FH AACHEN - UNIVERSITY OF APPLIED SCIENCE

Fachbereich 9 - Medizintechnik und Technomathematik

Anforderungsanalyse zur Implementierung von Sprachsteuerungen in robotergestützten Systemen mithilfe von LLMs

Frederic Moritz Valentin von Altrock

Matrikelnummer: 3282608

Betreuer:
Prof. Dr. rer. nat. Volker Sander
David Kötter, M.Sc.

Abgabedatum: 5. Januar 2025

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema "Anforderungsanalyse zur Implementierung von Sprachsteuerungen in robotergestützten Systemen mithilfe von LLMs" selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war. Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhändigen.

Aachen, den 5. Januar 2025

Frederic von Altrock

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Die alternde Bevölkerung und der gleichzeitige Rückgang der verfügbaren Arbeitskräfte [IAB] erfordern innovative Ansätze, um die Produktivität in verschiedenen Branchen zu erhalten oder sogar zu steigern. Eine mögliche Lösung ist die Nutzung von Robotern zur Automatisierung von Produktionsprozessen. [MMB23] Dies wirft jedoch grundlegende Fragen auf, etwa zur sicheren Steuerung solcher Systeme, zur dynamischen Verarbeitung und Kategorisierung von Sprachbefehlen sowie zu sicherheitskritischen Aspekten der Mensch-Roboter-Kollaboration.

Die sicherheitskritischen Aspekte und Herausforderungen sind zentrale Anforderungen, die sich im Zuge einer solchen Implementierung ergeben, insbesondere bei der dynamischen Interpretation und Verarbeitung von Sprachbefehlen in spezifischen Kontexten. Diese werfen in der Mensch-Roboter-Kollaboration sicherheitsrelevante Fragen auf, wie etwa die korrekte Zuordnung von Befehlen zu Funktionen und die Vermeidung von Fehlinterpretationen, um gefährlichen Situationen vorzubeugen.

1.2 Ziel

Das Ziel dieser Arbeit ist es, die Anforderungen zu analysieren, die bei der Integration von Large-Language-Modellen in die kollaborative Robotik entstehen, insbesondere im Kontext sprachgesteuerter und anderer multimodaler Befehlsgaben. Einerseits wird eine grundlegende Einführung zu Large-Language-Modellen gebeten, um die unterschiedlichen Modelle, Methoden und Vorgehensweisen verständlich aufzuzeigen. Andererseits liegt der Fokus darauf, die einzelnen Schritte und Abläufe in der Robotiksteuerung nachvollziehbar zu erläutern. Hierbei stehen insbesondere sicherheitskritische Fragen im Vordergrund, die sowohl die dynamische Interpretation und Kategorisierung von Sprachbefehlen durch das Large-Language-Modell als auch die Bewegungsplanung und Ausführung der Robotiksteuerung betreffen. Das Ziel dieser Seminararbeit ist es, die verschiedenen Anforderungen zu identifizieren und aufzuzeigen.

Kapitel 2

Stand der Technik

2.1 Robotiksteuerung

Die Steuerung ist das Herzstück eines Roboters und ist für die automatische und weitgehend autonome Ausführung von Aufgaben essentiell [PD19, S. 61]. In diesem Kapitel wird die Funktionsweise der Robotersteuerung, ihre Hauptkomponenten und die Zusammenarbeit mit den Motoren dargestellt. Im Fokus steht hierbei die *kollaborative Robotik*. Sie ist darauf ausgelegt, direkt und sicher mit Menschen zusammenzuarbeiten. Die Sicherheit wird über die in das Robotersystem integrierten Sensoren sichergestellt, die im potentiellen Kollisionsfall für eine Abschaltung des Roboters sorgen. [Wikc]

Die Hauptaufgaben der Robotiksteuerung lassen sich in folgende drei Punkte unterteilen:

1. Bewegungsplanung und Antriebsregelung zur Berechnung von *Trajektorien* und *Soll-Werten* für die Antriebe

2. Dynamikmodellierung unter Berücksichtigung der mechanischen Eigenschaften

3. Regelung zum Fehlerausgleich zwischen *Soll-* und *Ist-Werten*

Die Bewegungsplanung beginnt mit der Pfadberechnung des *Endeffektors*, wobei die vorgegebene Bewegung in kleine Abschnitte mittels des Interpolators in Bahnpunkte unterteilt wird, welche der Roboter dann nacheinander abfährt. Der *Endeffektor* bezeichnet hierbei ein Peripheriegerät, das am Ende des Roboterarms angebracht ist und die eigentliche Aufgabe ausführt.

Die Bahnpunkte werden anschließend von der Bewegungssteuerung anhand der *inversen kinematischen Transformation* in Sollgrößen umgerechnet. Diese Sollgrößen dienen der direkten Ansteuerung der Motoren, beispielsweise zur bestimmung der Gelenkpositionen, Beschleunigung oder Geschwindigkeit. Die Bewegungssteuerung überwacht kontinuierlich die gemessenen Gelenkpositionen, die von den Antrieben zurückgeliefert werden, und vergleicht diese mithilfe der *direkten kinematischen Transformation* mit den *Soll-Werten*, um sicherzustellen, dass die Abweichung akzeptabel bleibt. [PD19, S. 62–64] Die kinematische transformation beschreibt hierbei die Berechnung der Beziehung zwischen Position und Orientierung des Endeffektors im Raum und der Gelenkstellungen des

Roboters, wobei sie sich in der direkten und der inversen Kinematik unterscheidet. Die direkte Kinematik berechnet die Position und Orientierung des Endeffektors basierend auf den gegebenen Gelenkwinkeln oder Gelenkpositionen. Die inverse Kinematik berechnet die notwendigen Gelenkwinkel oder Gelenkpositionen, um eine vorgegebene Endeffektor Position zu erreichen. [Wikb] [Wika] Einen Überblick über den typischen Aufbau der Bewegungssteuerung gibt Abbildung 2.1:

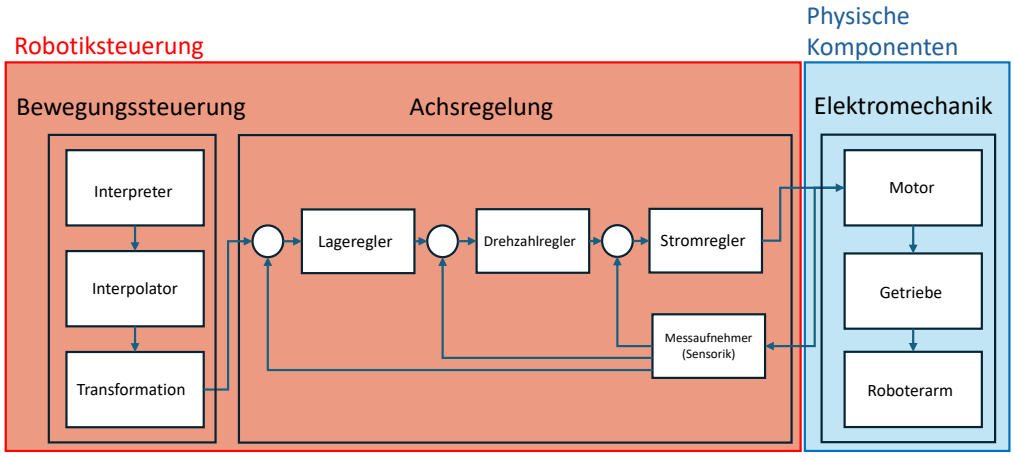

Abbildung 2.1: Aufbau der Bewegungssteuerung mit Achsregelung und Elektromechanik eines Industrieroboters. Abbildung angelehnt an [PD19, S. 63]

Die Dynamikmodellierung ergänzt den Bewegungsplan, indem sie die zu wirkenden Kräfte mit den berechneten Bewegungen verknüpft. Dabei wird zwischen der direkten und der inversen Dynamik unterschieden. Die direkte Dynamik beschreibt die Berechnung der Bewegungen, die aus den Antriebskräften und den Gelenkgeschwindigkeiten resultieren. Sie findet vor allem in Robotersimulationen Anwendung, da sich damit das Bewegungsverhalten eines Roboters ohne physische Ausführung analysieren lässt. Die inverse Dynamik beschreibt die generalisierten Kräfte, die erforderlich sind, um die geplante Bewegung zu realisieren. Die inverse Dynamik ist für ein reales System essentiell, da so die erforderlichen Kräfte für die Motoren berechnet und bereitstellt werden, um die genaue Bewegung auszuführen. [HSO20, S. 12] Die Regelung der Antriebe sorgt anschließend dafür, dass der Roboter die geforderte Bewegung ausführt. Die kontinuierlichen Rückmeldungen über die Gelenkpositionen und Gelenkgeschwindigkeiten werden genutzt, um die Steuerung der Motoren durch Vergleichen der *Soll-* und *Ist-Werte* zu überwachen [HSO20, S. 12–14]. [PD19, S. 62–64]

Nach der Definition der funktionalen Anforderungen der Robotersteuerung wird im folgenden auf die technischen Bestandteile eingegangen, die als nicht funktionale Anforderungen zu verstehen sind. Hierbei ergeben sich vier Bestandteile, welche essentiell für die Umsetzung der funktionalen Anforderungen sind: [PD19, S. 61]

1. Die Leistungselektronik
2. Der Steuerungscomputer
3. Das Handbediengerät
4. Die elektrischen Schnittstellen

Die Leistungselektronik ist direkt mit den im Roboter vorhandenen Motoren verbunden und ist so für den Antrieb der Motoren des Roboters und die Verarbeitung der Sensordaten verantwortlich. Für viele Roboter ist ein Frequenzumrichter zur Regelung der Motoren essentiell, da die Motoren servoelektrisch angetrieben werden müssen.

Der Steuerungscomputer ist für die Ausführung der Bewegungsplanung verantwortlich und ist mit einer Steuerungssoftware ausgestattet. Die aus der Bewegungsplanung resultierenden Sollwerte werden von der Steuerungssoftware auf dem Steuerungscomputer verwendet, um die erforderlichen Kräfte und Momente der resultierenden Bewegungen zu berechnen. Später werden die endgültigen Motorbefehle an die Leistungselektronik übergeben, welche diese in elektrische Signale zur Steuerung der Motoren umwandelt.

Das Handbediengerät dient der Mensch-Maschine Schnittstelle. Damit kann der Endbenutzer den Roboter über Touchscreen oder angebrachten Knöpfen kontrollieren und steuern. Es muss robust genug sein, um unter den gegebenen oder vorhandenen Umgebungsbedingungen zuverlässig zu funktionieren.

