Studiengang: Data Science & Management (M.Sc.)

Betreuer: Prof. Dr. Marcel Hebing

**Der Einfluss von Prompt Engineering auf Large Language Models im Argument Mining**

Masterarbeit

zur Erlangung des akademischen Grades eines

Masters of Science (M.Sc.)

Eingereicht von Benjamin Fels

Matrikelnummer: 190200

Datum: 28.01.2024

Executive Summary

Eine vollständige Zusammenfassung mit Fokus aus actionable insights

Inhaltsverzeichnis

[1 Einleitung 1](#_Toc185697447)

[1.1 Begriffsdefinition 3](#_Toc185697451)

[1.2 Stand der Forschung 4](#_Toc185697452)

[1.2.1 Argumente und Argument Mining 4](#_Toc185697453)

[1.2.2 Aufgaben des Argument minings 5](#_Toc185697454)

[1.2.3 Argument diagramming techniques / Argument Schemes 13](#_Toc185697455)

[1.2.4 Datensatz / Annotation von Argumenten 13](#_Toc185697456)

[1.2.5 Large Language Models 15](#_Toc185697457)

[2 Daten und Methoden 18](#_Toc185697458)

[2.1 Modelle 18](#_Toc185697459)

[2.2 Datensatzbeschreibung 19](#_Toc185697460)

[2.3 Methode 21](#_Toc185697461)

[3 Ergebnisse 25](#_Toc185697476)

[4 Diskussion und Handlungsempfehlungen 25](#_Toc185697477)

[5 Fragestellung 27](#_Toc185697478)

[6 Literaturverzeichnis / Quellenverzeichnis 28](#_Toc185697479)

[7 Anhänge & Projektdokumentation 31](#_Toc185697480)

[7.1 Modellvergleich 31](#_Toc185697481)

# Einleitung

- Vorarbeiten (eigene und extern, v.a. Literaturrecherche)

- Kontext der Arbeit klar verständlich machen

- Entwicklung der Forschungsfrage(n)

- Einleitung nicht zu kurz machen, mach immerhin 25% der Arbeit aus

In einer Zeit, in der Informationen in großer Menge und hoher Geschwindigkeit verbreitet werden, ist es von wachsender Bedeutung, wesentliche Aussagen schnell und präzise zu erfassen. Künstliche Intelligenz kann den Menschen bei seinen täglichen Aufgaben unterstützen.

Argumente sind ein wichtiger Bestandteil in der menschlichen Kommunikation. Peldszus & Stede (2013, S. 1) bezeichnen Argumentationen sogar als einen der zentralen Aspekte der menschlichen Kommunikation. Dabei werden Standpunkte anhand von Beispielen bestärkt mit dem Ziel die andere Seite von dem eigenen Standpunkt zu überzeugen. Gute Argumente sind zudem die Grundlage für eine fundierte Entscheidungsfindung bei verschiedenen Standpunkten (Stab & Gurevych, 2014, S. 1501). Das verstehen der argumentativen Struktur macht es nachvollziehbar, warum Menschen eine gewisse Meinung zu einem Thema haben (Cabrio & Villata, 2018, S. 5428; Lawrence & Reed, 2020, S. 765)

Stab & Gurevych (2014, S. 1501) führen an, dass die automatisierte Erkennung von Argumenten in Texten dazu beitragen kann, die Plausibilität der Argumentationsführung zu prüfen. Der Bereich, welcher sich mit der automatisierten Identifikation von Argumenten in Texten und der Beziehungen der Argumentationskomponenten untereinander auseinandersetzt, nennt sich Argument Mining. Klassische Ansätze für Argument Mining setzen häufig auf umfangreiche manuelle Identifikation, regelbasierte Verfahren oder spezialisierte maschinelle Lernmodelle, die auf spezifische Datensätze trainiert werden (Lawrence & Reed, 2020; Stab & Gurevych, 2017b). Diese Ansätze sind oft zeit- und kostenintensiv und erfordern Fachexpertise. Große Sprachmodelle (engl. Large Language Models, kurz LLMs) ermöglichen hierfür neue Ansätze. Sie liefern dem Stand der Technik entsprechende Ergebnisse bei gängigen Natural Language Processing (NLP) Aufgaben (Ozdemir, 2024, S. 46; Patil & Gudivada, 2024, S. 1). Zu diesen NLP-Aufgaben gehören beispielsweise maschinelle Übersetzung, Beantwortung von Fragen und Textgenerierung (Kochmar, 2022). Sprachmodelle sind nach Patil & Gudivada (2024, S. 4) Modelle, welche die Abfolge von Buchstaben oder Wörtern in Texten vorhersagen. LLMs sind demnach eine spezielle Form, welche auf einem großen Textkorpus trainiert wurden. Sie besitzen eine wesentlich größere Anzahl an Parametern und können eine Vielzahl von sprachbasierten Aufgaben durchführen. Bekannte LLMs sind die Generative Pre-trained Transformer (GPT) von OpenAI.

Hier die Inhalte einfügenwarum das Trainieren eines eigenen Modells sehr aufwändig ist.

Obwohl die Architektur der LLMs aufgabenunabhängig konzipiert ist, wird für ein Fine-Tuning ein aufgabenspezifischer Datensatz in einem Umfang von mehreren tausend bis hunderttausend Fällen benötigt (Brown et al., 2020, S. 3). Dies ist nicht bei jedem Anwendungsfall möglich. Zudem müsste für jede spezifische Aufgabe ein eigenes Modell trainiert werden, wofür wiederum jeweils eigene ausreichend große Datensätze benötigt werden würden (Brown et al., 2020, S. 6; Patil & Gudivada, 2024, S. 18).

Leistung von LLMs, die für „jederman“ verfügbar sind bei komplexen Aufgaben.

