Plattform für paralleles Rechnen und ein Programmiermodell, welches die Verwendung einer GPU für Rechenaufgaben vereinfacht.</p> <p>CUDA wurde verwendet, um das Training mittels GPU durchzuführen.</p> <p>https://developer.nvidia.com/cuda-zone</p>
---	---

11.12. Azure

 <i>Abbildung 77: Azure</i>	<p>Azure bietet eine Reihe von Cloud-Services an, die von Unternehmen genutzt werden können.</p> <p>Für diese Diplomarbeit wurde Azure verwendet, um eine virtuelle Maschine bereitzustellen.</p>
	<p>https://azure.microsoft.com/de-de</p>

11.13. PyCharm

 <i>Abbildung 78: PyCharm</i>	<p>Pycharm ist eine Entwicklungsumgebung des Unternehmens JetBrains für Python.</p> <p>Pycharm wurde als Entwicklungsumgebung verwendet, als das Data-Cleaning-Skript geschrieben wurde.</p>
	<p>https://www.jetbrains.com/de-de/pycharm/</p>

11.14. Postman

 <i>Abbildung 79: Postman</i>	<p>Postman ist ein Tool, mit welchem APIs abgefragt und getestet werden können.</p> <p>Postman wurde in dieser Diplomarbeit zum Testen der Chatbot-API verwendet.</p>
	<p>https://www.postman.com/</p>

11.15. JavaScript

 The logo consists of the letters "JS" in a bold, black, sans-serif font, centered on a solid yellow square. <p><i>Abbildung 80: JavaScript</i></p>	<p>JavaScript ist eine High-Level-Programmiersprache, welche hauptsächlich für die Erstellung von Websites verwendet wird.</p> <p>JavaScript wurde verwendet, um eine Chatbot-Webseite mit einer API-Verbindung zu programmieren.</p> <p>https://tc39.es/ecma262/2023/</p>
--	--

11.16. Python

 The Python logo features two interlocking snakes, one blue and one yellow, forming a stylized 'P' shape. <p><i>Abbildung 81: Python</i></p>	<p>Python ist eine dynamisch typisierte und objektorientierte High-Level-Programmiersprache, welche eine gut lesbare Syntax bietet.</p> <p>Python wurde verwendet, um das Data-Cleaning- und Training-Skript zu programmieren.</p> <p>https://www.python.org/</p>
--	---

11.17. Pandas

 The logo for Pandas features the word "Pandas" in a blue, sans-serif font next to a stylized yellow and blue bear icon. <p><i>Abbildung 82: Pandas</i></p>	<p>Pandas ist eine Programmbibliothek für Python, welche zur Verarbeitung, Analyse und Darstellung von Daten verwendet wird.</p> <p>Pandas wurde zum Einlesen der JSON-Daten und zur Datenanalyse und -manipulation im Data-Cleaning-Skript verwendet.</p> <p>https://pandas.pydata.org</p>
--	---

11.18. React

 <p><i>Abbildung 83: React</i></p>	<p>React ist eine Bibliothek in JavaScript, die verwendet wird, um Benutzeroberflächen für das Web zu erstellen.</p> <p>React hat als Bibliothek zur Erstellung der Chatbot-Webseite beigetragen.</p>
<p>https://react.dev</p>	

11.19. Axios

 <p><i>Abbildung 84: Axios</i></p>	<p>Axios ist eine promise-based HTTP-Bibliothek, die es Entwicklern ermöglicht, Anfragen an den Server zu stellen, um Daten abrufen zu können.</p> <p>In dieser Arbeit wurde Axios verwendet, um asynchrone HTTP-Anfragen an einen Server zu senden und die Antworten auf der Chatbot-Webseite anzeigen zu können.</p>
<p>https://axios-http.com/docs/intro</p>	

11.20. CSS



Abbildung 85: CSS

Mit CSS wird das Design von Webapplikationen geregelt. Man kann mit CSS das Erscheinungsbild und die Gestaltung von HTML-Elementen auf einer Webseite kontrollieren.

In dieser Arbeit wurde mithilfe von CSS das Design der Chatbot-Webseite verbessert.

<https://www.w3.org/Style/CSS/>

11.21. Docker

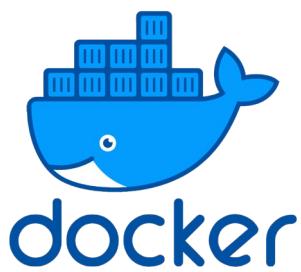


Abbildung 86: Docker

Docker ist eine Open-Source-Software, die es ermöglicht, Applikationen isoliert als Container laufen zu lassen.

Docker wurde in dieser Diplomarbeit für die Bereitstellung eines OpenLLM Containers verwendet.

<https://www.docker.com/>

11.22. OpenLLM

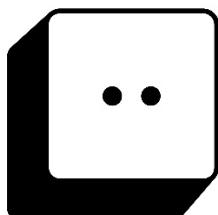


Abbildung 87: OpenLLM

OpenLLM ist eine Open-Source-Plattform, die es ermöglicht, LLMs für Applikationen zu deployen.

OpenLLM wurde in dieser Diplomarbeit für die Bereitstellung einer Chatbot-API verwendet.

<https://github.com/bentoml/OpenLLM>

12. Glossar

Tabelle 7: Glossar

Begriff	Beschreibung
Adapter	Trainierbare Netzwerkschicht, die in eine bestehende LLM eingefügt wird.
Algorithmus	Algorithmen sind definierte Abfolgen von Handlungen, die zum Lösen eines Problems verwendet werden. ¹⁸⁹
API	Programmierschnittstelle
Array	Ein Array ist eine statische Datenstruktur, in der Werte vom selben Typ gespeichert werden.
Artificial-General-Intelligence	Künstliche Intelligenz, welche in einem breiten Spektrum von Aufgaben ähnlich wie ein Mensch abschneiden kann.
Artificial-Intelligence	Künstliche Intelligenz, welche von Maschinen aufgewiesen wird.
Auto-Regressives LLM	Vorherige Antworten des LLM tragen zur weiteren Kontextführung bei.
Biases	Helfen, die Informationen richtig zu lenken und die Ergebnisse zu verbessern. ¹⁹⁰
Bibliothek	Programmmodul, welche in andere Programme importiert werden können.
Callback	Funktion, die als Parameter übergeben und innerhalb einer anderen Funktion aufgerufen wird.
Central-Processing-Unit	Computerteil, welcher für die Ausführung von Anweisungen zuständig ist.
Chatbot	Ein Chatbot ist ein Dialogsystem, welches das Schreiben mit einem technischen System erlaubt. ¹⁹¹
Cloud-Services	Direkt abrufbare Computer-Ressourcen, welche nicht aktiv vom Nutzer verwaltet werden.