Die elektrische Schnittstelle ist die Verbindungsschnittstelle für Peripheriegeräte, Endeffektoren und die Sicherheitstechnik, wie den Not-Aus-Schalter oder Lichtschranken. [PD19, S. 61–64]

2.2 Grundlagen zu Large Language Modellen und deren Funktionsweise

Moderne Robotiksysteme erfordern nicht nur präzise Steuerungsmechanismen, sondern auch fortschrittliche Technologien wie *Large-Language-Modelle*. Im Folgenden werden deren Grundlagen und Funktionsweise erläutert. *Foundation Models* sind große, vortrainierte Modelle, die mithilfe der *Deep-Learning* Technologie auf umfangreichen und vielfältigen Datensätzen trainiert worden sind [Pata]. *Deep-Learning* erlaubt es dem Modell, über mehrere Verarbeitungsebenen hinweg die Darstellung von Daten über komplexe Muster zu lernen [LBH15, S. 1]. Die zugrundeliegende Technologie hinter *Foundation Models* ist die *Transformer-Architektur*, welche 2017 in dem Paper "Attention Is All You need" vorgestellt wurde [Vas+17]. Die von A. Vaswani et al. angebrachte Verarbeitung natürlicher Sprache mittels einer vollständig auf *Attention* basierenden Architektur war wegweisend und bildet so die Grundlage für moderne *Foundation Models* [Vas+17,

S. 1]. Die *Transformer-Architektur* ist auch Grundlage für viele spezialisierte Anwendungen und so in der Lage, auf eine breite Palette von Aufgaben angewendet zu werden, ohne dass sie für jede spezifische Aufgabe neu trainiert werden muss [Pata]. Diese Fähigkeit ergibt sich aus der Kombination von Foundation Models, der Vektordatenbank und der semantischen Verarbeitung durch *Vektoreinbettungen* und wird auch als Zero-Shot Capabilities bezeichnet [IBMc]. Diese geben an, wie gut ein Modell darin ist, Aufgaben zu lösen oder Fragen zu beantworten, auf welche es nicht spezifisch trainiert worden ist [Fir+23, S. 3].

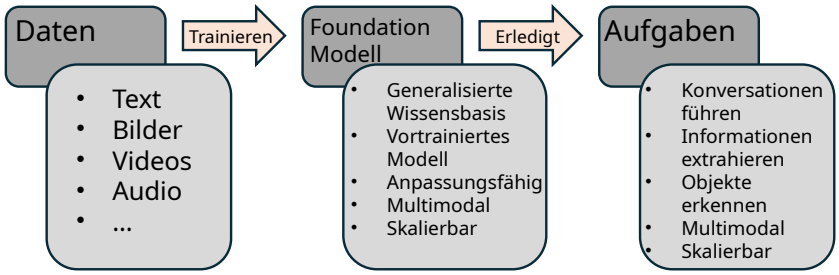

Abbildung 2.2: Die Abbildung zeigt, wie Foundation Modelle durch Training mit vielfältigen Daten entstehen und anschließend vielseitige Aufgaben erledigen können. Abbildung angelehnt an [Pata]

Eines der am weit verbreitetsten *Large-Language-Modells* ist beispielsweise GPT-3 (Generative Pre-trained Transformer), welches von *OpenAI* [Opeb] entwickelt wurde. Hierbei handelt es sich um ein *Natural-Language-Processing-Modell* zur natürlichen Sprachverarbeitung, welches mithilfe 570 Gigabyte Text aus dem Internet trainiert und für textbezogene Aufgaben wie Code- oder Textgenerierung, Übersetzung, Zusammenfassungen oder die Beantwortung von Fragen angepasst wurde [Sch][Bro+20, S. 8].

Large-Language-Modelle werden so nicht nur zur Verarbeitung und Generierung von Texten verwendet, sondern bilden zusätzlich auch die Grundlage für spezialisierte *Spracherkennungsmodelle*, welche in der Lage sind, menschliche Sprache zu verstehen und zu transkribieren, wie beispielsweise das Whisper-Modell von *OpenAI* [Opec]. Darüber hinaus werden *multimodale Anwendungen* vermehrt in der Bild- und Videoanalyse eingesetzt, wo sie visuelle Inhalte identifizieren und interpretieren können. Unter der multimodalität versteht man die Fähigkeit, dass unterschiedliche Arten von Daten, wie Bilder, Audio oder Text, gleichzeitig verarbeiten und analysieren zu können. [Ski]

Forscher stellten fest, dass Modelle mit zunehmender Größe bessere *Generalisierbarkeit* besitzen [Wan+24, S. 8]. Für die Robotik könnte dies von großem Interesse sein, da eine hohe *Generalisierbarkeit* elementar für Modelle zur besseren und genaueren Lösung unterschiedlichster Aufgabentypen ist. Man geht zusätzlich davon aus, dass sich durch die Kombination von *Sprachsteuerung* in Robotiksystemen mithilfe von *Large-Language-Modellen* eine intuitivere Interaktion zwischen Mensch und Maschine herstellen lässt als bisher bekannt [Pad+24]. Sie sind in der Lage, dynamischer und flexibler auf viele und auch neue Aufgaben zu reagieren und nicht nur vorprogrammierte Anweisungen auszuführen, wie es bisher bei der traditionellen Programmierung der Fall war.

Einige *Foundation-Modelle*, wie das GPT-Modell, basieren auf *künstlichen*

neuronalen Netzwerken [Vas+17, S. 2]. Ein *künstliches neuronales Netzwerk* ist ein Modell des maschinellen Lernens und trifft Entscheidungen ähnlich zu denen des menschlichen Gehirns, indem es mit dem *Attention-Mechanismus* die Gewichtung der Verbindungen von untereinander verknüpften Neuronen festlegt, welche angibt, wie stark ein Neuron das nächste beeinflusst [Vas+17, S. 3–4]. So kann die Grundidee der biologischen Neuronen nachgeahmt werden und bildet so das Herzstück von *Deep-Learning-Modellen* [Ses] [IBMa]. Erhaltenen Eingabedaten werden an die Neuronen in der Eingabeschicht weitergegeben [Ses]. Jede Verbindung erhält eine Gewichtung, welche den Einfluss zwischen den verbundenen Neuronen beschreibt [ITP]. Diese Gewichtungen werden für jede Verbindung zwischen den Neuronen summiert und der versteckten Schicht übergeben. Danach werden die berechneten Gewichtungen einer *Aktivierungsfunktion* übergeben, welche über die Aktivierung jenes Neuronen entscheidet. Die Aktivierung beschreibt hierbei die Bereitstellung und Weitergabe vom Signal, das als Input für die nächsten Neuronen dient. Dies wird so lange wiederholt, bis die Ausgabeschicht erreicht ist [ITP].

Bei dem Trainingsprozess eines solchen Netzwerkes kann das *Backpropagation-Verfahren* verwendet werden, welches die Gradienten berechnet, um dann über einen *Optimierungsalgorithmus* seine Fehler schrittweise zu reduzieren und zu optimieren. Dafür vergleicht das Verfahren das Ergebnis mit dem tatsächlich erwarteten Ergebnis und passt anhand dessen seine Gewichtungen an. [Ses] Folglich sorgt das zu präziseren Ausgaben [ITP].

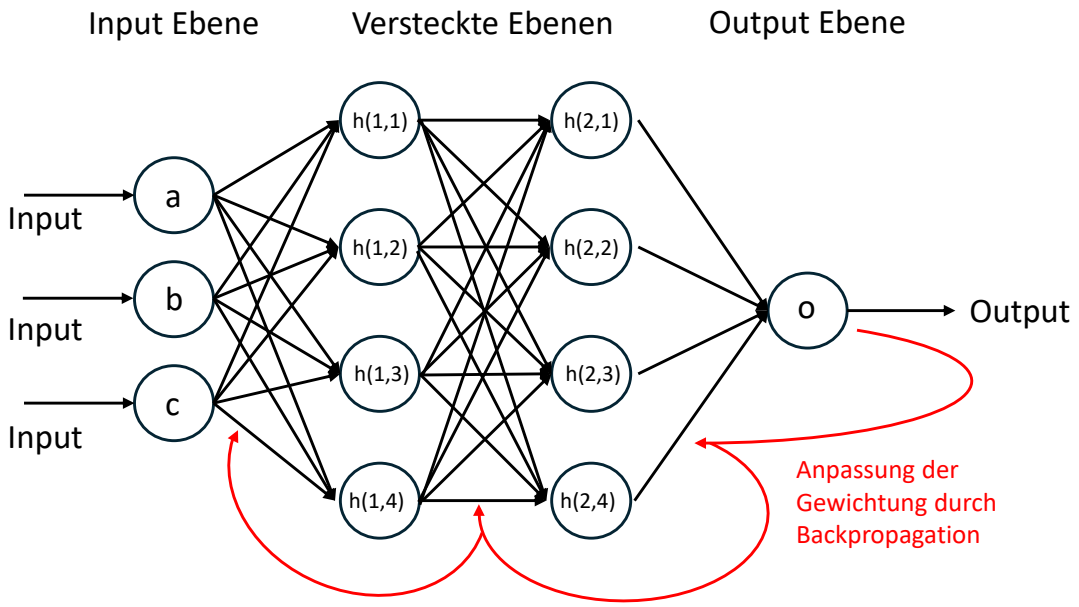

Abbildung 2.3: Darstellung eines tiefen neuronalen Netzwerks und des Backpropagation-Algorithmus. Die Pfeile verdeutlichen den Backpropagation-Prozess, bei dem der Fehler vom Output zur Input-Ebene zurückgerechnet wird, um die Gewichtungen entlang der Verbindungen zu optimieren.

Um über die generellen Trainingsprozesse von *Large-Language-Modellen* aufzuklären, wird im Folgenden auf die zentralen Komponenten im Trainingsprozess von *Large-Language-Modellen* eingegangen. Bevor so ein Modell trainiert werden kann, müssen die Rohdaten in eine für das Modell geeignete Form überführt

werden. Dies passiert im Prozess der Datenvorbereitung, die im wesentlichen aus folgenden zwei Schritten besteht:

1. Tokenisierung
2. Embedding-Prozess

Die *Tokenisierung* beschreibt den Prozess, in welchem die Wörter aus einem Text in *Token* umgewandelt werden, welche anschließend einem numerischen Index zugewiesen werden. Abhängig von der gewählten Methode und den verwendeten Kriterien, nach denen die Sätze in Wörter und folglich auch in *Token* umgewandelt werden, ergeben sich dadurch unterschiedlich große Vokabulare [Mun]. Eine anschauliche Darstellung dieses Prozesses, wie er vom OpenAI [Opeb] GPT3 Modell durchgeführt wird, zeigt Abbildung 2.4, in der ein Beispiel-Prompt in seine jeweiligen Token zerlegt wird.

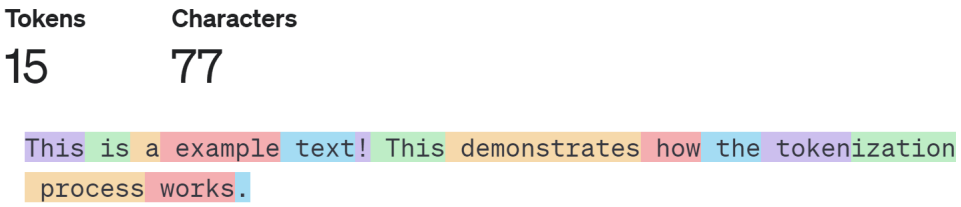

Abbildung 2.4: Visualisierung des Tokenisierungsprozesses vom GPT3 Modell. Der Beispiel-Prompt wird in einzelne Token zerlegt, wobei jedes Token einem numerischen Index zugeordnet wird.