„Prompt engineering is typically the best place to start\*\*. It is often the only method needed for use cases like summarization, translation, and code generation where a zero-shot approach can reach production levels of accuracy and consistency.“ (OpenAI, o. J.-b)

Die Ressourcen hierzu stehen nicht jedem zur Verfügung. Eine Alternative stellt das Prompting dar, welches sich auf die Methode bezieht, einem LLM bestimmte Eingabeaufforderungen zu übergeben, um eine gewünschte Reaktion hervorzurufen (Maharjan et al., 2024, S. 8).

In der geplanten Untersuchung soll die Wirksamkeit dieser Methoden an dem Anwendungsfall des Argument Minings untersucht werden. Der Einsatz vortrainierter Modelle ohne umfangreiche Anpassung könnte den Aufwand für Argument Mining erheblich reduzieren. Trotz der Popularität großer Sprachmodelle gibt es bislang wenig systematische Studien, die sich explizit auf deren Anwendung für Argument Mining und die Rolle von Eingabeaufforderungen fokussieren. Dort soll die geplante Untersuchung ansetzen.

Die Masterarbeit zielt darauf ab, die Potenziale und Herausforderungen großer Sprachmodelle im Kontext des Argument Minings zu erforschen. Der Fokus liegt dabei auf der Anwendung von Prompt Engineering, um die Generalisierungsfähigkeiten dieser Modelle gezielt zu steuern und deren Leistung ohne Fine-Tuning zu maximieren. Die zentrale **Forschungsfrage** lautet: Wie beeinflusst der Einsatz von Prompt Engineering Techniken die Leistung von Large Language Models bei der automatisierten Erkennung von Argumentationskomponenten und deren Strukturen?

Ansatz zum automatisieren des Prozesses, zum Überführen von unstrukturierten daten (Text) in strukturierte Argumente.

Zur Beantwortung der Forschungsfrage bedarf es (Aufbau der Untersuchung beschreiben) Begriffsdefinition, Stand der Forschung

## Begriffsdefinition

Definition von wesentlichen Begriffen aus der Forschungsfrage:

* Prompt Engineering & Prompt Engineering Techniken. Die Eingabeaufforderungen werden systematisch variiert und deren Leistung untersucht.
* Large Language Models
* Argument Mining / Argumentationskomponenten

## Stand der Forschung

### Argumente und Argument Mining

Nach Peldszus & Stede (2013) sowie Stab & Gurevych (2017b) besteht ein Argument aus mehreren Komponenten wie Behauptungen und Prämissen, welche eine bestimmte Struktur durch die Beziehungen zwischen ihnen aufweisen. Demnach wird unter einer Behauptung eine kontroverse Aussage verstanden, welche den zentralen Bestandteil eines Arguments darstellt. Prämissen sind hingegen Gründe für die Rechtfertigung oder Widerlegung solch einer Behauptung. Unter Argument Mining kann hingegen im Hinblick auf diverse Definitionen (Cabrio & Villata, 2018, S. 5427; Lawrence & Reed, 2020, S. 766; Peldszus & Stede, 2013, S. 2; Yeginbergen et al., 2024, S. 11688) die automatische Identifikation und Extraktion der Argumentationskomponenten und deren Beziehungen zueinander aus Texten verstanden werden. Argument Mining stammt aus dem Bereich des NLP (Yeginbergen et al., 2024, S. 11687), welcher wiederum ein Teil aus dem Bereich der künstlichen Intelligenz ist (Kochmar, 2022; Lu et al., 2024, S. 2). Hier ggf. künstliche Intelligenz definieren

### Aufgaben des Argument minings

Argument Mining lässt sich wiederum in Teilaufgaben zerlegen. Auch hier gibt es in der Literatur abweichende Ansichten, wie diese Teilaufgaben zu unterteilen sind (Cabrio & Villata, 2018, S. 5428; Lawrence & Reed, 2020, S. 787–799; Peldszus & Stede, 2013, S. 20; Stab & Gurevych, 2017b, S. 620–621; Yeginbergen et al., 2024, S. 11687). Hier tiefer ins Detail gehen wer welche/wieviele Aufgabe sieht

Mal werden zwei (Cabrio & Villata, 2018, S. 5428), mal drei (Stab & Gurevych, 2014, 2017b) Teilaufgaben benannt.

Inhaltlich sind sich die Teilaufgaben sehr ähnlich und werden je nach Vorgehensweise zusammengefasst.

Für diese Untersuchung wird die folgende dreiteilige Gliederung der Teilaufgaben herangezogen. Zunächst wird der argumentative Text von dem nicht-argumentativen Text getrennt, gefolgt von der Unterteilung der Argumentationskomponenten in Behauptungen und Prämissen. Abschließend werden die argumentativen Beziehungen identifiziert.

Da die Teilaufgaben aufeinander aufbauen, wirken sich Fehler am Anfang negativ auf die nachfolgenden Aufgaben aus (Stab & Gurevych, 2017b, S. 648–649). Anstatt jedoch Modelle für jede einzelne Teilaufgabe zu trainieren wie es beispielsweise in Stab & Gurevych (2017b) erfolgt ist, soll sich die Lernfähigkeit von LLMs zu Nutze gemacht werden.

Die Erkennung von Argumenten in Textdokumenten umfasst daher mehrere Teilaufgaben: (1) die Trennung von argumentativen und nicht-argumentativen Texteinheiten, (2) die Identifizierung von Behauptungen und Prämissen und (3) die Identifizierung von Beziehungen zwischen Argumentkomponenten. (Stab & Gurevych, 2014, S. 1501)

Anhand der argumentativen Beziehungen können Argumentationsgraphen konstruiert werden. Es gibt verschiedene Theorien zu den Strukturen von Argumenten mit verschiedener Komplexität (Peldszus & Stede, 2013, S. 3–14)

. Peldszus & Stede, 2013 führen eine Reihe von Konstellationen der Beziehgungen der Argumentationskomponenten auf.