¹⁸⁹ „Was ist ein Algorithmus?“¹⁹⁰ „Weights and Biases in machine learning | H2O.ai Wiki“.¹⁹¹ „Was ist ein Chatbot und wie funktioniert er?“

Content-Management-Systems	Software, mit der Benutzer Websites erstellen und Inhalte ohne technische Kenntnisse verwalten können. ¹⁹²
Data-Cleaning	Data Cleaning identifiziert Fehler, Duplikate sowie irrelevante Daten und entfernt diese aus dem Datenset. ¹⁹³
DataFrame	Ein Data Frame besteht aus mehreren <i>Pandas Series</i> , die durch einen Wert indexiert sind.
Deployment	Die Distribution einer Software für eine Zielgruppe.
DOM	Das Document Object Model ermöglicht Entwicklern auf standardisierte Weise den Zugriff auf die Strukturen von HTML- und XML-Dokumenten. ¹⁹⁴
Entrypoint	Gibt den Befehl an, der am Start eines Docker Containers ausgeführt wird.
Epoche	Anzahl der Durchläufe des gesamten Datensatzes während des Trainings. ¹⁹⁵
Fine-Tuning	Anpassung der LLM. Kann auch als Trainingsmethode verstanden werden, bei der alle Parameter des Modells angepasst werden.
Flag	Spezifiziert zusätzliche Optionen für einen Befehl.
Framework	Ein Framework ist ein Entwicklungsrahmen, welcher die Architektur der Software bestimmt. ¹⁹⁶
Funktion	Eine Funktion ist ein Codeblock, der verschiedene Aufgaben ausführt. Bei Bedarf kann er mehrmals aufgerufen werden. ¹⁹⁷
Funktionales Programmieren	Funktionales Programmieren zielt darauf ab, Programmieraufgaben mit Hilfe von Funktionen und deren Anwendung zu lösen. ¹⁹⁸
Gewichte	Spezifische Werte zwischen Beziehungen innerhalb einer LLM, die während des Trainings angepasst werden. ¹⁹⁹

¹⁹² „What Is a Content Management System (CMS)?“

¹⁹³ „Data Cleaning“.

¹⁹⁴ chriksikraus, „Was ist das DOM?“

¹⁹⁵ „What Is Epoch in Machine Learning?“

¹⁹⁶ „Was ist ein Framework?“

¹⁹⁷ „Was ist eine Funktion in der Programmierung? Wir erklären“.

¹⁹⁸ „Funktionale Programmierung“.

¹⁹⁹ „Weights and Biases in machine learning | H2O.ai Wiki“.

Graphics-Proces-sing-Unit	Prozessor, welcher speziell für die Berechnung von Grafiken entwickelt wurde.
Halluzination	Generation von falschen Informationen, die nicht auf Fakten basieren. ²⁰⁰
HTTP	Internetprotokoll, das die Kommunikation im Web ermöglicht.
Hyperparameter	Einstellungen, welche vor dem Training festgelegt werden und dieses steuern. ²⁰¹
Instruction-follo-wing-Dataset	Ein Instruction-Following-Dataset ist ein Datensatz, das Anweisungen für Aufgaben hat und die entsprechenden Ergebnisse enthält.
Interface	Eine Oberfläche, die es ermöglicht, mit einem Service zu kommunizieren.
JSON	JavaScript Object Notation, ist ein Datenformat in einfach lesbarer Form. ²⁰²
Klasse	Eine Vorlage, aus der Objekte erstellt werden. ²⁰³
Klassifizierung	Objekte werden anhand von Merkmalen in Gruppen eingeteilt.
Komponente	Komponenten sind unabhängige, wiederverwendbare Teile einer Benutzeroberfläche. ²⁰⁴
Large-Language-Model	Künstliche Intelligenzen, die darauf trainiert sind, Texte zu generieren und zu verstehen.
Logische Programmierung	Logische Programmierung ist ein Programmierparadigma, das auf mathematischer Logik beruht.
Machine-Learning	Lehre, die sich auf Daten und Algorithmen, welche zum Training von KI verwendet werden, bezieht.
Masking	Eine Methode, die angibt, welche Elemente einer Matrix verwendet werden sollen.
Natural-Language-Processing	Eine Lehre der künstlichen Intelligenz, die Maschinen menschlichen Text verstehen lässt.

²⁰⁰ „What Are AI Hallucinations?“

²⁰¹ „LLM Parameters Explained — The Cloud Girl“.

²⁰² „JSON“.

²⁰³ „Klassen Informatik“.

²⁰⁴ „Komponenten Und Props – React“.

Neuronales Netzwerk	Neuronale Netze versuchen, das menschliche Gehirn zu imitieren, um Probleme zu lösen. ²⁰⁵
Nvidia	US-amerikanisches Technologieunternehmen, das für die Entwicklung und Herstellung von GPUs bekannt ist.
Objekt-orientierte Programmierung	Objektorientierte Programmierung ist ein Programmierparadigma, bei dem das Softwaredesign auf Daten der Objekte basiert.
Open-Source	Software, die öffentlich gratis verfügbar ist und an welche öffentlich kontribuiert werden kann.
Overfitting	Keine akkurate Vorhersagen werden auf neue Daten getroffen.
Pandas Series	Pandas Series sind Objekte, die eindimensionalen Arrays entsprechen.
Parameter	Die Parameteranzahl setzt sich aus allen trainierbaren Gewichten zusammen. Unter Parametern können auch Übergabewerte verstanden werden. ²⁰⁶
Port-Mapping	Macht einen Port im Docker Container von außen zugänglich.
Promises	Promises sind ein Javascript Objekt für asynchrone Anfragen.
prozedurale Programmierung	Ist ein Programmierparadigma, dass sich durch die Definition und Nutzung von Prozeduren auszeichnet.
REST	Architektonischer Stil von APIs
Saubere Daten	„Saubere Daten bezeichnen Daten, die keine Ungenauigkeiten, Vollständigkeitsmängel, Inkonsistenzen, Irrelevanzen und Fehler enthalten.“ ²⁰⁷
Syntax	Regeln für die Struktur einer Programmiersprache. ²⁰⁸
Token	Kleinste Einheit, die eine LLM verarbeiten kann und umfasst in der Regel ein Wort, Wortteile oder Satzzeichen.
Tokenizer	Konvertiert Texte in Tokens. ²⁰⁹

²⁰⁵ „Was sind neuronale Netze? | IBM“.²⁰⁶ Brown u. a., „Language Models are Few-Shot Learners“, 8.²⁰⁷ OpenAI’s ChatGPT; Was sind saubere Daten im Bezug zu Datacleaning. Halte dich kurz. 18.03.2024²⁰⁸ „What is Syntax in Computer Programming? | Woz U“.²⁰⁹ „Tokenizers Explained – How Tokenizers Help AI Understand Language“.