Die Vorgehensweise bei der *Tokenisierung* hat also einen entscheidenden Einfluss auf die Leistung des Sprachmodells, da die Segmentierung der Wörter bestimmt, wie das Modell Sprache verarbeitet. Ein zu großes Vokabular benötigt eine sich stetig erhöhende Rechenleistung, die beim Lösen von Anfragen aufgebracht werden muss. Ein zu kleines Vokabular sorgt für ein hohes *Out-Of-Vocabulary* Auftreten [Yan24, S. 2–5]. Hierbei verdrängen die am häufigsten auftretenden Wörter die eher selteneren Wörter. Um ein passendes Mittelmaß beider Optionen zu erlangen, bietet sich das Konzept der *SubWord-Tokenisierung* an [Mun]. Zwei populäre Konzepte der *SubWord-Tokenisierung* sind *Byte-Pair Encoding*, welches unter anderem beim GPT-Modell von *OpenAI* [Opeb] verwendet wurde, oder aber auch die *WordPiece-Tokenisierung*, die beim BERT-Modell von *Google* [Goo] Verwendung fand [Tri] [Yeo].

Die den *Token* zugewiesenen numerischen Indexe werden im *Embedding-Prozess* verwendet, um den jeweiligen *Token* den passenden *Vektor-Embedding* zuzuordnen. Hierüber lassen sich mathematische Beziehungen zwischen Wörtern errechnen, wodurch sich die semantische Beziehung durch die Erkennung von *statischen Mustern* zwischen Wörtern mathematisch erfassen lässt [Pav]. Das *Embedding* beschreibt die Methode, Daten als Vektoren so darzustellen, dass semantische Beziehungen oder kontextbezogene Merkmale erfasst werden können.

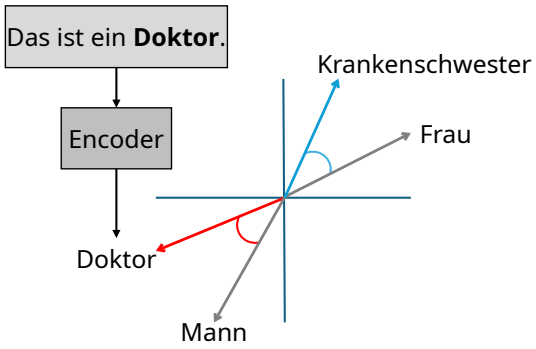

Abbildung 2.5: Vereinfachte Darstellung des Embedding-Prozesses im Vektorraum

Eine spezielle Form des *Embeddings* ist das *Word-Embedding*, wie beispielsweise *Word2Vec*. Durch Ableitungen der Beziehungen von Worten wird dafür eine passende Entsprechung identifiziert, was die Grundlage zur Ermittlung des nächsten Wortes in einem Text bildet. [Elab] [Elaa]

Während der *Embedding-Prozess* bereits grundlegende semantische Beziehungen zwischen den Wörtern erfasst, ermöglicht erst der *Self-Attention-Mechanismus* die eigentliche Erkennung von Mustern und Kontextbeziehungen innerhalb einer Sequenz. Im nächsten Schritt wird vom Prozess der Datenvorbereitung zum Trainingsprozess übergegangen, bei dem die kontextuelle Mustererkennung über *Attention-Mechanismen* stattfindet.

Der *Self-Attention-Mechanismus* verhilft den Modellen, Muster und Zusammenhänge aus Datensätzen zu erkennen. Es werden die verschiedenen Positionen einzelner Sequenzen betrachtet und in Beziehung zueinander gesetzt [Vas+17, S. 2]. Angelernt wird dieser Mechanismus nach dem *Self-Supervised-Learning* Prinzip, da hier keine explizit gelabelten Daten erforderlich sind. Das Modell wird mit ungelabelten Daten vortrainiert, um die entsprechenden Sprachstrukturen zu erkennen und zu erlernen, wobei die Labels aus den Daten selbst generiert werden [IBMb]. Damit soll die Sprache modelliert werden, indem die Wahrscheinlichkeit eines *Token*, basierend auf vorherigen *Token*, maximiert wird [Vas+17, S. 2–3]. Die *Multi-Head-Attention* ist eine darauf aufbauende Erweiterung, die es dem Modell ermöglicht, den *Self-Attention-Mechanismus* mehrfach parallel ausführen zu können [Vas+17, S. 4–5]. Kontextuelle Strukturen, grammatikalische Strukturen und semantische Beziehungen können so effektiver vom Modell verarbeitet werden [Fan+21, S. 4].

Um die in der *Pre-Training* Phase erzeugten kontextualisierten *Vektoreinbettungen* langfristig speichern und nutzen zu können, bietet sich eine Vektordatenbank an. Im Gegensatz zu traditionellen Datenbanken, die auf exakte Übereinstimmungen ausgelegt sind, ermöglichen *Vektordatenbanken* eine semantische Suche, bei der inhaltlich ähnliche und kontextbezogene Ergebnisse identifiziert werden können. Die durch den *Self-Attention-Mechanismus* und die *Multi-Head-Attention* erzeugten kontextualisierten *Vektoreinbettungen* repräsentieren die komplexen Muster, die das Deep-Learning-Modell erlernt hat. Diese werden in *Vektoreinbettungen* umgewandelt und können anschließend in *Vektordatenbanken* gespeichert werden [Int]. Die erzeugten Vektoren sind also als die numerische Darstellung der gelernten Muster zu verstehen, welche die Daten in hochdimensionalen Räumen repräsentieren. [Ion]

Optional kann nach dem *Pre-Training* vom Benutzer ein anwendungsspezifisches *Supervised-Fine-Tuning* durchgeführt werden. Im Vergleich zum *Self-Supervised-Learning* Prinzip wird hier mit gelabelten Datensätzen gearbeitet. Ziel ist es, die vom vortrainierten Modell gelernten Gewichtungen weiter zu verfeinern und an den spezifischen Anwendungsbereich anzupassen. Das vortrainierte Modell verfügt bereits über generalisierte Fähigkeiten, da es zuvor auf große Datensätze trainiert wurde. So werden also die Gewichtungen für ein spezielles Anwendungsgebiet weiter optimiert. [Sap] Der gesamte Ablauf wird noch einmal in Abbildung 2.6 veranschaulicht:

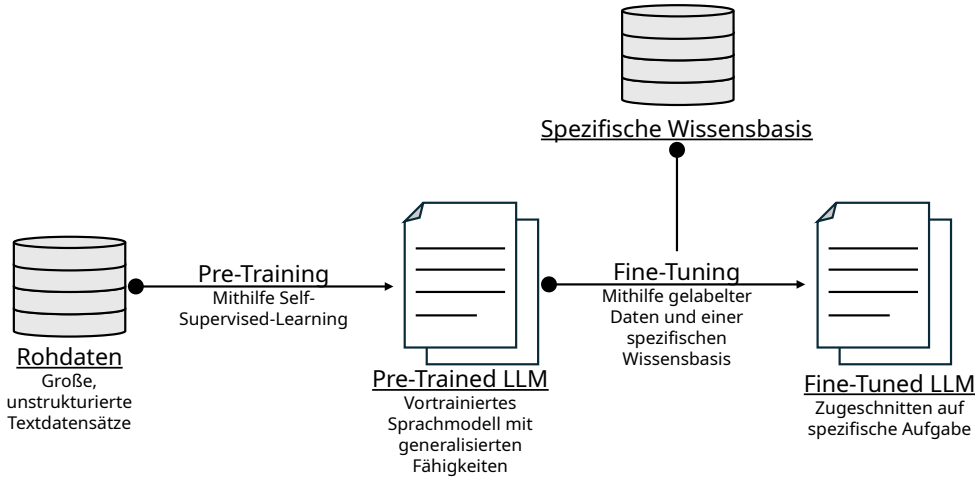

Abbildung 2.6: Vereinfachte Darstellung vom Pre-Training bis hin zum Fine-Tuned Modell.

2.3 Spezifische Modelle

Das folgende Kapitel geht auf die unterschiedlichen spezifischen Modelle ein, wobei hierfür als repräsentative Beispiele das GPT-Modell von *OpenAI* [Opeb] und das BERT-Modell von *Google* [Goo] zur Veranschaulichung hinzugezogen werden. Da beide Modelle auf zwei unterschiedliche *Transformer-Architekturen* zurückgreifen, kann man sie gut einander gegenüberstellen. Wo hingegen das GPT-Modell die *Decoder-Architektur* verwendet, greift das BERT-Modell auf die *Encoder-Architektur* zurück. [GG20, S. 10–12]

Die *Decoder-Architektur* des *Transformers*, wie sie im GPT-Modell verwendet wird, arbeitet *unidirektional*. Das Ziel bei der Entwicklung des GPT-Modells war die *Sprachmodellierung*, also die Spezialisierung auf die Vorhersage nachfolgender Wörter, basierend auf dem bisherigen Kontext [GG20, S. 11]. Die *Unidirektionalität* hierbei bedeutet, dass sich das Modell nur auf die vorangegangenen Wörter beziehen kann, nicht aber auf die Nachfolgenden. Die Eigenschaft wird durch das *Casual-Language-Modeling* Prinzip sichergestellt, bei dem das Modell autoregressiv trainiert wird, um den nächsten *Token*, basierend auf dem vorherigen, vorherzusagen [Vyk]. Das Modell greift nur auf den bisherigen Kontext zu und kann dadurch den Text schrittweise generieren. Die vorhergesagten Wörter werden als Input dem Modell erneut eingespeist und auf Basis dessen wird das nächste Wort vorhergesagt [GG20, S. 12]. So kann das Modell universell für Textgenerierung,

Übersetzung und Zusammenfassungsaufgaben eingesetzt werden, ohne dass es noch auf spezifische Aufgaben antrainiert werden muss [Opea].

Die *Encoder-Architektur* des *Transformers*, wie sie im BERT-Modell verwendet wird, arbeitet hingegen *bidirektional*. Das Ziel bei der Entwicklung des BERT-Modells war ein umfassendes Verständnis komplexer sprachlicher Zusammenhänge [Wol], also die Spezialisierung auf ein kontextuales Sprachverständnis [Dev+19, S. 1] [GG20, S. 10–11]. Die *Bidirektionalität* hierbei bedeutet, dass sich das Modell sowohl auf den vorherigen als auch auf den nachfolgenden Kontext eines Wortes beziehen kann [GG20, S. 11] [Aik].