Neuere Argument Mining Ansätze betrachten die Extraktion der Argumente als eine Sequenzetikettierungsaufgabe (sequence labeling task) (Cheng et al., 2022, S. 2282; Stab & Gurevych, 2017b, S. 636; Yeginbergen et al., 2024, S. 11688). POS-Tagging und NER sind auch sequence labeling tasks

Anstatt Modelle für jede einzelne Teilaufgabe zu trainieren, soll sich die Lernfähigkeit von LLMs zunutze gemacht werden. LLMs benötigen jedoch in der Regel große Datensätze, um ein Modell mit ausreichender Performance zu trainieren. Nicht immer besteht jedoch die Möglichkeit, einen großen Datensatz mit annotierten Daten zusammenzustellen, da dies zeitaufwändig und kostspielig sein kann (Tunstall et al., 2023, S. 289), wie auch in dem Fall von Argument Mining (Lawrence & Reed, 2020, S. 780). Während des Vortrainings entwickelt ein Sprachmodell Mustererkennungsfähigkeiten. Brown et al. (2020, S. 3–6) zeigen, dass diese Fähigkeiten dann bei Eingabeaufforderungen genutzt werden können, um das Modell anhand von Beispielen an die gewünschte Aufgabe anzupassen. Dies wird als In-Context Learning (ICL) bezeichnet. Die Gewichtungen des Modells werden dabei nicht verändert. ICL kann in Few-Shot Learning, One-Shot Learning und Zero-Shot Learning unterschieden werden (Patil & Gudivada, 2024, S. 23–25). Die Unterscheidung richtet sich danach, wie viele Beispiele in der Eingabeaufforderung übergeben werden. Prompt Engineering baut darauf auf und beschäftigt sich mit der Gestaltung von Eingabeaufforderungen, welche genutzt werden, um die Fähigkeiten von LLMs für bestimmte Aufgaben zu nutzen (Trad & Chehab, 2024, S. 369). Maharjan et al. (2024) konnten für medizinische Benchmarks zeigen, dass mittels Prompt Engineering Techniken ohne Fine-Tuning dem Stand der Technik entsprechende Ergebnisse für Open-Source Modelle erzielt werden können. Dem gegenüber stehen Untersuchungen wie Trad und Chehab (2024) oder (Yeginbergen et al., 2024), welche zu dem Ergebnis kommen, dass Prompt Engineering eine schlechtere Leistung hervorbringt als Fine-Tuning. Nichtsdestotrotz bleibt es aufgrund seiner Einfachheit und Flexibilität eine attraktive Alternative, insbesondere bei Anwendungsfällen, in denen die hohen Anforderungen von LLMs an Datenmenge und Hardware die Anwendung von Fine-Tuning unmöglich machen. Deshalb soll das Potential von Prompt Engineering speziell für das Argument Mining untersucht werden, um zu evaluieren, ob es trotz der beschriebenen Einschränkungen eine praktikable und ressourcenschonende Lösung für diese Aufgabe bieten kann.

Die manuelle Identifizierung von Argumenten in einem Text ist sehr aufwändig. Bei der Annotation von Argument-Datensätzen werden Richtlinien erstellt, welche von den Annotoren bei der Beschriftung herangezogen werden. Trotz dieser Richtlinien kann es zu Abweichungen aufgrund persönlicher Interpretation kommen. Beispiel/Quelle hier anführen.

### Argument diagramming techniques / Argument Schemes

Darstellung der Argumente in einer Strukturierten Form. Kann erst erfolgen, nachdem die Komponenten einer Argumentation extrahiert wurden.

Nach (Peldszus & Stede, 2013, S. 6) können Argumente in einem Argument-Graphen abgebildet werden, mit dem Prämissen und Schlossfolgerungen als Knoten und der Beziehung zwischen Ihnen als Pfeil. Die einfachste Konstellation eines Arguments wäre demnach eine Schlussfolgerung, welche von einer Prämisse unterstützt wird.

* Walton, D., Reed, C., & Macagno, F. (2008). Argumentation schemes. Cambridge University Press. doi:10.1017/CBO9780511802034.
* Reed, C., Walton, D., & Macagno, F. (2007).Argument diagramming in logic, law and artificial intelligence. The Knowledge Engineering Review, 22(1), 87–109. doi:10.1017/S0269888907001051.

### Datensatz / Annotation von Argumenten

*„Die Zuverlässigkeit eines annotierten Korpus wird durch die Berechnung der Übereinstimmung zwischen den Annotatoren gewährleistet, die den Grad der Übereinstimmung bei der Durchführung der Annotationsaufgabe zwischen den beteiligten Annotatoren misst.* *Bei der Erstellung eines Datensatzes für die Vorhersage von Beziehungen ist das statistische Maß für die Berechnung der Übereinstimmung zwischen den von den Kommentatoren vergebenen Bezeichnungen der Cohen's Kappa-Koeffizient. Wenn die Bewerter vollständig übereinstimmen, dann = 1, wenn es keine Übereinstimmung zwischen den Bewertern gibt, die über das hinausgeht, was zufällig zu erwarten wäre, = 0. Bei NLP-Aufgaben wird die Übereinstimmung als signifikant angesehen, wenn sie >0,6 ist.“* (Cabrio & Villata, 2018, S. 5428)🡪 For more details about inter-annotator agreement, we refer the reader to [Artstein, 2017] 🡪 [Artstein, 2017] Ron Artstein. Inter-annotator Agreement, pages 297–313. Springer Netherlands, Dordrecht, 2017.