Training	Prozess, in dem ein neuronales Netzwerk eine spezifische Aufgabe durch Daten lernt. ²¹⁰
Trainingsargumente	Parameter, welche die Huggingface-Trainer Klasse verwendet. ²¹¹
Transformer	Architekturstil für LLMs.
Umgebungsvariable /Environment-Variablen	Konfigurationsoptionen, die von Programmen verwendet werden.
Virtuelle Maschine	Virtuelles Betriebssystem auf einem physischen Hardwaresystem.
Video Random Access Memory	Spezieller kurzfristiger Speicher für Grafikkarten.
Webapplikation	Ein Softwareprogramm, das normalerweise auf einem Webserver ausgeführt wird. Man kann nur über einen Webbrower darauf zugreifen. ²¹²

²¹⁰ „What is AI training?“

²¹¹ „Trainer“.

²¹² „Webapplikationen und Webanwendungen - oneclick“.

13. Verzeichnisse

13.1. Abkürzungsverzeichnis

A

AGI	
Artificial General Intelligence.....	27
AI	
Artificial-Intelligence.....	23, 24, 25, 26, 27, 28, 35, 133
API	
Application Programming Interface 5, 6, 24, 25, 27, 29, 32, 37, 128, 130, 137, 140, 141, 142, 143, 148, 151, 156, 165, 170, 171, 173, 174, 182	
AWS	
Amazon Web Services.....	27, 133

B

BERT	
Bidirectional Encoder Representations from Transformers.....	31, 38
BLEU	
Bilingual Evaluation Understudy	18
BLOOM	
BigScience Large Open-science Open-access Multilingual.....	22, 30, 38, 45, 47, 49
bzw	
Beziehungsweise.....	7, 11, 20, 40, 43, 131, 139

C

CNN	
Convolutional Neural Networks.....	78, 79
CORS	
Cross-Origin Resource Sharing.....	137
CSS	
Cascading Style Sheets.....	150, 152, 156, 160, 161, 162, 163, 173
CUDA	
Compute Unified Device Architecture	117, 144, 145, 146, 147, 169
cURL	
Client for URL	140, 141, 142, 165

D

DOM	
Document Object Model.....	152, 153, 154, 155, 175

G

GPU	
Graphics Processing Unit	84, 89, 90, 91, 93, 117, 134, 139, 144, 164, 169

H

HTML

- HyperText Markup Language 149, 150, 152, 155, 173, 175
 - HTTP
Hypertext Transfer Protocol 137, 141, 149, 151, 156, 172, 176
-

J

JSON

- JavaScript Object Notation 50, 61, 63, 69, 70, 71, 72, 75, 76, 118, 142, 168, 171, 176
-

K

KI

- Künstliche Intelligenz 2, 52, 117, 168, 169, 176
-

L

LaMDA

- Language-Models for Dialog Applications 25, 32, 33, 37

LLM

- Large-Language-Model 3, 4, 6, 18, 20, 21, 22, 25, 29, 30, 31, 32, 33, 34, 35, 40, 41, 43, 44, 45, 47, 48, 77, 83, 84, 87, 88, 91, 98, 125, 134, 141, 144, 164, 165, 169, 174, 175, 177, 182

LoRA

- Low Rank Adaptation 90, 91, 95, 96, 97, 99, 100, 117, 121, 124, 128, 135, 141, 164, 168
-

M

MLM

- Masked Language Modeling 31

MMLU

- Massive Multitask Language Understanding 18
-

N

NLP

- Natural Language Processing 25, 28

NSP

- Next Sentence Prediction 31
-

O

OSP

- Objektstrukturplan 11
-

P

PaLM

- Pathways Language Model 25, 32, 33, 36, 37

PEFT

- Parameter-Efficient Fine-Tuning 87, 91, 168

PSP

Projektstrukturplan.....	9, 10
--------------------------	-------

R

REST	
Representational State Transfer	5, 6, 128, 140, 177
RLHF	
Reinforcement Learning from Human Feedback	88
RNN	
Recurrent Neural Networks	78, 80

S

SFT	
Supervised Fine-Tuning.....	87, 90, 91, 97, 100
SQL	
Structured Query Language	57, 59
SQuAD	
Stanford Question Answering Dataset.....	18, 43

T

TII	
Technology Innovation Institute	35

U

UI	
User-Interface	152
URL	
Uniform Ressource Locator.....	141, 142

V

VRAM	
Video Random Access Memory	84, 85, 87, 89, 90, 91, 164

X

XSS	
Cross-Site-Scripting-Angriffe	153

13.2. Autorenverzeichnis

Tabelle 8: Autorenverzeichnis

Kapitel	Autor
Impressum	Leonardo Golubovic
Julius Blum GmbH	Michael Blessing
Beschreibung der Diplomarbeit	Leonardo Golubovic
Projektmanagement	Irsad Celik
Evaluierung	Leonardo Golubovic
Data-Cleaning	Irsad Celik
Training der LLM	Michael Blessing
Deployment	Leonardo Golubovic
API-Webseitenanbindung	Irsad Celik
Fazit	Leonardo Golubovic
	Irsad Celik
	Michael Blessing
Verwendete Technologien	Leonardo Golubovic
	Irsad Celik
	Michael Blessing
Glossar	Leonardo Golubovic
	Irsad Celik
	Michael Blessing