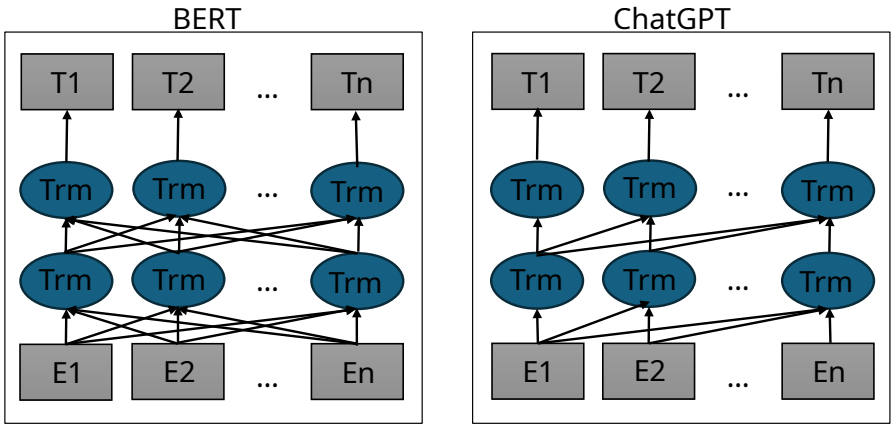

Abbildung 2.7: Vergleich der Architektur von BERT und ChatGPT: Transformer-Schichten (Trm) und Eingabe-Embeddings (En). BERT arbeitet Bidirektional, ChatGPT Unidirektional

Der *Masked-Language-Modeling* Mechanismus stellt das sicher. Hierbei werden dem Modell Texte mit Lücken vorgelegt, die basierend auf dem vollständigen Kontext vorherzusagen sind. So kann das Modell kontextbezogene Vorhersagen treffen und kann für die Beantwortung von Fragen, tiefem Textverständnis und semantischen Analysen verwendet werden. [Vyk][GG20, S. 10–11]

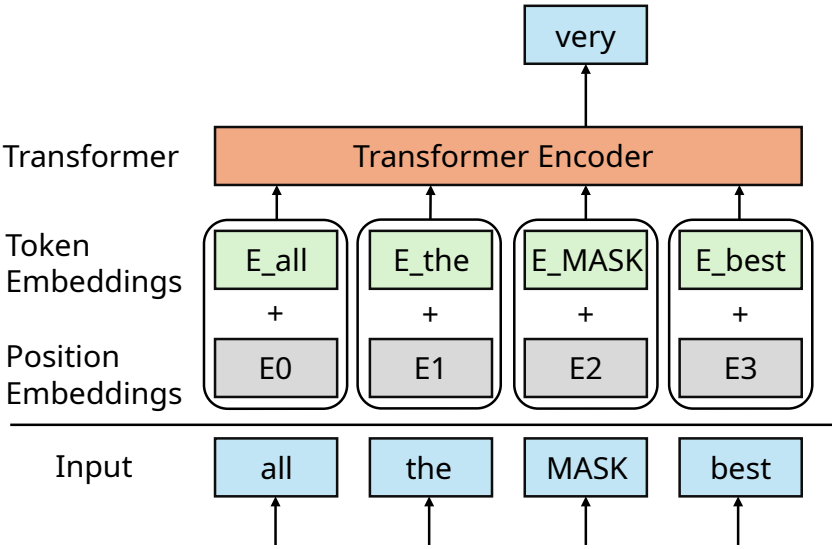

Abbildung 2.8: Visualisierung des Masked Language Modells. Das Modell zeigt den Verarbeitungsfluss von Input über Token- und Position-Embeddings bis hin zum Transformer. Die Maskierungen werden verwendet, um Vorhersagen für die maskierten Tokens basierend auf dem Kontext zu treffen.

Im Vergleich zum GPT-Modell besitzt es nicht vergleichbare *Generalisierungsfähigkeiten*. Um das Modell auf spezifische Aufgaben vorzubereiten, muss mithilfe von *Transfer-Learning* entsprechendes *Fine-Tuning* vorgenommen werden, um ein abgestimmtes Ergebnis zu erzielen. Das *Transfer-Learning* passt das Modell an spezifische Aufgaben an, indem die Gewichtungen des vortrainierten BERT-Modells angepasst und mittels *Fine-Tuning* über das *Backpropagation-Verfahren* aufgabenspezifisch optimiert werden. Beide Modelle greifen auf die *Unsupervised-Learning* Methoden zurück, wodurch die Modelle mit ungelabelten Daten antrainiert werden und so auf große Datensätze aus dem Internet zurückgegriffen werden kann. [GG20, S. 11–12] *Unsupervised-Learning* bietet so den Vorteil, dass die Daten, die zum Antrainieren verwendet werden, nicht vorher gelabelt werden müssen [Patb].

Kapitel 3

Anforderungen an die Robotiksteuerung

3.1 Sicherheitsaspekte

Die von *Isaac Asimov* formulierten drei *Gesetze der Robotik* bilden eine verwendbare Grundlage für das Verständnis von *Sicherheitsspezifikationen* in der Robotik: [Sic+08, S. 6]

1. Der Roboter soll keine Menschen verletzen oder es dem Benutzer ermöglichen, durch von ihm dem Roboter aufgetragene Handlungen verletzt zu werden.

2. Der Roboter muss die ihm aufgetragenen Befehle immer beachten, ohne dabei die erste Regel zu brechen.

3. Der Roboter soll sich selbst keinen Schaden zufügen, aber nur solange die erste und die zweite Regel bewahrt werden.

Bei den Regeln handelt es sich um fiktionale Regeln für das Verhalten von Robotern, doch illustrieren sie die potentiellen Risiken, auf die man bei der Zusammenarbeit zwischen Mensch und Roboter achten muss. [Sic+08, S. 6]

Damit Sicherheitsaspekte umgesetzt werden können, spielen vor allem Sensoren eine wichtige Rolle. Es muss analysiert werden, welche Art von Datensensoren für interne Zustände und die Umwelt benötigt werden. Für interne Zustände können Sensoren zur Positionserkennung eingesetzt werden. Für externe Zustände kommen Sensoren zur Kraftmessung oder Kameras zur Umgebungserkennung zum Einsatz. Nur so ist es möglich, auf unterschiedliche Einflüsse einzugehen und Sicherheitsvorkehrungen zu treffen. [Sic+08, S. 6] Die Wahl der geeigneten Sensorik spielt also für die *kollaborative Robotik* eine wichtige Rolle, welche ja darauf ausgelegt ist, direkt und sicher mit Menschen zusammenzuarbeiten. Der Autor Michael Hofbaur et al. bezeichnet hierfür die *multimodale Annäherungssensorik* zur Verwendung in kollaborativen Betriebsarten als besonders vielversprechend [HR19, S. 6]. Dabei bezieht er sich auf die Kombination verschiedener Sensortypen, um die Umgebung eines Roboters zu erfassen und seine Interaktionen sicher und effizient zu gestalten. Damit entsprechend auf unterschiedliche Ereignisse eingegangen werden kann, braucht es eine effektive

Verbindung zwischen Wahrnehmung und Aktion. Damit ist gemeint, dass die Verbindung zwischen Wahrnehmung und Aktion möglichst schnell und präzise sein muss, damit verlusstfrei und präzise gearbeitet werden kann. Die Informationen, die durch die Sensoren erhalten werden, müssen effektiv von der *Robotersteuerung* verarbeitet und an die Ausführungsschicht weitergeleitet werden, die für die Umsetzung der Steuerungsbefehle verantwortlich ist. [Sic+08, S. 6]

Im Bezug auf die *Kollisionsvermeidung* können auf Basis der *ISO TS 15066* [DINa] spezifisch definierte Grenzwerte für die *Kraft-* und *Druckeinwirkung* auf die unterschiedlichen Körperstellen herangezogen werden, um die Sicherheit bei kollaborativen Mensch-Roboter-Szenarien zu gewährleisten. Die in der *ISO TS 15066* [DINa] genannten Grenzwerte werden hier in zwei unterschiedliche Kategorien unterteilt. Die *statische Klemmsituation* beschreibt den andauernden Zustand, in welchem ein Körperteil zwischen dem Roboter und einer festen Umgebung eingeklemmt ist. Die einwirkende Kraft muss hierbei so gering sein, dass der Betroffene weiterhin in der Lage ist, sich selbst befreien zu können. Im Gegensatz dazu tritt die *transiente Stoßsituation* bei einem kurzzeitigen Kontakt zwischen Roboter und Mensch auf. Die Grenzwerte für *Kraft-* und *Druckeinwirkung* werden hierbei verdoppelt, da diese nur kurzfristig auftreten (vgl. Abbildung 3.1 [HR19, S. 2]). Eine Möglichkeit, dieser Sicherheitsanforderung gerecht zu werden, ist die kontinuierliche Überwachung der Antriebsmomente in den Gelenken. Dadurch können frühzeitig Abweichungen von den geplanten *Soll-Werten* oder normalen Betriebsbedingungen erkennt und daraus resultierende potentiell gefährliche Situationen verhindert werden. Ursprünglich wurde die *ISO TS 15066* [DINa] als Ergänzung zur *ISO 10218* [DINb] eingeführt, welche die Grundprinzipien für Industrieroboter und die damit zusammenhängenden zentralen kollaborativen Betriebsarten festlegt.

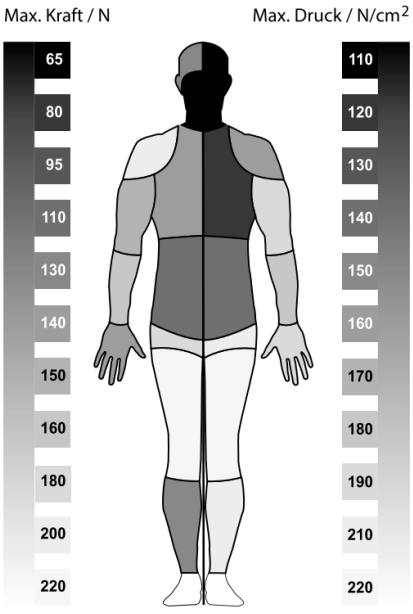

Abbildung 3.1: Grenzwerte für *statische Kraft-* und *Druckbelastungen* am menschlichen Körper zur Vermeidung von Schmerz und Verletzung. Abbildung aus [HR19, S. 2]

Diese umfassen folgende Optionen: [DINb]

1. Einen Sicherheitsbewerteten überwachten *Halt*
2. Die *Handführungsüberwachung*
3. Die *Geschwindigkeitsüberwachung*
4. Die *Abstandsüberwachung*
5. Die *Leistungsüberwachung*
6. Die *Kraftbegrenzung*

Testen und bewerten lässt sich die Roboterinteraktion unter gegebenen Bedingungen über die *biomechanische Validierung* von *Sensorfunktionen*. Hierbei werden die beim physischen Kontakt auftretenden Kräfte und Drücke messtechnisch erfasst und analysiert. Der *transiente Spitzenwert* zur Beurteilung eines *dynamischen Kontaktfalles* liegt innerhalb der ersten 500 ms, wo hingegen bei der *statischen Kontaktsituation* der *transiente Spitzenwert* für Zeiten größer als 500 ms definiert werden. [HR19, S. 3]

3.2 Bewegungsplanung und Steuerung

Um sicherzustellen, dass alle Bewegungspläne ohne Fehlertoleranz umgesetzt werden können, braucht es eine präzise vordefinierte *Trajektorienplanung*. Dem System kann so einen gewissen Spielraum an Parametern wie Geschwindigkeit oder Beschleunigung vorgegeben werden, welche manipuliert werden können, doch müssen die spezifischen Kräfte und Momente, die von den Antrieben aufgebracht werden, genau berechnet und abgestimmt sein. Die Komplexität hierbei liegt darin, die unterschiedlichen Bewegungen eines Gelenks zu bestimmen, wobei dies auch die Bewegungen anderer Gelenke beeinflusst. Obwohl dieser *Kopplungseffekt* die Bestimmung der Bewegungen erschwert, kann das Konzept der *Regelungstechnik* mit einem *geschlossenen Regelkreis* genutzt werden, um den aktuellen *Ist-Wert* mit dem gewünschten *Soll-Wert* zu vergleichen. Hierdurch können Abweichungen zwischen *Referenzwerten* und den von Sensoren erfassten *Ist-Werten* erkannt und entsprechend korrigiert werden. [Sic+08, S. 21]

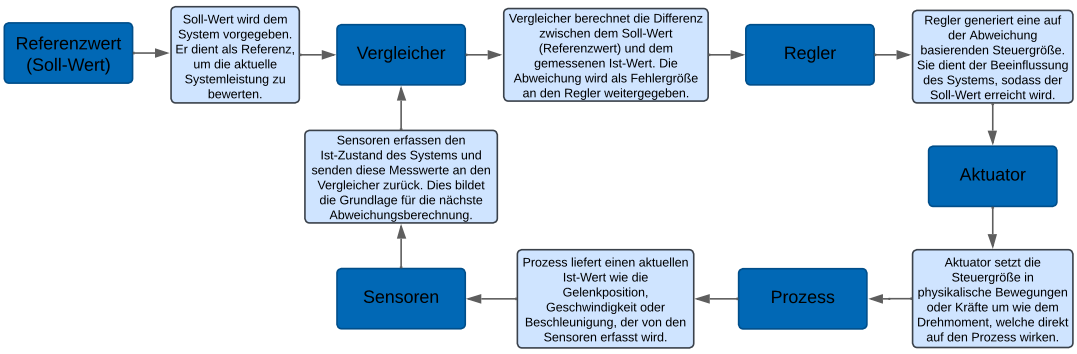

Abbildung 3.2: Darstellung eines geschlossenen Regelkreises aus der Regelungstechnik.

