

EINSATZ VON LLM TOOLS IM SOFTWARE ENGINEERING PROZESS

BACHELORARBEIT
ZUR ERLANGUNG DES AKADEMISCHEN GRADES
BACHELOR OF ENGINEERING (B.ENG.)

Jonas Gundlach

Betreuerin:
Prof. Dr. rer. nat. Andrea Baumann

Tag der Abgabe: XX.XX.XXXX

eingereicht bei
Universität der Bundeswehr München
Fakultät für Elektrotechnik und Technische Informatik



Universität der Bundeswehr München

Institut für
Software Engineering

Neubiberg, Juli 2024

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen.

Der Speicherung meiner Bachelorarbeit zum Zweck der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

Neubiberg, den XX.XX.XXXX

Jonas Gundlach

Zusammenfassung

TODO!!!!!!!!!!!!!!

Ein Abstract ist ein Text, der kurz und knapp die wichtigsten Punkte deiner Bachelorarbeit wiedergibt. Er soll die Fragestellung, Methoden, Quellen und Ergebnisse zusammenfassen, damit die Lesenden schnell die Relevanz der Bachelorarbeit für ihr Interessensgebiet beurteilen können.

Inhaltsverzeichnis

Erklärung	III
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung der Arbeit	1
1.3 Struktur der Arbeit	1
2 Grundlagen	3
2.1 LLM Tools	3
2.1.1 ChatGPT	4
2.1.2 Google Gemini	4
2.1.3 Le Chat	4
2.2 Software Engineering Prozess	5
2.2.1 einzelne Phasen	5
3 Anwendung von LLM Tools im Software Engineering	7
3.1 Einzelne Phasen	7
4 Praxisergebnisse und Vergleich	9
4.1 ??????	9
4.2 ChatGPT	9
4.3 Gemini	9
4.4 Le Chat	9
4.5 Vergleich	9
5 Herausforderungen und Limitationen	11
5.1 Technische Herausforderungen	11
5.2 Limitationen von LLM Tools im Software Engineering Kontext	11
6 Zusammenfassung und Ausblick	13
6.1 Zusammenfassung der Ergebnisse	13
6.2 Implikationen für die Praxis	13
6.3 Ausblick und zukünftige Entwicklungen	13
Abbildungsverzeichnis	V
Tabellenverzeichnis	VII

Quellcodeverzeichnis	IX
Stichwortverzeichnis	XI
Literaturverzeichnis	XIII

1 Einleitung

TODO!!!!!!!!!!

1.1 Motivation

In einer Welt, die von immer komplexeren und dynamischeren Softwareanwendungen geprägt ist, wird die effektive Nutzung von Technologien zunehmend entscheidend für den Erfolg von Softwareprojekten. In diesem Zusammenhang gewinnen Werkzeuge des Natural Language Processing eine immer größere Bedeutung. Insbesondere die jüngsten Fortschritte im Bereich der Large Language Models (LLM) eröffnen neue Möglichkeiten für die Verbesserung des Software Engineering Prozesses.

Die Motivation hinter dieser Arbeit liegt in der Erkundung des Einsatzes von LLM-Tools im Software Engineering und der Analyse ihrer Auswirkungen auf verschiedene Phasen des Entwicklungszyklus. Die Frage, wie diese fortschrittlichen LLM-Tools genutzt werden können, um den Softwareentwicklungsprozess effizienter, präziser und insgesamt erfolgreicher zu gestalten, steht im Mittelpunkt dieser Untersuchung.

TODO!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

1.2 Zielsetzung der Arbeit

Todo!!!!!!!!!!!!!!!!!!!!!!

1.3 Struktur der Arbeit

Todo!!!!!!!!!!!!!!!!!!!!!!

2 Grundlagen

TODO!!!!!!!!!!

2.1 LLM Tools

Unter LLM-Tools versteht man Sprachmodelle, die auf einem Large Language Model (LLM) basieren. Ein LLM ist ein Deep-Learning-Algorithmus, der mit sehr großen Datensätzen trainiert wird. Diese Modelle finden häufig Anwendung im Bereich des Natural Language Processing (NLP), wo sie verwendet werden, um Abfragen in natürlicher Sprache zu beantworten oder Ergebnisse zu liefern. LLMs können neue Inhalte verstehen, zusammenfassen, generieren und vorhersagen. Durch das Training sammeln LLMs Milliarden von Parametern, bei denen es sich um Variablen handelt, die im Modell angepasst werden, um neue Inhalte abzuleiten.