13.3. Abbildungsverzeichnis

Abbildung 1: PL Leonardo Golubovic	1
Abbildung 2: PM Michael Blessing	1
Abbildung 3: PM Irsad Celik	2
Abbildung 4: AG Christian Gaisberger	2
Abbildung 5: PB Diethard Kaufmann	2
Abbildung 6: Partnerfirma Julius Blum GmbH	3
Abbildung 7: Projektauftrag	8
Abbildung 8: PSP	10
Abbildung 9: OSP	11
Abbildung 10: Projektzieleplan	13
Abbildung 11: Projektmeilensteinplan	14
Abbildung 12: Projektumweltanalyse	15
Abbildung 13: Zeiterfassung	16
Abbildung 14: Evaluierungsmethoden von Chatbots	19
Abbildung 15: Erfasste Antwortzeiten	38
Abbildung 16: Erfasste Bootzeiten	39
Abbildung 17: Scoring-Analyse der LLMs	45
Abbildung 18: Gewichtung der Bootzeiten	46
Abbildung 19: Gewichtung der Antwortzeiten	46
Abbildung 20: JSON-Datei	50
Abbildung 21: Blum-Quelldatei vor Data-Cleaning	64
Abbildung 22: Blum-Quelldatei nach Data-Cleaning	64
Abbildung 23: Zurückgegebene Produkte	76
Abbildung 24: CNN	79
Abbildung 25: RNN	80
Abbildung 26: Transformer	81
Abbildung 27: Vortrainingsablauf einer LLM	83
Abbildung 28: RHLF-Fine-Tuning	88
Abbildung 29: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 1	99
Abbildung 30: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 2	99
Abbildung 31: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 3	99
Abbildung 32: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 4	100
Abbildung 33: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 5	100
Abbildung 34: Entscheidungsfindung Trainingsmethode: Chatbot-Antwort 6	100
Abbildung 35: Optimale Hyperparameter Englisch: Chatbot-Antwort 1	102
Abbildung 36: Optimale Hyperparameter Englisch: Chatbot-Antwort 2	102
Abbildung 37: Optimale Hyperparameter Englisch: Chatbot-Antwort 3	103
Abbildung 38: Optimale Hyperparameter Deutsch: Chatbot-Antwort 1	105
Abbildung 39: Optimale Hyperparameter Deutsch: Chatbot-Antwort 2	105
Abbildung 40: Optimale Hyperparameter Deutsch: Chatbot-Antwort 3	106
Abbildung 41: Abgeschlossenes Training Englisch: Chatbot-Antwort 1	108
Abbildung 42: Abgeschlossenes Training Englisch: Chatbot-Antwort 2	109
Abbildung 43: Abgeschlossenes Training Englisch: Chatbot-Antwort 3	109
Abbildung 44: Abgeschlossenes Training Englisch: Chatbot-Antwort 4	109
Abbildung 45: Abgeschlossenes Training Englisch: Chatbot-Antwort 5	110
Abbildung 46: Abgeschlossenes Training Englisch: Chatbot-Antwort 6	110
Abbildung 47: Abgeschlossenes Training Englisch: Chatbot-Antwort 7	110
Abbildung 48: Abgeschlossenes Training Englisch: Chatbot-Antwort 8	111

Abbildung 49: Abgeschlossenes Training Englisch: Chatbot-Antwort 9	111
Abbildung 50: Abgeschlossenes Training Englisch: Chatbot-Antwort 10	111
Abbildung 51: Abgeschlossenes Training Deutsch: Chatbot-Antwort 1	112
Abbildung 52: Abgeschlossenes Training Deutsch: Chatbot-Antwort 2	113
Abbildung 53: Abgeschlossenes Training Deutsch: Chatbot-Antwort 3	113
Abbildung 54: Abgeschlossenes Training Deutsch: Chatbot-Antwort 4	113
Abbildung 55: Abgeschlossenes Training Deutsch: Chatbot-Antwort 5	114
Abbildung 56: Abgeschlossenes Training Deutsch: Chatbot-Antwort 6	114
Abbildung 57: Abgeschlossenes Training Deutsch: Chatbot-Antwort 7	114
Abbildung 58: Abgeschlossenes Training Deutsch: Chatbot-Antwort 8	115
Abbildung 59: Abgeschlossenes Training Deutsch: Chatbot-Antwort 9	115
Abbildung 60: Abgeschlossenes Training Deutsch: Chatbot-Antwort 10	115
Abbildung 61: Zusammenhang zwischen Dockerfiles, Docker Images und Docker Containern.....	130
Abbildung 62: Cluster-Architektur bei Kubernetes.....	131
Abbildung 63: Postman-Interface zur API-Abfrage.....	142
Abbildung 64: Chatbot-Website.....	148
Abbildung 65: Chatbot-Website.....	163
Abbildung 66: Visual Studio Code	166
Abbildung 67: Termius	166
Abbildung 68: Microsoft Teams	167
Abbildung 69: Microsoft Word	167
Abbildung 70: Discord	167
Abbildung 71: Notepad++	168
Abbildung 72: Huggingface	168
Abbildung 73: PEFT	168
Abbildung 74: Transformers	169
Abbildung 75: PyTorch	169
Abbildung 76: CUDA.....	169
Abbildung 77: Azure.....	170
Abbildung 78: PyCharm.....	170
Abbildung 79: Postman	170
Abbildung 80: JavaScript	171
Abbildung 81: Python.....	171
Abbildung 82: Pandas.....	171
Abbildung 83: React	172
Abbildung 84: Axios.....	172
Abbildung 85: CSS	173
Abbildung 86: Docker.....	173
Abbildung 87: OpenLLM.....	173

13.4. Code-Verzeichnis

Code-Abschnitt 1: Data-Cleaning-Funktion	66
Code-Abschnitt 2: Filterfunktion.....	67
Code-Abschnitt 3: Importierte Libraries	70
Code-Abschnitt 4: Pfadbestimmung	71
Code-Abschnitt 5: Einlesen der Daten, Fragen und Anweisungen	72
Code-Abschnitt 6: Komplette Data-Cleaning-Funktion.....	73
Code-Abschnitt 7: Daten-Extraktions-Funktion	74
Code-Abschnitt 8: Datenverarbeitung und Speicherung	75
Code-Abschnitt 9: Laden der Trainingsdatensätze	118
Code-Abschnitt 10: Laden des Llama-Modells.....	119
Code-Abschnitt 11: Konvertierung des Datensatzes ins Alpaca-Format	120
Code-Abschnitt 12: Vollständige Datensatztokenisierung.....	120
Code-Abschnitt 13: LoRA-Setup	121
Code-Abschnitt 14: Vorbereitung des Trainings	121
Code-Abschnitt 15: SafePeftModelCallback-Funktion.....	122
Code-Abschnitt 16: Hyperparameter-Konfiguration für deutsches Training.....	123
Code-Abschnitt 17: Hyperparameter-Konfiguration für englisches Training	124
Code-Abschnitt 18: Ausführen des LoRA-Trainings	124
Code-Abschnitt 19: Einbindung eines Adapters	125
Code-Abschnitt 20: OpenLLM Docker-Vorlage	134
Code-Abschnitt 21: Docker-Befehl für OpenLLM-Llama2-Container mit LoRAs.....	135
Code-Abschnitt 22: Dockerfile	136
Code-Abschnitt 23: Docker-Befehl mit provisorischem Image aus Dockerfile	138
Code-Abschnitt 24: Erstellen eines Docker-Images	138
Code-Abschnitt 25: Compose.yaml-Datei.....	139
Code-Abschnitt 26: cURL-Befehl zur API-Abfrage.....	141
Code-Abschnitt 27: Befehle zur vollständigen Treiber-Deinstallation.....	145
Code-Abschnitt 28: Befehle zur Installation der NVIDIA-Treiber.....	146
Code-Abschnitt 29: Befehle zur Installation der CUDA-Treiber.....	147
Code-Abschnitt 30: Befehle zur Installation des NVIDIA-Container-Toolkits	147
Code-Abschnitt 31: Beispiel für JSX-Tags	152
Code-Abschnitt 32: Importierte Libraries	156
Code-Abschnitt 33: Zustandsmanagement.....	157
Code-Abschnitt 34: sendMessage-Funktion	158
Code-Abschnitt 35: User-Interface	159
Code-Abschnitt 36: CSS-1.....	160
Code-Abschnitt 37: CSS-2.....	161
Code-Abschnitt 38: CSS-3.....	162