3.3 Mechanische Strukturen und Freiheitsgrade

In folgendem Abschnitt wird in Bezug auf die mechanische Gestaltung des Roboters auf die benötigten *Freiheitsgrade*, die sichere Konfiguration für *Gelenkbewegungen* und den Einfluss von vorab spezifizierten *Bewegungsabläufen* eingegangen werden, da diese entscheidend für seine Funktionalität und Sicherheit sind. Im Fokus soll hierbei die *Mensch-Roboter-Kollaboration* stehen. Die Anzahl der *Freiheitsgrade*, auch *Degrees-of-Freedom* genannt, ist entscheidend für die Fähigkeit des Roboters wie etwa präzise Bewegungsabläufe, das Erreichen schwer zugänglicher Positionen oder die Durchführung komplexer Aufgaben im vorgesehenen Anwendungsbereich. Da die Beweglichkeit eines *Manipulators* durch die Gelenke gewährleistet wird, stehen die spezifisch zu lösenden Aufgaben in Abhängigkeit zu der Anzahl an *Freiheitsgraden*. Hierbei stellt jedes Gelenk genau einen *Freiheitsgrad* bereit, unabhängig davon, was für ein spezifisches Gelenk verwendet wird. Zu den typischen Gelenktypen zählen beispielsweise *rotatorische Gelenke*, *prismatische Gelenke*, sowie weitere wie *Schrauben-, zylindrische, sphärische* oder *planare Gelenke*. [Sic+08, S. 7]

Laut dem Autor Bruno Siciliano et al. beträgt die Mindestanzahl der *Freiheitsgrade* zur freien Positionierung und Orientierung eines Objekts im dreidimensionalen Raum mindestens sechs. Drei *Freiheitsgrade* sind zur Positionierung vom Endeffektor erforderlich, und drei weitere dienen der Orientierung. Eine Erhöhung um weitere *Freiheitsgrade* zur Steigerung der Flexibilität des Roboters ist möglich, führt allerdings zu höheren Kosten und einer erhöhten Regelungskomplexität. Aus kinematischer Sicht führt dies zu Redundanz, wodurch die Komplexität weiter erhöht wird. Die tatsächliche Anzahl der *Freiheitsgrade* wird zusätzlich durch die mechanische Struktur des Roboters beeinflusst. Das hinzufügen eines zusätzlichen Gelenks führt also zu einem weiteren *Freiheitsgrad*, doch reduziert sich die Anzahl der *Freiheitsgrade* in geschlossenen Ketten aufgrund der *mechanischen Zwänge*, da sich die Gelenke gegenseitig beeinflussen wie in Abbildung 3.3 verdeutlicht. Dadurch ist die Gesamtanzahl der verfügbaren *Freiheitsgrade* kleiner als die Anzahl der Gelenke. [Sic+08, S. 7–8]

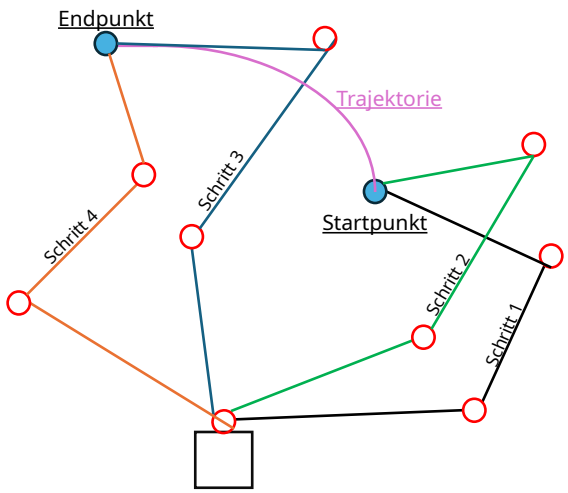

Abbildung 3.3: Veranschaulichung der Mechanischen Zwänge bei der Bewegungsplanung

Nicht nur die Anzahl der *Freiheitsgrade* eines Roboters ist in Bezug auf die Sicherheitsaspekte wichtig, sondern auch die Konfiguration der *Gelenkbewegungen*. Diese beeinflussen nicht nur die Effizienz oder die Präzision eines Roboters, sondern spielen auch in Bezug auf eine sichere Mensch-Roboter Interaktion eine entscheidende Rolle. Angesichts der in *ISO TS 15066* [DINa] genannten maximalen *transienten Kontaktkräfte*, die bei einer Mensch-Roboter Interaktion auftreten dürfen, muss eine entsprechende Optimierung der Gelenkkonfigurationen stattfinden. So kann sichergestellt werden, dass die Grenzwerte nicht überschritten werden. Zur Bewertung der Sicherheit wird im beschriebenen Experiment von Michael Hofbaur et al. [HR19] die auftretenden *transienten Kontaktkräfte* von zwei Gelenkkonfigurationen in einer simulierten *statischen Kontaktsituation* analysiert. Die *ISO TS 15066* [DINa] legt für ein *statisches Klemmen* der Hand Zeiträume von mehr als 500 ms fest, in denen die maximale Kraft- und Druckgrenze nicht überschritten werden darf. [HR19, S. 2–4]

Neben der Konfiguration der *Gelenkbewegungen* hat auch die gewählte Geschwindigkeit, zusammen mit den vorab spezifizierten *Bewegungsabläufen*, einen großen Einfluss auf die Erhöhung der Sicherheit in kollaborativen Roboterszenarien. Nach der *EN ISO 10218* [DINb] sind vorab spezifizierte *Bewegungsabläufe* und festgelegte, reduzierte Geschwindigkeiten zwingend erforderlich. So kann gewährleistet werden, dass die Grenzwerte für auftretende Kräfte in einem *Kollisionsfall* nicht überschritten werden. Diese Vorgaben stellen so auch eine Einschränkung der Flexibilität dar, da keine Anpassungen während der Mensch-Roboter Interaktion erlaubt sind. Ein weiteres Experiment von Michael Hofbaur et al. [HR19] mit einem *Universal Robot UR10* legt dies dar, wo ausgehend von einem als sicher bewerteten Ablagevorgang eine Veränderung des Ablageortes und der Geschwindigkeit einen exponentiellen Anstieg der *Kollisionskraft* zur Folge hat. Obwohl solche Anpassungen oftmals gewünscht sind, untermauert das durchgeführte Experiment die Bedeutung vorab definierter Parameter und deckt zugleich die begrenzten Möglichkeiten für dynamische Anpassungen in der Praxis auf. [HR19, S. 5]

3.4 Anforderungen

Abschließend fasst Abbildung 3.4 die funktionalen und nicht-funktionalen Anforderungen an die Robotiksteuerung übersichtlich zusammen.

Anforderungen	
Nicht Funktional	Funktional
• Qualitativ hochwertige, multimodale Sensorik • Biomechanische Validierung • ISO-Standards (TS 15066 & 10218) • Effizienz und Präzision • Komplexität der Steuerung hinsichtlich der Kopplungseffekte • Grenzwerte zur mechanischen Belastung • Möglichst geringe Anforderungen an die Rechenleistung	• Einhaltung der Gesetze der Robotik • Trajektorienplanung und auch Umsetzung • Wahrnehmung und Aktion mittels Sensorik und externer Daten • Kollisionsvermeidung • Geschwindigkeits- und Abstandsüberwachung • Gelenkkonfigurationen • Vorab definierte Bewegungsabläufe

Abbildung 3.4: Gegenüberstellung der funktionalen und der nicht-funktionalen Anforderungen an die Robotiksteuerung.

Kapitel 4

Anforderungen an das Large-Language-Modell

Um Large-Language-Modelle als übergeordnete Steuerungseinheit nutzen zu können, wird in der Anforderungsanalyse untersucht, welche Voraussetzungen und Eigenschaften erfüllt sein müssen. Dabei stehen insbesondere die Fähigkeit zur Verarbeitung und zum Verständnis natürlicher Sprache, die Interpretation bereitgestellter Daten, die Erstellung von Bewegungsplänen sowie die Möglichkeit zur kontinuierlichen Optimierung von Algorithmen durch Optimierungsschleifen im Fokus.

Die Wahl zwischen einem maßgeschneiderten *Large-Language-Modelle* und der Nutzung eines vortrainierten Modells hängt von Faktoren wie Anwendungsbereich, Kosten und verfügbarer Datenmenge ab, doch ist das Antrainieren eines eigenen Modells mit hohen Rechenressourcen, gut aufbereiteter Daten und folglich hohen Kosten verbunden [Zan]. Auch ist nicht immer gewährleistet, dass sich ausreichend qualitativ hochwertige Daten zum Antrainieren finden lassen, oder aber es ist mit einem sehr hohen Aufwand verbunden [Sha]. So bietet die Nutzung eines vortrainierten Modells, auch als *Off-The-Shelve-Modelle* bezeichnet, eine kostengünstigere und zugleich effiziente Alternative [Pad+24, S. 2].

Ein gutes vortrainiertes Modell zeichnet sich durch eine solide Grundlage aus, die durch Ansätze wie das *Unsupervised-Learning* oder auch dem *Self-Supervised-Learning* Prinzip erreicht werden kann. Hierfür werden große Mengen an unstrukturierten Daten benötigt, um einen hohen Generalisierungsgrad zu schaffen und als leistungsfähiges Model zu dienen.

Prinzipien wie das *Unsupervised-Learning* tragen ergänzend zu der umfangreiche Wissensbasis, die im Rahmen des *Pre-Trainings* aufgebaut wird, auch zur Entwicklung der *Zero-Shot-Fähigkeit* bei. Die Fähigkeit beschreibt, dass durch die Entwicklung einer breiten Wissensbasis und gelernten kontextueller Zusammenhänge das Modell Aufgaben bewältigen kann, die nicht explizit im Training berücksichtigt worden sind.