LLMs basieren auf einem Transformer-Modell, das Eingaben in Token umwandelt und dann gleichzeitig mathematische Gleichungen ausführt, um Beziehungen zwischen den Token zu ermitteln. Dadurch kann der Computer Muster erkennen, die auch ein Mensch wahrnehmen würde, wenn ihm die gleiche Frage gestellt wird. Zudem verwenden Transformer-Modelle Selbstaufmerksamkeitsmechanismen, die es dem Modell ermöglichen, schneller zu lernen als herkömmliche Modelle. Dadurch kann das Transformer-Modell verschiedene Teile der Sequenz oder den gesamten Kontext eines Satzes berücksichtigen, um Vorhersagen zu generieren.

Grundsätzlich bestehen LLMs aus vier neuronalen Netzwerkschichten: der wiederkehrenden Ebene, der Einbettungsebene, der Feedforward-Ebene und der Aufmerksamkeitsebene. Diese Schichten arbeiten zusammen, um den Eingabetext zu verarbeiten und Ausgabeinhalte zu generieren.

Die wiederkehrende Ebene dient dazu, die Wörter des Eingabetextes der Reihe nach zu interpretieren und die Beziehungen zwischen den Wörtern in einem Satz zu erfassen. Die Einbettungsebene erfasst die semantische und syntaktische Bedeutung der Eingabe, sodass das Modell den Kontext verstehen kann. Die Feedforward-Schicht besteht aus mehreren vollständig verbundenen Schichten, die die Eingabeeinbettungen transformieren. Dadurch ermöglichen diese Schichten dem Modell, Abstraktionen auf höherer Ebene zu verstehen und somit die Absicht des Benutzers mit der Texteingabe zu erfassen. Die Aufmerksamkeitsebene ermöglicht es dem Sprachmodell, sich auf einzelne Teile des Eingabetextes zu konzentrieren, die für die Aufgabe relevant sind, und dadurch die genauesten Ausgaben zu generieren. Damit ein großes Sprachmodell Texteingaben empfangen und eine Ausgabevorhersage generieren kann, muss es zunächst allgemein geschult und anschließend feinabgestimmt werden, um spezifische Aufgaben ausführen zu können. Für die Schulung werden riesige Datenmengen im Petabyte-Bereich benötigt. Das Training verläuft mehrstufig und beginnt in der Regel mit einem unbeaufsichtigten Lernansatz, bei dem das Modell mit unstrukturierten und unbeschrifteten Daten trainiert wird, da diese in größeren

Mengen verfügbar sind. In dieser Phase leitet das Modell Beziehungen zwischen verschiedenen Wörtern und Konzepten ab.

Anschließend erfolgt die Schulung und Feinabstimmung durch eine Form des selbstüberwachten Lernens. Dabei wird eine Datenkennzeichnung durchgeführt, durch die das Modell verschiedene Konzepte besser und genauer identifizieren kann. Im nächsten Schritt durchläuft das LLM den transformatorischen neuronalen Netzwerkprozess im Rahmen des Deep Learning. Die Transformer-Modellarchitektur ermöglicht es dem LLM, mittels eines Selbstaufmerksamkeitsmechanismus Beziehungen und Verbindungen zwischen Wörtern und Konzepten zu erkennen, indem bestimmten Elementen (Token) Bewertungen zugewiesen werden, um die Beziehungen festzulegen.

Entscheidend über die Leistungsfähigkeit und die Qualität des Sprachmodells sind die Datensätze, welche zum trainieren benutzt werden. Im folgenden wird auf die drei unterschiedliche LLM-Tools eingegangen, welche in dieser Arbeit näher untersucht werden.

2.1.1 ChatGPT

ChatGPT wurde von dem Unternehmen OpenAI im Jahr 2022 veröffentlicht. Es gibt eine kostenfreie Version und eine Version, die ein kostenpflichtiges Abo benötigt. In der Arbeit wird mit der kostenfreien Version gearbeitet, um für die Mehrheit relevant zu sein. Die kostenfreie Version ist die GPT-3.5. Diese arbeitet mit Trainingsdaten, die bis Januar 2022 gehen. Man arbeitet lediglich mit Texteingaben und -ausgaben. Außerdem werden die Eingaben zu Trainingszwecken genutzt und gespeichert. Die kostenpflichtige GPT-4-Version arbeitet hingegen mit Daten bis 2023 und kann über Bing auf aktuelle Informationen zugreifen. Außerdem hat sie die Fähigkeit, auch Bilder zu verarbeiten und zu generieren. Daneben hat man auch die Möglichkeit, die Verarbeitung der Eingaben auszuschalten.