13.5. Tabellenverzeichnis

Tabelle 1: Zeiterfassung	16
Tabelle 2: Hyperparameter-Definition 1	92
Tabelle 3: Hyperparameter-Definition 2	93
Tabelle 4: Hyperparameter-Definition 3	94
Tabelle 5: LoRA-Konfiguration-Definition 1	95
Tabelle 6: LoRA-Konfiguration-Definition 2	96
Tabelle 7: Glossar	174
Tabelle 8: Autorenverzeichnis.....	182

13.6. Quellenverzeichnis

- „1. Introduction — Installation Guide for Linux 12.4 documentation“. Zugegriffen 30. März 2024.
<https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#removing-cuda-tk-and-driver>.
- „4_NoSQL_Document_storage_v1.4.pdf“, o. J.
„About“. Zugegriffen 22. März 2024. <https://openai.com/about>.
- Almazrouei, Ebtesam, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, u. a. „The Falcon Series of Open Language Models“. arXiv, 29. November 2023. <http://arxiv.org/abs/2311.16867>.
- Android Authority. „What Is Google LaMDA? Here’s What You Need to Know“, 16. April 2023.
<https://www.androidauthority.com/what-is-google-lamda-3313487/>.
- April 2019, Johanna Saladas Updated 27 February 2024 | Published 3. „What is cURL and how does it relate to APIs?“ IBM Developer. Zugegriffen 30. März 2024. <https://developer.ibm.com/articles/what-is-curl-command/>.
- AskHandle. „The Essential Role of Data Cleaning in Chatbot Training“, 22. November 2023.
<https://www.askhandle.com/blog/data-cleaning-in-chatbot-training>.
- AskHandle. „The Essential Role of Data Cleaning in Chatbot Training“, 22. November 2023.
<https://www.askhandle.com/blog/data-cleaning-in-chatbot-training>.
- Baseten. „A Guide to LLM Inference and Performance“, 17. November 2023. <https://www.baseten.co/blog/llm-transformer-inference-guide/>.
- „BentoCloud: Streamline the Path to Production AI“. Zugegriffen 29. März 2024. <https://bentoml.com/cloud>.
- BentoML Documentation. „Get Started“. Zugegriffen 29. März 2024. <http://docs.bentoml.com/bentocloud/get-started.html>.
- BentoML Documentation. „Introduction“. Zugegriffen 29. März 2024. <http://docs.bentoml.com/get-started/introduction.html>.
- „BET-PM 07a Projektumweltanalyse FRB 2021.pdf“, o. J.
- „BET-PM 08 PM Projektplanung FRB 2021.pdf“, o. J.
- „BET-PM 11 PM Terminplanung FRB 2022.pdf“, o. J.
- „BigScience Research Workshop“. Zugegriffen 24. März 2024. <https://bigscience.huggingface.co/>.
- „bigscience/bloom · Hugging Face“. Zugegriffen 24. März 2024. <https://hugging-face.co/bigscience/bloom>.
- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, u. a. „Language Models are Few-Shot Learners“. arXiv, 22. Juli 2020.
<http://arxiv.org/abs/2005.14165>.
- Brownlee, Jason. „Understand the Impact of Learning Rate on Neural Network Performance“. *MachineLearningMastery.Com* (blog), 24. Jänner 2019. <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>.
- Business Systemhaus AG. „Was ist ein Framework? Definition & Erklärung - BSH AG“. Zugegriffen 1. April 2024. <https://bsh-ag.de/it-wissensdatenbank/framework/>.
- Butler, Sydney. „GPT 3.5 vs. GPT 4: What’s the Difference?“ How-To Geek, 31. März 2023.
<https://www.howtogeek.com/882274/gpt-3-5-vs-gpt-4/>.
- BWL-Lexikon.de. „▷ Nutzwertanalyse » Definition, Erklärung & Beispiele + Übungsfragen“. Zugegriffen 31. März 2024. <https://www.bwl-lexikon.de/wiki/nutzwertanalyse/>.
- „ChatGPT“. Zugegriffen 22. März 2024. <https://openai.com/chatgpt>.
- chrissikraus. „Was ist das DOM?“ Dev-Insider, 25. März 2022. <https://www.dev-insider.de/was-ist-das-dom-a-ddd08045680129a6b2e00d2dcf6bf40e/>.
- CircleCI. „Making HTTP Requests with Axios“, 19. Mai 2022. <https://circleci.com/blog/making-http-requests-with-axios/>.
- „Custom ChatGPT - AI Chatbot for Your Website | Botsonic“. Zugegriffen 19. März 2024. <https://writersonic.com/botsionic>.
- „Data Cleaning: Definition, Benefits, And How-To | Tableau“. Zugegriffen 1. April 2024.
<https://www.tableau.com/learn/articles/what-is-data-cleaning>.