Um ein vortrainiertes Modell auf spezifische Robotikanforderungen anzupassen, lässt sich mithilfe von *Transfer-Learning* ein solches Modell dann spezifisch auf die gegebenen Anforderungen abstimmen. Die bestehenden Gewichtungen bleiben unverändert, um neue, fachspezifische Informationen effizient hinzuzufügen. Danach wird beim *Fine-Tuning* das neu antrainierte Wissen, auch durch die Veränderung der Gewichtungen, dem bereits existierenden Wissensstand mit

angefügt. [Cha]

Off-The-Shelve-Modelle bieten so eine gute Grundlage, doch erfordert es unabhängig von der Wahl des Modells weiterhin klare funktionale und nichtfunktional definierte Anforderungen. Diese definieren, welche Fähigkeiten das Modell aufweisen muss und welche technischen Voraussetzungen für eine erfolgreiche Implementierung erfüllt sein müssen. Zuerst werden im folgenden Text die funktionalen Anforderungen betrachtet:

1. Unterstützung für *Multimodalität* zur Verwertung von Bild- und Toninformationen
2. Kontextverständnis und Reduktion von Mehrdeutigkeit (*Bias*)
3. Befehlsinterpretation zur Handlungsplanung
4. Planung auf *High-Level-* und *Low-Level-Ebene*

4.1 Multimodalität

Damit Informationen aus unterschiedlichen Datenquellen gleichzeitig integriert werden können, muss der *Encoder* entsprechend angepasst werden. Diese Fähigkeit zur simultanen Verarbeitung wird durch *Multimodalität* unterstützt, die es Modellen ermöglicht, Inhalte aus unterschiedlichen Medienformaten besser zu verstehen und zu generieren [Zen+23, S. 3–4]. *Fusion-Encoder* ermöglichen Interaktionen zwischen *Modalitäten*, indem sie bereits angesprochene Techniken wie *Self-Attention* verwenden. *Dual-Encoder* hingegen kodieren die *Modalitäten* separat mithilfe getrennter *Single-Modal-Encoder* und berechnen danach die Ähnlichkeit durch einfache Skalarprodukte oder *flache Attention-Schichten*. Diese beiden Mechanismen helfen dem Modell, relevante Beziehungen zwischen den *Modalitäten* zu erkennen. [Wu+23, S. 3–4] Die Verbindung zwischen Sensorik und *Multimodalität* könnte insbesondere für die *kollaborative Robotik* von Bedeutung sein, da eine effektive Kommunikation zwischen Wahrnehmung und Umsetzung erforderlich ist, auch im Hinblick auf die Echtzeitfähigkeit des Systems.

4.2 Bias - Voreingenommenheit

Ähnlich wie bei vorab definierten Bewegungsabläufen in Robotersystemen, die durch vorab definierte Vorgaben Sicherheit gewährleisten soll, stellt die Minimierung der Voreingenommenheit eine zentrale Voraussetzung dar, um die Befehle eindeutig interpretieren zu können und dadurch falsche Entscheidungen aufgrund von Mehrdeutigkeit zu vermeiden. Daraus ergibt sich auch die zentrale Anforderung an das *Large-Language-Modell*, welches so entwickelt werden muss, dass sich die Voreingenommenheit in den Ergebnissen minimiert. Der Begriff *Bias* im Kontext von *Large-Language-Modellen* bezeichnet die Voreingenommenheit des Modells, die aus den zugrunde liegenden Trainingsdaten resultiert. Gallegos et al. [Gal+19] gibt eine ausführliche Definition des Begriffs *Bias* an, welche die Voreingenommenheit als ungleiche Behandlung oder ungleiche Ergebnisse zwischen

sozialen Gruppen beschreibt, die auf historischen und strukturellen Machtungleichgewichten basieren und zu Repräsentationsschäden und Diskriminierung führen [Gal+19, S. 2, 9] [JMH24, S. 2]. Die Voreingenommenheit stellt in diesem Kontext allerdings eine allgemeine Herausforderung dar, unabhängig davon, ob es sich um Diskriminierung im sozialen Kontext handelt. Diese Anforderung betrifft sowohl das grundsätzliche Antrainieren von einem solchen Modell als auch das *Fine-Tuning* auf spezifische Anwendungsfälle. Während des initialen Trainings von Modellen wie beispielsweise dem GPT-Modell von *OpenAI* [Opeb] oder dem BERT-Modell von *Google* [Goo] riesige unbearbeitete Datenmengen aus dem Internet verwendet werden, die außerhalb des direkten Einflussbereichs liegen, bietet das *Fine-Tuning* die Möglichkeit, spezifisches Wissen durch den gezielten Einsatz von manuell ausgewählten, hochwertigen Daten effizient zu integrieren. Die vom *Fine-Tuning* verwendeten Daten spielen eine zentrale Rolle, um spezifische Anforderungen bestehender *Large-Language-Modelle* zu adressieren. Über die bereitgestellten Daten kann direkter Einfluss darauf genommen werden, dass die Modelle keine weiteren Stereotypen, falsche Repräsentierungen, Doppeldeutigkeiten oder andere Vorurteile übernehmen. [Gal+19, S. 2] [JMH24, S. 2] Die Schwierigkeit liegt also darin, dass durch die generell entstandene Voreingenommenheit des Modells technische Entscheidungen getroffen oder priorisiert werden können, die so nicht optimal oder gewünscht sind. Daher ist eine kontinuierliche Prüfung und Verbesserung der Daten- und Modellqualität erforderlich, um sicherzustellen, dass die Modelle sowohl ethisch als auch technisch den Anforderungen gerecht werden.

4.3 Befehlsinterpretation und Handlungsplanung

Diese Fähigkeit steht in engem Zusammenhang mit der im ersten Punkt genannten Anforderung des Kontextverständnisses. Ein tiefgreifendes Verständnis des sprachlichen Kontextes ist die Voraussetzung dafür, dass das *Large-Language-Modell* Befehle korrekt interpretieren kann. Damit also ein *Large-Language-Modell* in der *kollaborativen Robotik* eingesetzt werden kann, ist ein tiefgreifendes Kontextverständnis essenziell. Das System muss am Ende in der Lage sein, sprachliche Befehle korrekt zu interpretieren, damit anschließend die ihm zur Verfügung stehenden Funktionen den Befehlen zugeordnet werden können. Das *Large-Language-Modell* muss auf Anweisungen wie “Räume den Raum auf.”, “Mach den Raum sauber! ” oder “Räume alle Sachen vom Boden auf.” mit der gleichen grundsätzlichen Aktion reagieren: Der Initiierung einer “Aufräumarbeit”, wie sie in Abbildung 4.1 dargestellt ist. Sprachliche Anweisungen müssen also unabhängig von ihrer Formulierung konsistent einer einheitlichen Aktion zugeordnet werden. Anschließend muss das *Large-Language-Modell* auf der *Planungsebene* die entsprechende Sequenz von Aufgaben planen, um diese später auf der *Ausführungsebene* umsetzen zu können. Die *Planungsebene* umfasst dabei die Erstellung einer logischen Abfolge von Aufgaben, während die *Ausführungsebene* für die konkrete Umsetzung dieser Aufgaben verantwortlich ist. Die zur Verfügung stehenden Befehle sind dabei fest definiert während variable Parameter wie Koordinaten weiterhin dynamisch vom *Large-Language-Modell* anpassbar sind.

4.4 High-Level-Planung und Low-Level-Umsetzung

Die *Multimodalität* und auch die Wahl der zu verwendenden Sensorik spielen auch auf der *High-Level-Ebene* eine zentrale Rolle, da das *Large-Language-Modell* die Informationen der verschiedenen Informationsquellen kombiniert und auswertet. Unter der *High-Level-Planung* versteht man die Ebene, auf der die strategischen Ziele in konkrete Schritte übersetzt werden. So können komplexe Befehle interpretiert und in eine logische Abfolge von auszuführenden Schritten unterteilt werden. Im Vergleich dazu erfolgt in der *Low-Level-Ebene* die konkrete Umsetzung dieser Schritte in die akute Ausführung. So können durch das gleichzeitige Verarbeiten der Informationen strategische Entscheidungen vom *Large-Language-Modell* getroffen werden, da flexibel auf unterschiedliche Umgebungsbedingungen reagiert werden kann und schließlich in die *High-Level-Planung* mit einfließt.

Zusätzlich zu der *Multimodalität* spielt so auch die Trajektorienplanung aus der Robotiksteuerung eine zentrale Rolle, in welcher die Bewegungsabläufe vorab definiert werden und so effizient umgesetzt werden können. Trotz der vorgegebenen Bewegungsabläufe steht dem Modell so ein begrenzter Spielraum an Parametern zur Verfügung, die er mithilfe der zusätzlichen Informationen aus der *multimodalen Sensorik* entsprechend anpassen kann. Diese Echtzeit-Umgebungsdaten, wie die Objektposition oder die Zielkoordinaten, können dann genutzt werden, um bestimmte Parameter innerhalb der vorgegebenen Bewegungsplanung anzupassen.

Die von der *High-Level-Ebene* angepassten Parameter, die auf den ausgewerteten *multimodalen Sensorikdaten* basieren, werden anschließend an die *Low-Level-Ebene* übergeben. Diese übernimmt ausschließlich die präzise Steuerung der Roboterachsen und die exakte Umsetzung der vorgegebenen Bewegungsabläufe. Dabei bleibt die Arbeitsweise der *Low-Level-Ebene* stets unverändert, unabhängig davon, wie stark die Bewegungsparameter zuvor durch die *High-Level-Ebene* angepasst worden sind. Die Anforderung an das *Large-Language-Modell* besteht daher nicht nur darin, strategische Entscheidungen zu treffen, sondern auch darin, sicherzustellen, dass die *multimodale Sensorik* einen direkten Einfluss auf diese Entscheidungen hat. Innerhalb eines festgelegten Rahmens ermöglicht dies die Anpassung der Bewegungsabläufe an aktuelle Umgebungsbedingungen.