Einen Vergleich verfügbarer Datensätze bieten beispielsweise (Cabrio & Villata, 2018, S. 5432).

*„Darüber hinaus wurden seit Beginn der Forschung im Bereich AM verschiedene heterogene Datensätze erstellt. Aufgrund der Unausgereiftheit eines aufstrebenden Feldes und des Mangels an klaren Definitionen wurde jeder Datensatz auf der Grundlage leicht unterschiedlicher Definitionen der Argumentkomponenten und der zwischen ihnen bestehenden Beziehungen annotiert, was die Möglichkeit eines einfachen Abgleichs zwischen den Datensätzen verhindert.“* (Cabrio & Villata, 2018, S. 5432).

(Lawrence & Reed, 2020, S. 780)*: „Eine der Herausforderungen, mit denen sich die derzeitigen Ansätze zum Argument Mining konfrontiert sehen, ist der Mangel an großen Mengen an angemessen kommentierten Argumenten, die als Trainings- und Testdaten dienen können. In jüngster Zeit wurden mehrere Anstrengungen unternommen, um diese Situation durch die Erstellung von Korpora in einer Reihe von verschiedenen Bereichen zu verbessern.“*

*„Obwohl das Ziel des Argument Mining die Extraktion von Argumentationsstrukturen aus natürlichem Text ist, stellt die Verfügbarkeit großer Mengen entsprechend annotierter Trainingsdaten eine große Herausforderung bei der Durchführung dar.“* (Lawrence & Reed, 2020, S. 798)Als Alternative Ansatz Verwendung von Prompt Engineering, da wenig Daten verwendet werden.

Um dem Mangel an annotierten Daten entgegenzuwirken, beschäftigen sich neuere Arbeiten mit der Erstellung von Annotations-Richtlinien (Lawrence & Reed, 2020, S. 806). Diese Verwendung der spezifischen Annotations-Richtlinien bedeutet, dass diese sich auf den jeweiligen Bereich in dem sie entwickelt wurden, beschränken(Lawrence & Reed, 2020, S. 806). Einen einheitlichen Methodik gibt es derzeit nicht. (Lawrence & Reed, 2020, S. 786)

In (Cabrio & Villata, 2018, S. 5429) tragen die leistungsfähigsten Algorithmen, die am häufigsten verwendeten Merkmale, und die freigegebenen Datensätze zusammen. Demnach werden syntaktische und positionsbezogene Merkmale am häufigsten verwendet (Cabrio & Villata, 2018, S. 5431)

### Large Language Models

Abgrenzung Prompt Engineering, Fine-Tuning und RAG

Die Integration von domänenspezifischen Wissen in LLMs stellt eine Herausforderung dar. Es erscheint sinnvoll das Modell auf die eigenen Bedürfnisse/Aufgaben anzupassen. Hierzu gibt es verschiedene Ansätze wie fine-tuning/transfer learning, RAG oder Prompt Engineering. Das trainieren eines eigenen Modells benötigt viele Daten und ist kostenintensiv aufgrund der benötigten Hardware. Fine-Tuning ist günstiger, da ein vortrainiertes Modell verwendet werden kann. Benötigt aber dennoch einige Daten und Rechenkapazitäten. Für einen ersten Einstieg bleibt somit nur Prompt-Engineering.

Die grundlegenden LLMs eignen sich für allgemeine bzw. übergreifende Tätigkeiten, jedoch weniger für domänenspezifische Aufgaben, wie in den Untersuchungen von Lu et al. (2024) zu entnehmen. Traditionelle Modelle sind für spezifische Aufgaben trainiert (Single-task-learning). LLMs hingegen sind in der Lage mehrere Aufgabe zu übernehmen (multi-task-learning).

Argument Mining besteht aus mehreren Teilaufgaben, welche bisher meist (Beispiele aufführen) von einzelnen Modellen übernommen wurden. In dieser Untersuchung soll sich die Fähigkeiten von LLMs zu Nutze gemacht werden, mehrere Aufgaben zu bewältigen.

LLMs benötigen einen rieseigen Korpus für das Training. Prompt Engineering benötigt wenig bis gar keine Daten zum trainieren.

In (Maharjan et al., 2024) konnte für Medizinische Benchmarks gezeigt werden, dass mittels Prompt Engineering Techniken ohne fine-tuning SOTA-Ergebnisse für open source Modelle erzielt werden konnte. Dabei wurde deutlich, dass bei fine-tuned Modellen die Zero-Shot Performance besser ist.

*“* *Im Gegensatz zu früheren Arbeiten zeigen wir, dass beim Argument Mining der Datentransfer bessere Ergebnisse erzielt als der Modelltransfer und dass das Finetuning besser abschneidet als few-shot-Methoden. Bei ersterem scheint die Domäne des für den Datentransfer verwendeten Datensatzes ein entscheidender Faktor zu sein, während sich bei few-shot die Art der Aufgabe (Länge und Komplexität der Sequenzspannen) und die Stichprobenmethode als entscheidend erweisen.”* (Yeginbergen et al., 2024, S. 11687)

Enorme Hardwareanforderungen beim Trainieren eigener LLMs wie in Table 1 (Patil & Gudivada, 2024, S. 6) von zu sehen. Zur Einordnung, eine H100 Grafikkarte kostet …

**Transfer Learning**

# Daten und Methoden

- Vorgehen Datengenerierung und -aufbereitung

- Datenlage und -qualität

- Vorgehen und Methoden der Datenanalyse

## Modelle

Llama 3.2-3B instruct

Abgleich mit Llama 3.2-1B instruct und GPT?