- „Data Cleaning: Definition, Techniken, Bedeutung in der Data Science“. Zugegriffen 8. Februar 2024.
<https://datascientest.com/de/data-cleaning>.
- „Data Collator“. Zugegriffen 18. März 2024. https://huggingface.co/docs/transformers/main_classes/data_collator.
- DataRobot AI Platform. „Introduction to Optimizers“. Zugegriffen 19. März 2024. <https://www.datarobot.com/blog/introduction-to-optimizers/>.
- „Der Unterschied zwischen KI-Inferenz und Training: Was ist KI-Inferenz?“ Zugegriffen 28. Februar 2024. <https://www.cloudflare.com/de-de/learning/ai/inference-vs-training/>.
- Docker Documentation. „Dockerfile Reference“, 34:41 + +0100 100 n. Chr.
<https://docs.docker.com/reference/dockerfile/>.
- Docker Documentation. „Overview of Best Practices for Writing Dockerfiles“, 46:27 + +0100 100 n. Chr. https://docs.docker.com/develop/develop-images/dockerfile_best-practices/.
- Docker Documentation. „What Is a Container?“, 39:35 + +0530 530.
<https://docs.docker.com/guides/docker-concepts/the-basics/what-is-a-container/>.
- Docker Documentation. „What Is an Image?“, 39:35 + +0530 530.
<https://docs.docker.com/guides/docker-concepts/the-basics/what-is-an-image/>.
- Docker Documentation. „Why Use Compose?“ <https://docs.docker.com/compose/intro/features-uses/>.
- Drost, Dorian. „Different Ways of Training LLMs“. Medium, 21. Juli 2023. <https://towardsdatascience.com/different-ways-of-training-langs-c57885f388ed>.
- Duca, Angelica Lo. „What Are Steps, Epochs, and Batch Size in Deep Learning“. *Syntax-Error* (blog), 27. Juni 2022. <https://medium.com/syntaxerrorpub/what-are-steps-epochs-and-batch-size-in-deep-learning-5c942539a5f8>.
- Editor, Content. „What Is Batch Size: Artificial Intelligence Explained“. *Chatgptguide.Ai* (blog), 26. Februar 2024. <https://www.chatgptguide.ai/2024/02/26/what-is-batch-size-artificial-intelligence-explained/>.
- Enterprise AI. „What Are Recurrent Neural Networks and How Do They Work?“ Zugegriffen 17. Februar 2024. <https://www.techtarget.com/searchenterpriseai/definition/recurrent-neural-networks>.
- Enterprise AI. „What Is a Convolutional Neural Network (CNN)?“ Zugegriffen 17. Februar 2024.
<https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>.
- Enterprise AI. „What Is a Transformer Model? | Definition from TechTarget“. Zugegriffen 17. Februar 2024. <https://www.techtarget.com/searchenterpriseai/definition/transformer-model>.
- „Evaluating Large Language Models: Methods, Best Practices & Tools | Lakera – Protecting AI teams that disrupt the world.“ Zugegriffen 29. März 2024. <https://www.lakera.ai/blog/large-language-model-evaluation>.
- „Falcon LLM“. Zugegriffen 26. März 2024. <https://falconllm.tii.ae/falcon.html>.
- „Firmenmitglieder Blum – Julius Blum GmbH | Bundesverband Industrie Kommunikation e.V.“ Zugegriffen 18. März 2024. <https://bvik.org/firmenmitglieder/julius-blum-gmbh/>.
- freeCodeCamp.org. „Tokenizers Explained – How Tokenizers Help AI Understand Language“, 27. März 2024. <https://www.freecodecamp.org/news/how-tokenizers-shape-ai-understanding/>.
- Frogglew. „What Is Azure Machine Learning? - Azure Machine Learning“, 29. Jänner 2024.
<https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning?view=azureml-api-2>.
- GeeksforGeeks. „Python JSON“, 9. Dezember 2020. <https://www.geeksforgeeks.org/python-json/>.
- GeeksforGeeks. „Python Random Module“, 22. Juli 2021. <https://www.geeksforgeeks.org/python-random-module/>.
- GeeksforGeeks. „R Programming Language - Introduction“, 3. April 2020. <https://www.geeksforgeeks.org/r-programming-language-introduction/>.
- „Generation with LLMs“. Zugegriffen 24. März 2024. https://huggingface.co/docs/transformers/main/llm_tutorial.
- „Getting Started with Docker - Images and Containers“. Zugegriffen 28. März 2024. <https://cultivatehq.com/posts/docker/>.

- GitHub. „Llama/MODEL_CARD.Md at Main · Meta-Llama/Llama“. Zugegriffen 14. Februar 2024.
https://github.com/meta-llama/llama/blob/main/MODEL_CARD.md.
- GmbH, Julius Blum. „Blum moving ideas“. Julius Blum GmbH. Zugegriffen 18. März 2024.
<https://www.blum.com/at/de/unternehmen/ueber-blum/>.
- Google. „Introducing PaLM 2“, 10. Mai 2023. <https://blog.google/technology/ai/google-palm-2-ai-large-language-model/>.
- Google AI for Developers. „Gemini API Pricing“. Zugegriffen 24. März 2024. <https://ai.google.dev/pricing>.
- Google AI for Developers. „PaLM API Overview“. Zugegriffen 24. März 2024.
https://ai.google.dev/palm_docs.
- „Google Bard API Documentation 2024: Developers Resources Guide“, 17. November 2023.
<https://aibard.online/bard-api-documentation/>.
- „Google Engineer Says Lamda AI System May Have Its Own Feelings“. 13. Juni 2022.
<https://www.bbc.com/news/technology-61784011>.
- „google-bert/bert-base-multilingual-cased · Hugging Face“, 11. März 2024. <https://huggingface.co/google-bert/bert-base-multilingual-cased>.
- „GPT-4“. Zugegriffen 23. März 2024. <https://openai.com/research/gpt-4>.
- Grant, Nico, und Cade Metz. „Google Releases Bard, Its Competitor in the Race to Create A.I. Chatbots“. *The New York Times*, 21. März 2023, Abschn. Technology. <https://www.nytimes.com/2023/03/21/technology/google-bard-chatbot.html>.
- „Home - Bard Now Gemini“, 17. November 2023. <https://aibard.online/>.
- „Hugging Face – The AI community building the future.“, 29. März 2024. <https://huggingface.co/>.
- „Installing the NVIDIA Container Toolkit — NVIDIA Container Toolkit 1.14.5 documentation“. Zugegriffen 30. März 2024. <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html#>.
- IONOS Digital Guide. „Was ist CSS? Definition und Anwendung“, 21. Juni 2022. <https://www.ionos.de/digitalguide/websites/webdesign/was-ist-css/>.
- IONOS Digital Guide. „Was ist ein Algorithmus? Definition und Beispiele“, 26. Oktober 2022.
<https://www.ionos.at/digitalguide/online-marketing/web-analyse/was-ist-ein-algorithmus/>.
- IONOS Digital Guide. „Was ist HTML (Hyper Text Markup Language)?“, 14. März 2023.
<https://www.ionos.at/digitalguide/websites/web-entwicklung/was-ist-html/>.
- „Jasper | AI Copilot for Enterprise Marketing Teams“. Zugegriffen 19. März 2024. <https://www.jasper.ai/>.
- „JavaScript: Funktion, Vor- & Nachteile - IT-Talents.de“. Zugegriffen 28. März 2024. <https://it-talents.de/it-wissen/javascript/>.
- „Javascript Promise – wenns funktioniert hat, dann ... | mediaevent.de“, 10. September 2023.
<https://www.mediaevent.de/javascript/promise.html>.
- „JSON“. Zugegriffen 1. April 2024. <https://www.json.org/json-de.html>.
- Kazim, Washija. „What Is Software Deployment? Process and Best Practices“. Zugegriffen 20. März 2024. <https://learn.g2.com/software-deployment>.
- Kinsta®. „What Is a Content Management System (CMS)?“, 18. Oktober 2023.
<https://kinsta.com/knowledgebase/content-management-system/>.
- „Komponenten Und Props – React“. Zugegriffen 1. April 2024. <https://de.reactjs.org/docs/components-and-props.html>.
- „LaMDA“. In Wikipedia, 5. März 2024. <https://en.wikipedia.org/w/index.php?title=LaMDA&oldid=1212032030>.
- Lead, The QA, und Andreea Draniceanu. „What Is Software Quality Management?“ The QA Lead, 14. Dezember 2023. <https://theqalead.com/quality-engineering-planning-strategy/what-is-software-quality-management/>.
- Li, Katherine (Yi). „How to Choose a Learning Rate Scheduler for Neural Networks“. neptune.ai, 22. Juli 2022. <https://neptune.ai/blog/how-to-choose-a-learning-rate-scheduler>.
- „LLM Parameters Explained — The Cloud Girl“. Zugegriffen 23. März 2024.
<https://www.thecloudgirl.dev/blog/llm-parameters-explained>.