2.1.2 Google Gemini

Google Gemini (ehemals Bard) wurde von DeepMind entwickelt, was ein Tochterunternehmen von Google ist. Das Erstveröffentlichungsdatum war der 6. Dezember 2023 und basiert auf der Meena-Architektur von Google AI, die 2020 veröffentlicht wurde.

2.1.3 Le Chat

Le Chat ist ein, von dem französischen KI-Startup Mistral AI veröffentlichter Chatbot. Mistral AI startete mit seinen frei verwendbaren Sprachmodellen, also auf der Grundlage von Open Source, erfolgreich durch. Nun(????) hat Mistral AI sein bislang größtes Modell 'Mistral Large' veröffentlicht. Diesmal allerdings nicht auf der Basis von Open Source, sondern ausschließlich über die eigene Webseite und der KI-Infrastruktur Microsoft Azure. Es lassen sich allerdings API-Keys für Programmierschnittstellen erstellen, um z.B. Mistral Large über seinen eigenen Server laufen zu lassen und so für andere User auf der eigenen Homepage verfügbar zu machen. Mit Mistral Large wurde auch Le Chat veröffentlicht, welcher aktuell(?????????) kostenfrei verwendet werden kann. Le Chat bietet derzeit noch sehr wenige Funktionen an. Es stehen lediglich Texteingabe und -ausgabe zur Verfügung. Die Datengrundlage reicht nur bis 2021, weshalb es auch hier, für die Jahre 2022 bis heute, zu der Problematik der Halluzination

kommen kann. Grundsätzlich kann man zwischen drei Sprachmodellen auswählen: Large, Next und Small. Large bietet überlegene Denkfähigkeit, Next ist ein Prototyp-Modell für erhöhte Kürze und Small arbeitet schnell und kosteneffektiv. [1]

2.2 Software Engineering Prozess

Test Verweis auf Buch [2] Das ist ein Text.

2.2.1 einzelne Phasen

3 Anwendung von LLM Tools im Software Engineering

TODO!!!!!!!

3.1 Einzelne Phasen

4 Praxisergebnisse und Vergleich

TODO!!!!!!!

4.1 ??????

4.2 ChatGPT

4.3 Gemini

4.4 Le Chat

4.5 Vergleich

5 Herausforderungen und Limitationen

TODO!!!!!!!

5.1 Technische Herausforderungen

TODO!!!!!!!!!!!!

5.2 Limitationen von LLM Tools im Software Engineering Kontext

6 Zusammenfassung und Ausblick

TODO!!!!!!!!!!

6.1 Zusammenfassung der Ergebnisse

TODO!!!!!!!!!!!!!!

6.2 Implikationen für die Praxis

TODO!!!!!!!!!!!!!!

6.3 Ausblick und zukünftige Entwicklungen

TODO!!!!!!!!!!!!!!

Anhang

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Stichwortverzeichnis

????, 9

Anwendung von LLM Tools im Software Engineering, 7

Ausblick und zukünftige Entwicklungen, 13

ChatGPT, 4, 9

Einleitung, 1

Einzelne Phasen, 7

einzelne Phasen, 5

Gemini, 9

Google Gemini, 4

Grundlagen, 3

Herausforderungen und Limitationen, 11

Implikationen für die Praxis, 13

Le Chat, 4, 9

Limitationen von LLM Tools im Software Engineering Kontext, 11

LLM Tools, 3

Motivation, 1

Praxisergebnisse und Vergleich, 9

Struktur der Arbeit, 1

Technische Herausforderungen, 11

Vergleich, 9

Zielsetzung der Arbeit, 1

Zusammenfassung der Ergebnisse, 13

Zusammenfassung und Ausblick, 13

Literaturverzeichnis

- [1] Matthias Kindt.
Mistral AI: Französisches KI-Startup stellt Chatbot "Le Chat" und Kooperation mit Microsoft Azure vor.
Zugriff am 16. Mai 2024. 2024. URL: <https://www.unidigital.news/mistral-ai-franzoesisches-ki-startup-stellt-chatbot-le-chat-und-kooperation-mit-microsoft-azure-vor/> (siehe Seite 5).
- [2] Ian Sommerville. *Software Engineering*. 9. Auflage. Harlow, England: Addison-Wesley, 2010.
ISBN: 978-0-13-703515-1 (siehe Seite 5).