Gpt-4o-mini

Um die Ausgaben des LLMs weniger zufällig und somit möglichst reproduzierbar werden zu lassen wurde die Temperatur auf Null festgelegt (OpenAI, o. J.-a).

## Datensatzbeschreibung

Es gibt verschiedene Datensätze, welche sich in ihrem Schwerpunkt und den Annotationen unterscheiden (Lawrence & Reed, 2020, S. 780–786). Für die Auswahl eines geeigneten Datensatzes wurden diverse Kriterien herangezogen. Zunächst sollte der Datensatz vorab nicht bereits von dem nicht-argumentativen Text befreit worden sein, um die Realität bestmöglich abzubilden (Stab & Gurevych, 2017b, S. 620). Desweiten wird der Ansatz verfolgt, nicht für jede Teilaufgabe des Data Minings einen eigenen Datensatz zu verwenden. Stattdessen soll sich der Datensatz über die drei Teilaufgaben hinweg verwenden lassen. Es bedarf folglich eines annotierten Datensatzes, in dem die Argumentationskomponenten und die argumentativen Beziehungen ausgewiesen werden. Ein bereits annotierter Datensatz ermöglicht es anhand der Grundwahrheit die Ausgaben des LLMs zu evaluieren. Sofern kein passender Datensatz verfügbar ist, bestünde die Möglichkeit einen eigenen Datensatz zu erstellen. Aufgrund der Komplexität bei der Datenbeschriftung wird sich jedoch dagegen entschieden. Zur Orientierung: Die Annotations-Richtlinien von Stab & Gurevych (2017b, S. 630) umfassen 31 Seiten. Entsprechend der zuvor genannten Anforderungen wird der Argument Annotated Essays (Version 2) Datensatz (AAEC) (Stab & Gurevych, 2017a) als geeignet betrachtet. Dieser Datensatz ist das Ergebnis der Arbeit von Stab & Gurevych (2017b) wie sie in dem Artikel „Parsing Argumentation Structures in Persuasive Essays“ beschrieben wird.

Neben dem geringeren zeitlichen Aufwand ist auch die eine höhere Qualität der Annotation von Vorteil.

Der Datensatz besteht aus 420 von der Webseite essayforum.com zufällig ausgewählten überzeugenden Aufsätzen (Stab & Gurevych, 2017b, S. 630). Solche überzeugenden Aufsätze eignen sich gut für das Argument Mining, da sie ein bestimmtes Thema erläutern wobei der Autor versucht die Leser von seinem Standpunkt zu überzeugen (Cabrio & Villata, 2018, S. 5429). Die Annotatoren nutzen zur Beschriftung der Aufsätze das Brat Rapid Annotation Tool. <https://aclanthology.org/E12-2021.pdf> // <http://brat.nlplab.org/>

Zum AAEC

* Annotated with discourse-level argumentation structures.
* Korpus und Annotation-Guidelines frei verfügbar.
* End-to-end argumentation structure parser, that identifies argument components at the token level
* Essays annotated by the expert annotator were used as training data (80%), essays annotated by the other annotators used as test data (20%)

Die Auswahl dieses Datensatzes wird von (Yeginbergen et al., 2024, S. 11688) als “perhaps the most popular NLP dataset manually annotated with argument structures” bezeichnet

Der Datensatz enthält sowohl die vollständigen Aufsätze als Text-Dateien, als auch die Annotationen als ann-Dateien. Bei den Annotationen werden die Argumentationskomponenten Hauptaussage (MajorClaims), Behauptungen (claims) und Prämissen (premises) unterschieden. Nach Stab & Gurevych (2017b, S. 627) beinhalten solche Aufsätze folgende argumentative Struktur. Eine Hauptaussage spiegelt den Standpunkt des Autoren wider, wobei diese anhand von Argumenten unterstützt oder angegriffen werden. Ein Argument besteht aus einer Behauptung und mindestens einer Prämisse. Prämissen sind dabei die Gründe für ein Argument. Um die Haltung der Argumente zu unterscheiden, sind die Behauptungen als für oder gegen markiert. Prämissen hingegen können eine Behauptung oder eine andere Prämisse unterstützen oder angreifen. Es ist möglich, dass es mehrere Hauptaussagen zu einem Text gibt. Hierbei wurde jedoch nicht annotiert, auf welche Hauptaussagen sich die Behauptungen beziehen.

Beschriftung einfügen: Argumentationskomponenten und deren Strukturen. Abbildung ggf. in Anhang packen

Ein Bild, das Text, Diagramm, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Eigene Darstellung nach Stab & Gurevych (2017b).

**Datenaufbereitung**

Die IDs geändert, da alle argumentativen Komponenten als TX mit X als fortlaufende Nummer bezeichnet sind. Es wird dabei somit nicht zwischen den Argumentationskomponenten unterschieden. Dieser Schritt wurde vorgenommen, um die Beziehungen besser nachvollziehen zu können, sodass die annotierten Daten dann in JSON überführt werden können. Es wurde sich für ein semi-strukturiertes Format entschieden, da sowohl die Daten, als auch die Ausgabe des LLMs in dieses Format überführt werden können und sie somit für die Evaluation miteinander verglichen werden. Zudem können die Ergebnisse dann für jede Argumentationskomponente individuell betrachtet werden.