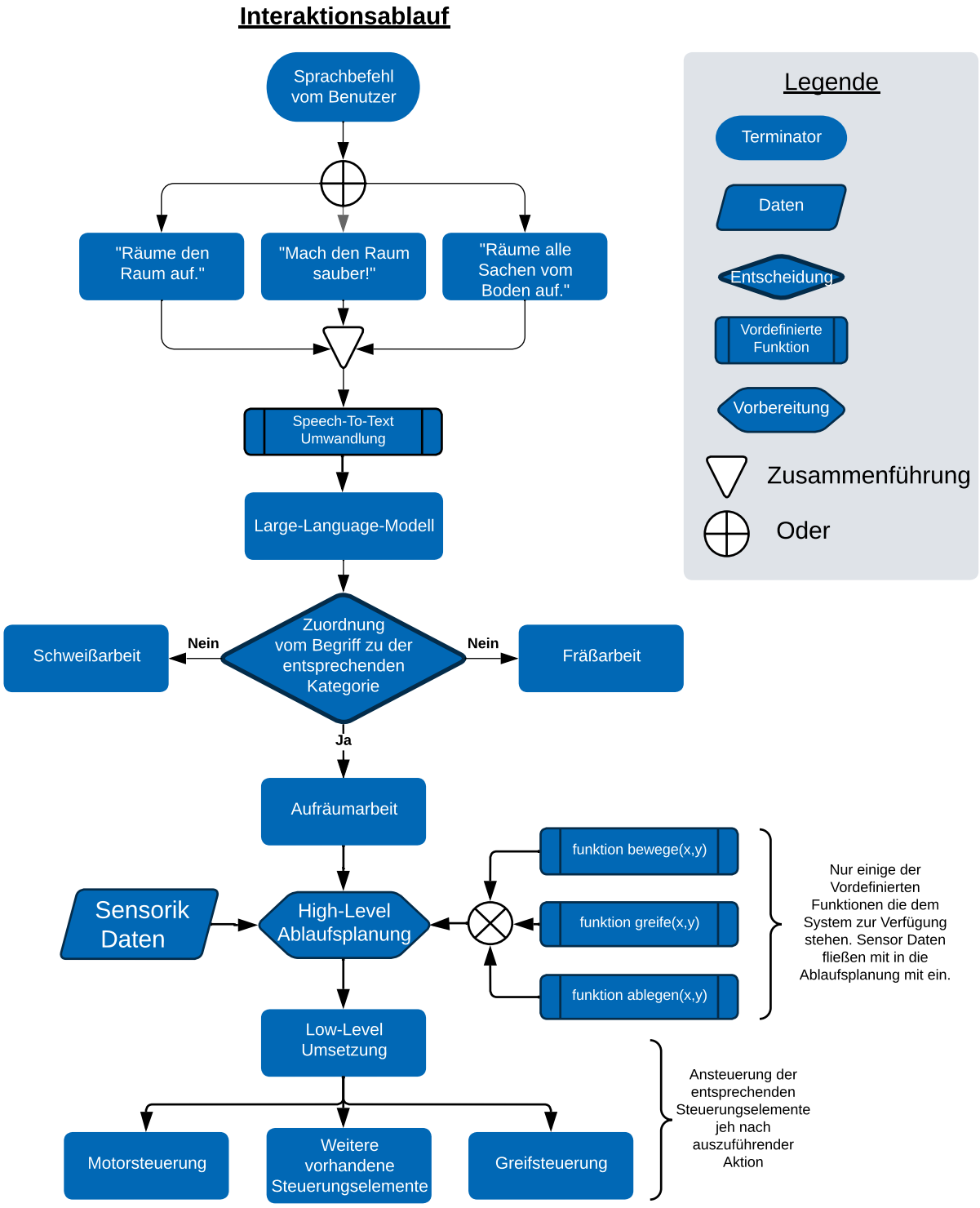

Abbildung 4.1: Interaktionsablauf ähnlicher, aber unterschiedlicher Sprachbefehle des Benutzers hin über das Large-Language-Modell über die High-Leve-Planung bis zur Low-Level-Steuerung.

Die nicht-funktionalen Anforderungen an das *Large-Language-Modell* sind:

1. Sensorische Genauigkeit und Qualität
2. Effizienz hinsichtlich der Verarbeitung
3. Datenqualität und Vorbereitung
4. Reglementierungen und Einschränkungen

Die sensorische Genauigkeit stellt eine Anforderung für das Zusammenspiel von Robotiksteuerung und *Large-Language-Modellen* dar. Wie bereits in den Anforderungen an die Robotiksteuerung hervorgehoben, bilden präzise und umfangreiche Sensordaten die Grundlage für korrekte Entscheidungen. Insbesondere im Bereich der *kollaborativen Robotik*, bei der eine direkte Interaktion mit Mensch und Roboter stattfindet, bilden ungenaue, fehlerhafte oder verzögerte Sensordaten eine Gefahr für potentiell gefährliche Situationen. Wenn beispielsweise die Kamera nur eingeschränkte Bereiche abdecken kann, fehlen dem Modell Informationen, was zu Fehlinterpretationen führen kann. [HR19, S. 2, 6] Das Modell soll auch in der Lage sein, Entscheidungen schnell und effizient treffen zu können. Dazu zählt auch die Bereitstellung der Sensordaten in Echtzeit, sodass alle Daten auch ohne Verzögerungen bereitgestellt und verarbeitet werden können. Diese Anforderung ist eng mit der Rechenkomplexität der Robotiksteuerung verknüpft, da aufwendige Berechnungen die Echtzeitlauffähigkeit beeinträchtigen könnten. Die Rechenkomplexität soll so gering wie möglich gehalten werden, um präzise Entscheidungen zur Kollisionsvermeidung oder Abstandsüberwachung treffen zu können. Die für das Training und *Fine-Tuning* verwendeten Daten müssen von hoher Qualität sein. Das bedeutet, dass die Daten spezifisch für die Aufgaben der Robotik passend und möglichst frei von *Bias* sind.

4.5 Anforderungen

Abschließend fasst Abbildung 4.2 die funktionalen und nicht-funktionalen Anforderungen an das Large-Language-Modell übersichtlich zusammen.

Anforderungen	
Nicht Funktional	Funktional
• Sensorische Genauigkeit und Qualität • Effizienz hinsichtlich der Verarbeitung • Datenqualität und Vorbereitung • Reglementierungen und Einschränkungen	• Unterstützung für Multimodalität zur Verwertung von Bild- und Toninformationen • Kontextverständnis und Reduktion von Mehrdeutigkeit (Bias) • Befehlsinterpretation zur Handlungsplanung • Planung auf High-Level- und Low-Level-Ebene

Abbildung 4.2: Gegenüberstellung der funktionalen und der nicht-funktionalen Anforderungen an die Robotiksteuerung.

Kapitel 5

Abschluss

5.1 Zusammenfassung

Die Seminararbeit analysiert umfassend die Anforderungen, die bei der Implementierung von Large-Language-Modellen in sprach gesteuerten Robotiksystemen berücksichtigt werden müssen. Im Fokus steht eine detaillierte Anforderungsanalyse, die sowohl technische Einschränkungen als auch sicherheitskritische Aspekte umfasst. Dabei wird untersucht, welche Voraussetzungen und Eigenschaften ein Large-Language-Modell im Zusammenspiel mit der Robotiksteuerung erfüllen muss, um erfolgreich als unterstützende Komponente in der Mensch-Roboter-Kollaboration zu agieren. Im Zuge der Analyse wird deutlich, dass ein Large-Language-Modell durch seine Fähigkeit zur dynamischen Interpretation von Sprachbefehlen und seinem kontextuelles Verständnis einen Mehrwert für robotergestützte Systeme bieten kann. Insbesondere die Unterstützung multimodaler Datenverarbeitung und die Möglichkeit, als zentrale Komponente sowohl auf High-Level- als auch Low-Level-Ebene zu agieren, sind entscheidende Faktoren, die die Integration von Large-Language-Modellen in Robotiksysteme besonders vorteilhaft machen und deren Nutzen hervorheben. Dennoch zeigt die Arbeit, dass die erfolgreiche Integration stark von der Anpassung an spezifische Anwendungsfälle abhängt und eine Vielzahl funktionaler und nicht-funktionaler Anforderungen berücksichtigt werden muss. Abschließend hebt die Arbeit hervor, dass die Nutzung von LLMs in der Robotik ein großes Potenzial mit sich bringt, doch erfordert es eine sorgfältige Planung und Umsetzung, um die technischen, ethischen, regulatorischen und sicherheitsspezifischen Anforderungen zu erfüllen.

5.2 Ausblick

Die Ergebnisse dieser Seminararbeit verdeutlichen, dass insbesondere die Forschung im Bereich der sicherheitsbewerteten Sensorik noch deutliche Lücken aufweist. Sicherheitsbewertete Sensorik umfasst jene Technologien, die innerhalb eines Robotiksystems genutzt werden können, um präzise Messwerte zu erfassen und auf dieser Basis sicherheitskritische Entscheidungen, wie beispielsweise Not-Stopp-Situationen, zuverlässig umzusetzen. Diese Erkenntnis stützt sich unter anderem auf die Arbeiten von Hofbaur, der die begrenzte Verfügbarkeit sicherheitsbewerteter Sensoren und deren Implikationen für die kollaborative Robotik

herausstellt [HR19, S. 1–3].

Darüber hinaus kann davon ausgegangen werden, dass in Zukunft neue Methoden und technologische Fortschritte im Bereich der Large-Language-Modelle entstehen, die dazu beitragen können, die in dieser Arbeit behandelten Herausforderungen effizienter zu lösen. Dies könnte die Entwicklung spezifischer "Off-The-Shelf"-Modelle für den Einsatz in der Robotik umfassen, ebenso wie eine Reduktion des Aufwands für die Datenvorbereitung beim Training oder Fine-Tuning solcher Modelle.

Eine weitere wichtige Perspektive ergibt sich durch die direkte Anwendung und Umsetzung der in dieser Arbeit diskutierten Anforderungen in realen Robotiksystemen. Konkrete Implementierungen und umfangreiche Tests könnten nicht nur zusätzliche Erkenntnisse wie der Sensorik liefern, sondern auch spezifische Anpassungen ermöglichen, die über die in dieser Arbeit aufgezeigten allgemeinen Anforderungen hinausgehen. Solche Anwendungen würden eine wichtige Grundlage für die Weiterentwicklung neuer Methoden und Vorgehensweisen darstellen, die gezielt auf individuelle Robotikmodelle abgestimmt werden können.

Literatur

[Aik] KIToolsFürAlle - Aiki. *BERT - Wie künstliche Intelligenz unser Sprachverständnis revolutioniert*. Zugriffsdatum: 2024-11-12. URL: https://ki-tools-fuer-alle.de/bert-wie-kuenstliche-intelligenz-unser-sprachverstaendnis-revolutioniert/.

[Bro+20] Tom B. Brown u. a. *Language Models are Few-Shot Learners*. [v4] Wed, 22 Jul 2020. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[Cha] Dev.to - Victor Chaba. *Understanding the Differences: Fine-Tuning vs. Transfer Learning*. Zugriffsdatum: 2024-12-11. URL: https://dev.to/luxacademy/understanding-the-differences-fine-tuning-vs-transfer-learning-370.

[Dev+19] Jacob Devlin u. a. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [v2] Fri, 24 May 2019. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[DINa] DIN1. *Robots and robotic devices - Collaborative robots*. Standard DIN TS 15066 (April 2017), Zugriffsdatum: 2024-12-19. URL: https://www.din.de/de/mitwirken/normenausschuesse/nam/veroeffentlichungen/wdc-beuth:din21:263754912.

[DINb] DIN2. *Robotics - Safety requirements - Part 1: Industrial robots*. Standard DIN EN ISO 10218-1 (2021), Zugriffsdatum: 2024-12-19. URL: https://www.din.de/de/mitwirken/normenausschuesse/nam/veroeffentlichungen/wdc-beuth:din21:136373717.

[Elaa] Elastic. *Vector Embedding*. Zugriffsdatum: 2024-11-11. URL: https://www.elastic.co/de/what-is/vector-embedding.

[Elab] Elastic. *Word Embedding*. Zugriffsdatum: 2024-11-11. URL: https://www.elastic.co/de/what-is/word-embedding.

[Fan+21] Chih-Hsien Fang u. a. „Multi-head Attention with Hint Mechanisms for Joint Extraction of Entity and Relation“. In: *Database Systems for Advanced Applications. DASFAA 2021 International Workshops*. Springer International Publishing, 2021, S. 321–335. DOI: 10.1007/978-3-030-73216-5. URL: https://doi.org/10.1007/978-3-030-73216-5_22.

[Fir+23] Roya Firoozi u. a. *Foundation Models in Robotics: Applications, Challenges, and the Future*. [v1] Wed, 13 Dec 2023. 2023. arXiv: 2312.07843 [cs.RO]. URL: https://arxiv.org/abs/2312.07843.