- „„Löcher‘ von „Louis Sachar‘ - Buch - ,978-3-407-74098-4““. Zugegriffen 18. April 2024.
<https://www.thalia.de/shop/home/artikeldetails/A1001119323>.
- „LoRA“. Zugegriffen 20. März 2024. https://huggingface.co/docs/peft/main/en/conceptual_guides/lora.
- Manyi. „More about LoraConfig from PEFT“. *Medium* (blog), 5. September 2023. <https://medium.com/@manyi.yim/more-about-loraconfig-from-peft-581cf54643db>.
- „Mastering Low-Rank Adaptation (LoRA): Enhancing Large Language Models for Efficient Adaptation“. Zugegriffen 17. März 2024. <https://www.datacamp.com/tutorial/mastering-low-rank-adaptation-lora-enhancing-large-language-models-for-efficient-adaptation>.
- MIT Technology Review. „Why Meta’s Latest Large Language Model Survived Only Three Days Online“. Zugegriffen 26. März 2024. <https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/>.
- Mudadla, Sujatha. „Weight Decay in Deep Learning.“ *Medium* (blog), 13. Dezember 2023.
<https://medium.com/@sujathamudadla1213/weight-decay-in-deep-learning-8fb8b5dd825c>.
- Netomi. „Platform“. Zugegriffen 22. März 2024. <https://www.netomi.com/platform>.
- „Nodes“. Zugegriffen 29. März 2024. <https://kubernetes.io/docs/concepts/architecture/nodes/>.
- NVIDIA Data Science Glossary. „What Is Pandas Python?“ Zugegriffen 18. März 2024.
<https://www.nvidia.com/en-us/glossary/pandas-python/>.
- NVIDIA Developer. „CUDA Toolkit 12.1 Downloads“. Zugegriffen 30. März 2024. <https://developer.nvidia.com/cuda-downloads>.
- Otten, Neri Van. „Pre-Trained Models Complete Guide [How To & 21 Top Models In PyTorch, TensorFlow & HuggingFace]“. Spot Intelligence, 13. Februar 2024. <https://spotintelligence.com/2023/10/13/pre-trained-models/>.
- P, Vera. „Was Ist React: Funktionen Verstehen Und Wie Man Es Für Die Moderne Webentwicklung Einsetzt“. *Hostinger Tutorials* (blog), 1. März 2024. <https://www.hostinger.de/tutorials/was-ist-react>.
- „paperswithcode/galai“. Jupyter Notebook. 2022. Reprint, Papers with code, 31. März 2024.
<https://github.com/paperswithcode/galai>.
- „Parameter-Efficient Fine-Tuning using 🤗 PEFT“. Zugegriffen 1. März 2024. <https://huggingface.co/blog/peft>.
- „peft/examples/int8_training/peft_bnb_whisper_large_v2_training.ipynb at main · huggingface/peft“. Zugegriffen 20. März 2024. https://github.com/huggingface/peft/blob/main/examples/int8_training/peft_bnb_whisper_large_v2_training.ipynb.
- Penedo, Guilherme, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, und Julien Launay. „The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only“. arXiv, 1. Juni 2023. <http://arxiv.org/abs/2306.01116>.
- Pham, Aaron, Chaoyu Yang, Sean Sheng, Shenyang Zhao, Sauyon Lee, Bo Jiang, Fog Dong, Xipeng Guan, und Frost Ming. „OpenLLM: Operating LLMs in production“. Python, Juni 2023.
<https://github.com/bentoml/OpenLLM>.
- . „OpenLLM: Operating LLMs in production“. Python, Juni 2023. <https://github.com/bentoml/OpenLLM>.
- Ph.D, Cameron R. Wolfe. „Understanding and Using Supervised Fine-Tuning (SFT) for Language Models“. Substack newsletter. *Deep (Learning) Focus* (blog), 11. September 2023. <https://cameronrwolfe.substack.com/p/understanding-and-using-supervised>.
- „Plans & Pricing - Jasper“. Zugegriffen 19. März 2024. <https://www.jasper.ai/pricing>.
- Platform9. „Kubernetes Concepts and Architecture“. Platform9, 28. Mai 2019. <https://platform9.com/blog/kubernetes-enterprise-chapter-2-kubernetes-architecture-concepts/>.
- Postman API Platform. „What Is Postman? Postman API Platform“. Zugegriffen 30. März 2024.
<https://www.postman.com/product/what-is-postman/>.
- „Production-Grade Container Orchestration“. Zugegriffen 28. März 2024. <https://kubernetes.io/>.