## Methode

Im Rahmen der Abschlussarbeit wird ein experimenteller Ansatz verfolgt. Es soll explorativ untersucht werden, inwieweit die Eingabeaufforderungen zur Leistungssteigerung des LLMs für den Anwendungsfall Argument Mining beitragen können. Hierzu werden einem Modell systematisch verschiedene Eingabeaufforderungen übergeben, mit dem Ziel, die Auswirkung der Prompt Engineering Techniken zu untersuchen. Konkret bedeutet dies beispielsweise, dass die Anzahl der Beispiele in einer Eingabeaufforderung variiert wird. Damit dem Modell in der Eingabeaufforderung Beispiele übergeben werden können, werden argumentative Texte benötigt, in denen die Argumentationskomponenten und deren Beziehungen zueinander annotiert sind. Da der Schwerpunkt auf der Anwendung von Eingabeaufforderungen in LLMs für Argument Mining und nicht auf der sprachwissenschaftlichen Theorie zu Argumenten liegt, soll kein eigener Datensatz erstellt werden. Dieser Datensatz ist so aufzubereiten, dass daraus die Vorlagen für die Eingabeaufforderungen gemäß der verschiedenen Prompt Engineering Techniken erstellt und an das LLM übergeben werden können. Der Datensatz ist zusätzlich in einen Trainings- und einen Testdatensatz zu unterteilen. Die Ausgaben des LLMs sollen zur Evaluation der Leistung mit der Grundwahrheit abgeglichen werden. Hierzu sind geeignete Metriken heranzuziehen.

***Eingabeaufforderungen***

LLMs sind jedoch in der Lage Aufgaben zu erfüllen, für welche sie während des Trainings nicht explizit trainiert wurden. LLMs besitzen die Fähigkeit von den Beispielen aus dem Prompt zu lernen (Tunstall et al., 2023, S. 330).

Prompting bezieht sich auf die Methode, einem LLM bestimmte Eingaben zu machen, um eine gewünschte Reaktion hervorzurufen (Maharjan et al., 2024, S. 8). Vorteil gegenüber feintuning: Es werden keine Trainingsdaten benötigt. Nachteil gegenüber feintuning: Man kann keinen Nutzen aus den gelabelten Daten ziehen, selbst wenn man diese zur Verfügung stehen hat (Tunstall et al., 2023, S. 330). Stimmt nicht ganz, da man bei few-shot-learning auch aus gelabelten Daten einen Nutzen ziehen kann.

Eingabeaufforderungen in Englisch, da der Datensatz in Englisch ist die Sprachen nicht gemischt werden sollten.

**Zero-Shot Learning**

Kein Beispiel übergeben

**One-Shot Learning**

Ein Beispiel übergeben

**Few-shot Learning**

* Zufällige Auswahl der Beispiele
* Systematische Auswahl der Beispiele

Wie viel Beispiele braucht es, damit es als Few-Shot Learning zählt?

Nach Brown et al. (2020, S. 6) werden dabei in der Regel zwischen 10 und 100 Beispiele übergeben.

Große Texte mit vielen Tokens erfordern mehr Rechenressourcen und längere Verarbeitungszeiten, deshalb ist die Effiziente handhabe der Tokenanzahl ein wichtiger Aspekt bei der Entwicklung von generative AI-Modellen.

Das Problem ist, das die maximalen Tokenanzahl für die HuggingFace API bei 4096 liegt. Das bedeutet, dass in Summe die Eingabe Token und die Ausgabe Tokens diesen Wert nicht überschreiten dürfen. Eingabeaufforderungen oberhalb dieser Begrenzung landen in einer Fehlermeldung. Die maximale Ausgabetokens müssen sich somit auch so gewählt werden, sodass die Ausgaben nicht zu früh abgebrochen werden.

Dies hat ebenfalls zur Folge, dass in den Eingabeaufforderungen nicht unbegrenzt viele Beispiele übergeben werden können. Bei 100 ausgabetoken dürfen 3000 tokens eingegeben werden. Dies wird bereits bei 3 Beispielen überschritten

Diese Grenze ist bereits bei X Beispielen erreicht. Es ist somit notwendig eine andere Möglichkeit zu finden, um dem Daten die relevanten Informationen zu übergeben, ohne diese Grenze zu überschreiten. Dafür wird das Kontext-Fenster des Modells herangezogen, welches mit 128 Tsd. Token Quelle größer ist, als die Grenze bei Eingabeaufforderungen.

Das Modell besitzt jedoch ein größeres Kontext-Fenster Quelle. Ggf. kann das ausgenutzt werden, indem die Beispiele nacheinander übergeben werden.

ODER Teilaufgaben in aufeinanderfolgenden Prompts ausführen lassen. Der Output eines Prompts wird von dem darauffolgenden Prompt genutzt (refinement, prompt chaining, sequential few-shot prompting, iterative re-prompting).

Ansätze wie die automatische Begrenzung der Eingabeaufforderung (quelle) bei erreichen der Token-Grenze würden wesentliche Informationen beschneiden. Auch die Unterteilung der Essays in Teilabschnitte würde das gesamtheitliche Verständnis des Textes verzerren.

Language Models are Few-Shot-Learners (Brown et al., 2020)

(Brown et al., 2020, S. 1)zeigten für GPT-3, dass „scaling up language models greatly improves task-agnostik few-shot performance“, Dafür wurde GPT-3 ohne „gradient updates“ oder fine-tuning angewendet.