[Gal+19] Isabel O. Gallegos u. a. *Bias and Fairness in Large Language Models: A Survey*. [v3] Fri, 12 Jul 2024. 2019. arXiv: 2309.00770 [cs.CL]. URL: https://arxiv.org/abs/2309.00770.

[GG20] Benyamin Ghojogh und Ali Ghodsi. „Attention Mechanism, Transformers, BERT, and GPT: Tutorial and Survey“. In: (2020). DOI: 10.31219/osf.io/m6gcn. URL: https://doi.org/10.31219/osf.io/mru2x.

[Goo] Google. *Aplabet Investor Relations*. Zugriffsdatum: 2024-11-19. URL: https://abc.xyz/.

[HR19] Michael Hofbaur und Michael Rathmair. „Physische Sicherheit in der Mensch-Roboter Kollaboration“. In: *e & i Elektrotechnik und Informationstechnik* 136.7 (2019), S. 301–306. ISSN: 1613-7620. DOI: 10.1007/s00502-019-00743-2. URL: https://doi.org/10.1007/s00502-019-00743-2.

[HSO20] Tim-Lukas Habich, Moritz Schappler und Tobias Ortmaier. *Grundlagen der Robotik*. Springer Berlin Heidelberg, 2020. ISBN: 978-3-662-62424-1. DOI: 10.1007/978-3-662-62424-1_59-3. URL: https://doi.org/10.1007/978-3-662-62424-1_59-3.

[IAB] IAB-Forum. *Wie sich der demografische Wandel auf den deutschen Arbeitsmarkt auswirkt*. Zugriffsdatum: 2025-01-05. URL: https://www.iab-forum.de/wie-sich-der-demografische-wandel-auf-den-deutschen-arbeitsmarkt-auswirkt/.

[IBMa] IBM. *Neural Networks*. Zugriffsdatum: 2024-11-08. URL: https://www.ibm.com/de-de/topics/neural-networks.

[IBMb] IBM. *Self Supervised Learning*. Zugriffsdatum: 2024-12-06. URL: https://www.ibm.com/topics/self-supervised-learning.

[IBMc] IBM. *Zero-Shot-Learning*. Zugriffsdatum: 2024-11-07. URL: https://www.ibm.com/de-de/topics/zero-shot-learning.

[Int] Intersystems. *Was sind Vektor Embeddings?* Zugriffsdatum: 2024-11-27. URL: https://www.intersystems.com/de/resources/was-sind-vector-embeddings-alles-was-sie-wissen-mussen/.

[Ion] Ionos. *Vektordatenbanken - Was ist das?* Zugriffsdatum: 2024-11-27. URL: https://www.ionos.de/digitalguide/server/knowhow/vektordatenbank/.

[ITP] ITP. *Neuronale Netze*. Zugriffsdatum: 2024-11-11. URL: https://www.it-p.de/lexikon/neuronale-netze/.

[JMH24] Hyejun Jeong, Shiqing Ma und Amir Houmansadr. *Bias Similarity Across Large Language Models*. [v1] Tue, 15 Oct 2024. 2024. arXiv: 2410.12010 [cs.LG]. URL: https://arxiv.org/abs/2410.12010.

[LBH15] Yann Lecun, Yoshua Bengio und Geoffrey Hinton. „Deep learning“. In: *Nature* 521.7553 (2015), S. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://doi.org/10.1038/nature14539.

[MMB23] Ali Ahmad Malik, Tariq Masood und Alexander Brem. *Intelligent humanoids in manufacturing to address worker shortage and skill gaps: Case of Tesla Optimus*. [v1] Tue, 11 Apr 2023. 2023. arXiv: 2304.04949 [cs.RO]. URL: https://arxiv.org/abs/2304.04949.

[Mun] TowardsDataScience - Eram Munawwar. *Comprehensive Guide to SubWord-Tokenisers*. Zugriffsdatum: 2024-11-11. URL: https://towardsdatascience.com/a-comprehensive-guide-to-subword-tokenisers-4bbd3bad9a7c.

[Opea] OpenAI. *Better Language Models*. Zugriffsdatum: 2024-11-12. URL: https://openai.com/index/better-language-models/.

[Opeb] OpenAI. *OpenAI - Pioneering research on the path to AGI*. Zugriffsdatum: 2024-11-19. URL: https://openai.com/research/.

[Opec] OpenAI. *Whisper - OpenAI Speech to Text*. Zugriffsdatum: 2024-11-19. URL: https://platform.openai.com/docs/guides/speech-to-text.

[Pad+24] Akhil Padmanabha u. a. „VoicePilot: Harnessing LLMs as Speech Interfaces for Physically Assistive Robots“. In: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. [v2] Wed, 17 Jul 2024. ACM, 2024. DOI: 10.1145/3654777.3676401. URL: http://dx.doi.org/10.1145/3654777.3676401.

[Pata] AlexanderThamm - Patrick. *Foundation Models - Eine Einführung*. Zugriffsdatum: 2024-11-07. URL: https://www.alexanderthamm.com/de/blog/foundation-models-eine-einfuehrung/.

[Patb] AlexanderThamm - Patrick. *Unsupervised Learning kompakt erklärt*. Zugriffsdatum: 2024-11-12. URL: https://www.alexanderthamm.com/de/blog/unsupervised-learning-kompakt-erklaert/.

[Pav] TheAtlantic - John Pavlus. *Does AI understand language*. Zugriffsdatum: 2024-11-11. URL: https://www.theatlantic.com/technology/archive/2024/09/does-ai-understand-language/680056/.

[PD19] Andreas Pott und Thomas Dietz. *Steuerungstechnik - Industrielle Robotersysteme: Entscheiderwissen für die Planung und Umsetzung wirtschaftlicher Roboterlösungen*. 1. Aufl. Published as part of the eBook Packages Computer Science and Engineering (German Language). Springer Vieweg Wiesbaden, 2019. DOI: 10.1007/978-3-658-25345-5. URL: https://doi.org/10.1007/978-3-658-25345-5.

[Sap] Sapien. *What is Supervised Fine-Tuning? - Overview and Techniques*. Zugriffsdatum: 2024-12-06. URL: https://www.sapien.io/blog/what-is-supervised-fine-tuning-overview-and-techniques#:~:text=Supervised%20Fine%2DTuning%20(SFT)%20involves%20training%20models%20with%20labeled,resource%20constraints%20are%20a%20priority..

[Sch] TheDecoder - Maximilian Schreiner. *GPT-3: Die derzeit mächtigste Sprach-KI*. Zugriffsdatum: 2024-11-07. URL: https://www.the-decoder.de/openai-gpt-3-das-ist-die-derzeit-maechtigste-sprach-ki/.

[Ses] Acquisa - Thomas Sesli. *Neuronale Netze - einfach erklärt*. Zugriffsdatum: 2024-11-11. URL: https://www.acquisa.de/magazin/neuronale-netze.

[Sha] Shaip. *Guide on AI Training Data*. Zugriffsdatum: 2024-12-11. URL: https://de.shaip.com/blog/the-only-guide-on-ai-training-data-you-will-need-in/.

[Sic+08] Bruno Siciliano u. a. *Robotics: Modelling, Planning and Control*. 1. Aufl. Advanced Textbooks in Control and Signal Processing. Published as part of the eBook Packages Engineering, Engineering (R0). Springer London, 2008. DOI: 10.1007/978-1-84628-642-1. URL: https://doi.org/10.1007/978-1-84628-642-1.

[Ski] Skimai. *Multimodale KI: Anwendungsfälle*. Zugriffsdatum: 2024-11-07. URL: https://www.skimai.com/de/was-ist-multimodale-ki-anwendungsfalle-fur-multimodale-ki/.

[Tri] Medium - Sweety Tripathy. *Comparing GPT Tokenizers*. Zugriffsdatum: 2024-11-11. URL: https://medium.com/@sweety.tripathi13/comparing-gpt-tokenizers-968b60f5a72b#:~:text=GPT%2D3%20uses%20a%20byte,to%20the%20GPT%2D2%20tokenizer.

[Vas+17] Ashish Vaswani u. a. *Attention Is All You Need*. [v7] Wed, 2 Aug 2023. 2017. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[Vyk] Medium - Tomas Vykruta. *Understanding Causal LLMs, Masked LLMs, and Seq2Seq: A Guide to Language Model Training*. Zugriffsdatum: 2024-11-12. URL: https://medium.com/@tom_21755/understanding-causal-llms-masked-llm-s-and-seq2seq-a-guide-to-language-model-training-d4457bbd07fa.

[Wan+24] Xinyi Wang u. a. *Generalization v.s. Memorization: Tracing Language Models' Capabilities Back to Pretraining Data*. [v4] Wed, 27 Nov 2024. 2024. arXiv: 2407.14985 [cs.CL]. URL: https://arxiv.org/abs/2407.14985.

[Wika] Wikipedia. *Direkte Kinematik*. Zugriffsdatum: 2024-12-13. URL: https://de.wikipedia.org/wiki/Direkte_Kinematik.

[Wikb] Wikipedia. *Inverse Kinematik*. Zugriffsdatum: 2024-12-13. URL: https://de.wikipedia.org/wiki/Inverse_Kinematik.

[Wikc] Wikipedia. *Kollaborativer Roboter*. Zugriffsdatum: 2024-12-02. URL: https://de.wikipedia.org/wiki/Kollaborativer_Roboter.

[Wol] Xpert - Konrad Wolfenstein. *Bidirectional Encoder Representations from Transformers (BERT)*. Zugriffsdatum: 2024-11-12. URL: https://xpert.digital/bidirectional-encoder-representations-from-transformers/.

[Wu+23] Jiayang Wu u. a. *Multimodal Large Language Models: A Survey*. [v1] Wed, 22 Nov 2023. 2023. arXiv: 2311.13165 [cs.AI]. URL: https://arxiv.org/abs/2311.13165.

[Yan24] Jinbiao Yang. *Rethinking Tokenization: Crafting Better Tokenizers for Large Language Models*. [v1] Fri, 1 Mar 2024. 2024. arXiv: 2403.00417 [cs.CL]. URL: https://arxiv.org/abs/2403.00417.

[Yeo] Medium - Atharv Yeolekar. *Wordpiece Tokenization BPE Variant*. Zugriffsdatum: 2024-11-11. URL: https://medium.com/@atharv6f_47401/wordpiece-tokenization-a-bpe-variant-73cc48865cbf.

[Zan] Statista - Florian Zandt. *Geschätzte Trainingskosten für ausgewählte KI-Modelle nach Trainingsmethodik in Millionen US-Dollar*. Zugriffsdatum: 2024-12-11. URL: https://de.statista.com/infografik/33540/geschaetzte-trainingskosten-fuer-ausgewaehlte-ki-modelle-nach-trainingsmethodik--in-millionen-us-dollar-/.

[Zen+23] Fanlong Zeng u. a. *Large Language Models for Robotics: A Survey*. [v1] Mon, 13 Nov 2023. 2023. arXiv: 2311.07226 [cs.RO]. URL: https://arxiv.org/abs/2311.07226.

Anmerkung

Die Seitenangabe der jeweiligen Quellen wurde immer im Bezug auf die jeweilige PDF-Seitenangabe angegeben.