- „Projektzieleplan | Diplomarbeiten an Berufsbildenden Schulen | Bundesministerium für Bildung, Wissenschaft und Forschung“. Zugegriffen 25. März 2024. <https://www.diplomarbeiten-bbs.at/tools-aus-dem-pm/projektzieleplan>.
- Prompt Engineering. „The Black Box Problem: Opaque Inner Workings of Large Language Models“, 23. Oktober 2023. <https://promptengineering.org/the-black-box-problem-opaque-inner-workings-of-large-language-models/>.
- Prune. „Answer to „What does „learning rate warm-up“ mean?““ Stack Overflow, 1. Mai 2019. <https://stackoverflow.com/a/55942518>.
- Python documentation. „Os — Miscellaneous Operating System Interfaces“. Zugegriffen 18. März 2024. <https://docs.python.org/3/library/os.html>.
- Python.org. „What Is Python? Executive Summary“. Zugegriffen 14. Februar 2024. <https://www.python.org/doc/essays/blurb/>.
- „Research“. Zugegriffen 22. März 2024. <https://openai.com/research/overview>.
- Rotenberg, Raz. „What Is Gradient Accumulation in Deep Learning?“ Medium, 23. Jänner 2020. <https://towardsdatascience.com/what-is-gradient-accumulation-in-deep-learning-ec034122cfa>.
- „sambanovasystems/BLOOMChat-176B-v1 · Hugging Face“, 2. März 2024. <https://hugging-face.co/sambanovasystems/BLOOMChat-176B-v1>.
- Saxena, Mohini. „Low Level Programming Language Examples“. Medium (blog), 27. Oktober 2023. <https://medium.com/@mohinisaxena/low-level-programming-language-examples-9658f578f468>.
- Schnurr, Roland. „Was ist ein Projektauftrag?“ SIXSIGMABLACKBELT (blog), 20. Mai 2018. <https://www.sixsigmablackbelt.de/was-ist-ein-projektauftrag/>.
- sdgilley. „Introduction - Training“. Zugegriffen 20. März 2024. <https://learn.microsoft.com/en-us/training/modules/intro-to-azure-ml/1-introduction>.
- „SQL Data Cleaning Techniques for Accurate Analytics | Airbyte“. Zugegriffen 18. März 2024. <https://airbyte.com/data-engineering-resources/sql-data-cleaning>.
- „SQuAD - the Stanford Question Answering Dataset“. Zugegriffen 31. März 2024. <https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Oxygen.html>.
- StudySmarter. „Funktionale Programmierung: Vorteile, Beispiel, Sprachen“. Zugegriffen 1. April 2024. <https://www.studysmarter.de/schule/informatik/programmieren-basics/funktionale-programmierung/>.
- StudySmarter. „Klassen Informatik: Objekte, Definition & Bedeutung“. Zugegriffen 1. April 2024. <https://www.studysmarter.de/schule/informatik/programmieren-basics/klassen-informatik/>.
- Sumo Logic. „Discover What Software Deployment Is | Definition and Overview“. Zugegriffen 20. März 2024. <https://www.sumologic.com/glossary/software-deployment/>.
- Systems, SambaNova. „SambaNova Systems | Generative AI, Optimized for Enterprise and Government“. Zugegriffen 19. März 2024. <https://sambanova.ai>.
- „tatsu-lab/stanford_alpaca“. Python. 2023. Reprint, Tatsu's shared repositories, 1. April 2024. https://github.com/tatsu-lab/stanford_alpaca.
- Taylor, Ross, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, und Robert Stojnic. „Galactica: A Large Language Model for Science“. arXiv, 16. November 2022. <http://arxiv.org/abs/2211.09085>.
- „Technology Innovation Institute | Advanced Technology Research Institute | International Center for Technology Research | TII“, 26. Februar 2024. <https://www.tii.ae/>.
- „The Comprehensive Guide to LLM Evaluation“. Zugegriffen 31. März 2024. <https://www.air-train.ai/blog/the-comprehensive-guide-to-llm-evaluation>.
- „The Stanford Question Answering Dataset“. Zugegriffen 18. April 2024. <https://rajpurkar.github.io/SQuAD-explorer/>.
- TheKnowledgeAcademy. „What Is Data Cleaning? Process and Tools“. Zugegriffen 21. Februar 2024. <https://www.theknowledgeacademy.com/blog/data-cleaning/>.
- Thoppilan, Romal, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, u. a. „LaMDA: Language Models for Dialog Applications“. arXiv, 10. Februar 2022. <http://arxiv.org/abs/2201.08239>.

- Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, u. a. „Llama 2: Open Foundation and Fine-Tuned Chat Models“. arXiv, 19. Juli 2023. <http://arxiv.org/abs/2307.09288>.
- „Trainer“. Zugegriffen 23. März 2024. https://huggingface.co/docs/transformers/v4.39.2/en/main_classes/trainer#transformers.TrainingArguments.
- Tuama, Daragh Ó. „What Is A Programming Language?“ *Code Institute Global* (blog), 2. September 2022. <https://codeinstitute.net/global/blog/what-is-a-programming-language/>.
- UNext. „What Is Epoch in Machine Learning? | UNext“, 24. November 2022. <https://unext.com/blogs/machine-learning/epoch-in-machine-learning/>.
- vikancha-MSFT. „Azure N-Series GPU Driver Setup for Linux - Azure Virtual Machines“, 1. Februar 2024. <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/n-series-driver-setup>.
- „Was ist ein Chatbot und wie funktioniert er?“ Zugegriffen 1. April 2024. <https://www.moin.ai/was-ist-ein-chatbot>.
- „Was ist eine Funktion in der Programmierung? Wir erklären“. Zugegriffen 1. April 2024. <https://de.moyens.net/tech/was-ist-eine-funktion-in-der-programmierung-wir-erklaeren/>.
- „Was sind neuronale Netze? | IBM“. Zugegriffen 21. Februar 2024. <https://www.ibm.com/de-de/topics/neural-networks>.
- „Webapplikationen und Webanwendungen - oneclick“, 15. Jänner 2019. <https://oneclick-cloud.com/de/blog/trends/webapplikationen-webanwendungen/>.
- „Weights and Biases in machine learning | H2O.ai Wiki“. Zugegriffen 22. März 2024. <https://h2o.ai/wiki/weights-and-biases/>.
- „What Are AI Hallucinations? | Built In“. Zugegriffen 22. März 2024. <https://builtin.com/artificial-intelligence/ai-hallucination>.
- „What Are Large Language Models (LLMs)? | IBM“. Zugegriffen 22. März 2024. <https://www.ibm.com/topics/large-language-models>.
- „What Is a Kubernetes Cluster?“ Zugegriffen 28. März 2024. <https://www.redhat.com/en/topics/containers/what-is-a-kubernetes-cluster>.
- „What is a Vector?“ Zugegriffen 12. März 2024. <https://h2o.ai/wiki/vector/>.
- „What is AI training?“ Zugegriffen 23. März 2024. <https://www.linkedin.com/pulse/what-ai-training-andy-main-2f4oe>.
- „What is Overfitting? | IBM“. Zugegriffen 17. Februar 2024. <https://www.ibm.com/topics/overfitting>.
- „What is Syntax in Computer Programming? | Woz U“. Zugegriffen 1. April 2024. <https://woz-u.com/blog/what-is-syntax-in-computer-programming/>.
- ZDNET. „What Is Google’s Gemini AI Tool (Formerly Bard)? Everything You Need to Know“. Zugegriffen 24. März 2024. <https://www.zdnet.com/article/what-is-googles-gemini-ai-tool-formerly-bard-everything-you-need-to-know/>.