*However, a major limitation to this approach is that while the architecture is task-agnostic, there is still a need for task-specific datasets and task-specific fine-tuning: to achieve strong performance on a desired task typically requires fine-tuning on a dataset of thousands to hundreds of thousands of examples specific to that task. Removing this limitation would be desirable, for several reasons. First, from a practical perspective, the need for a large dataset of labeled examples for every new task limits the applicability of language models. There exists a very wide range of possible useful language tasks, encompassing anything from correcting grammar, to generating examples of an abstract concept, to critiquing a short story. For many of these tasks it is difficult to collect a large supervised training dataset, especially when the process must be repeated for every new task. Second, the potential to exploit spurious correlations in training data fundamentally grows with the expressiveness of the model and the narrowness of the training distribution. This can create problems for the pre-training plus fine-tuning paradigm, where models are designed to be large to absorb information during pre-training, but are then fine-tuned on very narrow task distributions.* *Third, humans do not require large supervised datasets to learn most language tasks – a brief directive in natural language (e.g. “please tell me if this sentence describes something happy or something sad”) or at most a tiny number of demonstrations (e.g. “here are two examples of people acting brave; please give a third example of bravery”) is often sufficient to enable a human to perform a new task to at least a reasonable degree of competence. Aside from pointing to a conceptual limitation in our current NLP techniques, this adaptability has practical advantages – it allows humans to seamlessly mix together or switch between many tasks and skills, for example performing addition during a lengthy dialogue. To be broadly useful, we would someday like our NLP systems to have this same fluidity and generality.* (Brown et al., 2020, S. 3–4)

*During unsupervised pre-training, a language model develops a broad set of skills and pattern recognition abilities. It then uses these abilities at inference time to rapidly adapt to or recognize the desired task. We use the term “in-context learning” to describe the inner loop of this process, which occurs within the forward-pass upon each sequence.* (Brown et al., 2020, S. 3)

**Chain-of-thought**

Wei, J., Wang, X., Schuurmans, D., et al . Chain-of-thought prompting elicits reasoning in large language models. in Neural Information Processing Systems. New Orleans, LA, 2022

**Self-consistency**

Wang, X., Wei, J., Schuurmans, D., et al . Self-consistency improves chain of thought reasoning in language models. Epub ahead of print 2023. <https://doi.org/10.48550/arXiv.2203.11171>.

* + where we run the prompt multiple times through the model and implement a majority voting strategy to get the answer

# Ergebnisse

- Möglichst objektive Darstellung der Ergebnisse

- Idealerweise 1-3 zentrale Visualisierungen der Kernaussagen. Pro-Tipp: Aussagestarke Grafik mit ausführlicher Tabelle im Anhang

Plot von Evaluationsmetrik in Abhängigkeit des Token Counts (=Kosten) pro Prompt Technik. Auch wenn die Kosten pro Token pro Modell abweichen, können sie als Kostentreiber verstanden werden.

Bspw. Kennzahl bilden: Genauigkeit pro Token?

# Diskussion und Handlungsempfehlungen

- Diskussion der Ergebnisse

- Möglicherweise Ableitung von Handlungsempfehlungen

- Limitationen, weiterführende (neue) Forschungsfragen, etc.

**Praktischer Mehrwert**

Die geplante Abschlussarbeit wird voraussichtlich Aufschluss darüber geben, wie leistungsfähig LLMs für das Argument Mining sind und welche Prompt Engineering Techniken in diesem Zusammenhang die besten Ergebnisse liefern. Sie würde sowohl zur Weiterentwicklung der Forschung auf dem Gebiet des Argument Minings als auch zur praktischen Anwendung von LLMs in realen Anwendungsfällen beitragen. Eine effiziente und strukturierte Extraktion von Argumenten kann dazu beitragen, Diskussionen nachvollziehbarer zu machen, sodass komplexe Themen besser durchdrungen und darauf aufbauend fundierte Entscheidungen getroffen werden können.

**Evaluation**

Evaluationsmetrik wählen

1. BERTScore: misst semantische Ähnlichkeit. Soll es auf huggingface geben.
2. BLEU (precision) / ROGUE (recall)
3. Klassische Klassifikationsmetriken: Accuracy, Precision, Recall, F1-Score,…

“Mochales Palau and Moens perform machine learning, using a variety of features on different levels of description. Their implementation proceeds in two steps: First, sentences are being classified as either ‘argumentative’ or ‘non-argumentative’. In their implementation with a multinomial naive Bayes classifier and a maximum-entropy model, their best average accuracy was almost 74%. […] The second step tries to further classify the ‘argumentative’ sentences into the categories of ‘premise’ and ‘conclusion’, and here they achieved F-measures of 68% and 74%, respectively.” (Peldszus & Stede, 2013, S. 25–26)

Mochales Palau, R., & Moens, M.-F. (2009). Argumentation mining: The detection, classification and structure of arguments in text. In Proceedings of the ICAIL 2009, Barcelona, Spain (pp. 98–109)

F1-score (Yeginbergen et al., 2024, S. 11691)

*„Es ist wichtig, hier zu betonen, dass auch die menschliche Zustimmung (die im Allgemeinen als Obergrenze für die automatische Leistung bei Annotationsaufgaben angesehen wird) von der Komplexität der AM-Aufgaben beeinflusst wird“* (Cabrio & Villata, 2018, S. 5431).

**Limitationen**

Die Ergebnisse gelten nur für die untersuchten LLMs

Nur für englische Sprache

**Handlungsempfehlungen**

* Untersuchung für weitere LLMs
* Untersuchung der Performance, wenn theoretische Wissen wie die Annotation Richtlinien mit RAG eingebaut werden.
* Anwendung von Fine-Tuning.
* Untersuchung für weitere Sprachen
* Weiterentwicklung indem Argumentationsstrukturen in Strukturdiagrammen aufgeführt werden, um die Extrahierten Teile visuell leicht verständlich aufzuarbeiten.

# Fragestellung

Wie können LLMs auf eine spezifische Aufgabe traininiert werden

1. Wie gut sind LLMs für argument Mining geeignet (Identifizierung von claims, premises und conclusions) bei der Anwendung von Zero-shot und few-shot learning
2. Können LLMs argumentative Strukturen (claim-premise-pairs) zuverlässig erkennen ?
3. Welche Herausforderungen gibt es bei der Anwendung von LLMs fürs Argument Mining? (Verschiedene Bestandteile von Argumenten, Mehrere Teilaufgaben,…)
4. Welche Prompt Engineering Techniken sind am Effektivsten zur Verbesserung der Perfromance eines LLMs zum Argument Mining?
5. Wie beeinflusst die Anzahl der übergebenen Beispiele die Leistung des LLMs?
6. Wie unterscheiden sich die Modelle mit unterschiedlichen Strukturen) dabei untereinander? Gemini, GPT4o, GPT2?
7. Wie effektiv sind LLMs bei der Anpassung an verschiedene Argumentationssstile bei verschiedenen Domänen (Politik, Recht, Wissenschaft)

# Literaturverzeichnis / Quellenverzeichnis

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. http://arxiv.org/abs/2005.14165

Cabrio, E., & Villata, S. (2018). Five Years of Argument Mining: A Data-driven Analysis. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 5427–5433. https://doi.org/10.24963/ijcai.2018/766

Cheng, L., Bing, L., He, R., Yu, Q., Zhang, Y., & Si, L. (2022). IAM: A Comprehensive and Large-Scale Dataset for Integrated Argument Mining Tasks. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2277–2287. https://doi.org/10.18653/v1/2022.acl-long.162

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3. Aufl.). O’Reilly Media, Inc.

Kochmar, E. (2022). *Getting started with Natural Language Processing*. Manning Publications.

Lawrence, J., & Reed, C. (2020). Argument Mining: A Survey. *Computational Linguistics*, *45*(4), 765–818. https://doi.org/10.1162/coli\_a\_00364

Lu, R.-S., Lin, C.-C., & Tsao, H.-Y. (2024). Empowering Large Language Models to Leverage Domain-Specific Knowledge in E-Learning. *Applied Sciences*, *14*(12), 5264. https://doi.org/10.3390/app14125264

Maharjan, J., Garikipati, A., Singh, N. P., Cyrus, L., Sharma, M., Ciobanu, M., Barnes, G., Thapa, R., Mao, Q., & Das, R. (2024). OpenMedLM: Prompt engineering can out-perform fine-tuning in medical question-answering with open-source large language models. *Scientific Reports*, *14*(1), 14156. https://doi.org/10.1038/s41598-024-64827-6

OpenAI. (o. J.-a). *Chat. Temperature*. OpenAI Platform. API Reference. Abgerufen 20. Dezember 2024, von https://platform.openai.com/docs/api-reference/chat/create

OpenAI. (o. J.-b). *Optimizing LLM Accuracy*. OpenAI Platform. Docs. Abgerufen 20. Dezember 2024, von https://platform.openai.com/docs/guides/optimizing-llm-accuracy

Ozdemir, S. (2024). *Praxiseinstieg Large Language Models: Strategien und Best Practices für den Einsatz von ChatGPT und anderen LLMs* (F. Langenau, Übers.; 1. Aufl., deutsche Ausgabe). O’Reilly.

Patil, R., & Gudivada, V. (2024). A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs). *Applied Sciences*, *14*(5), 2074. https://doi.org/10.3390/app14052074

Peldszus, A., & Stede, M. (2013). From Argument Diagrams to Argumentation Mining in Texts: A Survey. *International Journal of Cognitive Informatics and Natural Intelligence*, *7*(1), 1–31. https://doi.org/10.4018/jcini.2013010101

Stab, C., & Gurevych, I. (2014). *Annotating Argument Components and Relations in Persuasive Essays*.

Stab, C., & Gurevych, I. (2017a). *Argument Annotated Essays (version 2)* [Dataset]. https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2422

Stab, C., & Gurevych, I. (2017b). Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, *43*(3), 619–659. https://doi.org/10.1162/COLI\_a\_00295

Trad, F., & Chehab, A. (2024). Prompt Engineering or Fine-Tuning? A Case Study on Phishing Detection with Large Language Models. *Machine Learning and Knowledge Extraction*, *6*(1), 367–384. https://doi.org/10.3390/make6010018

Tunstall, L., Werra, L. von, Wolf, T., & Géron, A. (2023). *Natural Language Processing mit Transformern: Sprachanwendungen mit Hugging Face erstellen* (M. Fraaß, Übers.; 2. Aufl.). O’Reilly.

Yeginbergen, A., Oronoz, M., & Agerri, R. (2024). Argument Mining in Data Scarce Settings: Cross-lingual Transfer and Few-shot Techniques. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11687–11699. https://doi.org/10.18653/v1/2024.acl-long.628

# Anhänge & Projektdokumentation

Projektdokumentation umfasst:

* Readme-Datei
* Instrumente
* Daten
* EDA
* Qualitätssicherung?
* Code der Datenaufbereitung
* Coder der Datenanalyse
* Weiterführende Materialien

## Modellvergleich

Modellvergleich als Tabelle aufarbeiten und in Anhang packen

|  |  |  |  |
| --- | --- | --- | --- |
| Modell | API-Kosten |  | Link |
| Gemini 1.5 Flash | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung  USD |  | https://ai.google.dev/pricing#1\_5flash |
| Gemini 1.5 Flash-8B | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung  USD |  | https://ai.google.dev/pricing#1\_5flash-8B |
| Claude 3.5 Sonnet | Ein Bild, das Text, Schrift, Screenshot, weiß enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Schrift, Screenshot, weiß enthält.  Automatisch generierte Beschreibung |  | https://www.anthropic.com/pricing#anthropic-api |
| GPT-4o | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |  | https://openai.com/api/pricing/ |
| GPT-4o mini | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |  | https://openai.com/api/pricing/ |
| Llama small 0B-8B | $0.0004 /1K tokens |  | https://www.llama-api.com/pricing |
| Llama small 8B-30B | $0.0016 /1K tokens |  | https://www.llama-api.com/pricing |
| Llama small >30B | $0.0028 / 1K tokens |  | https://www.llama-api.com/pricing |