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

~~Modulbezeichnung: Advanced Digital Business Strategy~~

Betreuer: Prof. Dr. Marcel Hebing

**Titel der Studienarbeit**

**Argument Mining mittels eines Large Language Modells unter Verwendung von Prompt Engineering Techniken**

**/**

**Anwendung von Prompt Engineering Techniken auf LLMs für Argument Mining**

**/**

**Automatische Erkennung von Argumentationsstrukturen durch Large Language Models und Prompt Engineering**

**/**

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

**/**

**Argument Mining durch optimierte Prompts: Einsatz von Large Language Models für die Argumentationsanalyse**

Exposé zur Masterarbeit

Eingereicht von Benjamin Fels

Matrikelnummer: 190200

Datum: 28.01.2023

Inhaltsverzeichnis

[1 Themen zur Klärung mit Prof. 1](#_Toc181076554)

[2 Offene Themen für Untersuchung 1](#_Toc181076555)

[3 Einleitung 3](#_Toc181076556)

[4 Stand der Forschung 6](#_Toc181076557)

[4.1 Allgemein 6](#_Toc181076558)

[4.2 Transfer-Learning 7](#_Toc181076559)

[4.3 Infos zu LLMs 8](#_Toc181076560)

[4.4 Prompting / Prompt Engineering 9](#_Toc181076561)

[4.5 Retrieval-Augmented Generation (RAG) 11](#_Toc181076562)

[4.6 Argument Mining /argumentation mining 11](#_Toc181076563)

[4.7 Argument 13](#_Toc181076564)

[4.8 Aufgaben des Argument Minings 14](#_Toc181076565)

[4.9 Argument diagramming techniques / Argument Schemes 23](#_Toc181076566)

[4.10 Datensatz 23](#_Toc181076567)

[4.11 Evaluation / Evaluationsmetrik 27](#_Toc181076568)

[5 Fragestellung 28](#_Toc181076569)

[6 Methodisches Vorgehen 29](#_Toc181076570)

[7 Erwarteter praktischer Mehrwert 31](#_Toc181076571)

[8 Gliederung 31](#_Toc181076572)

[9 Zeitplan 31](#_Toc181076573)

[10 Literaturverzeichnis / Quellenverzeichnis 33](#_Toc181076574)

# Themen zur Klärung mit Prof.

* Wie genau muss das Exposé bereits sein?: Müssen
  + Auswahl des Modells,
  + Evaluationsmetrik,
  + etc, feststehen?
* Wie genau Fokus schon abstecken?
  + Bildliche Darstellung der Argumentationsstruktur,
  + Fine-tuning ja/nein?
  + Experimente beschreiben?
* Erneute Präsentation gewünscht?
* Beiträge und Berichte wissenschaftlicher Konferenzen oder Tagungen sind als Quelle für deine Abschlussarbeit gut geeignet. <https://www.scribbr.de/apa-standard/beispiel/konferenzbericht-konferenzbeitrag/>
* Evaluation von Argumentationsgraphen auf eine nicht-manuelle Weise, Gibt es Metrik dafür?

# Offene Themen für Untersuchung

1. Praxisorientierte Arbeit?
2. Welche Methoden zu customization verwenden?
3. Modell wählen , GPT, BERT, T5,…
   1. Licence for research?
   2. Gemini
   3. GPT 4o /mini
   4. GPT3 Playground
   5. Sonstige
      1. Llama
      2. BERT
      3. T5
      4. Mistral NeMo

|  |  |  |  |
| --- | --- | --- | --- |
| Modell | API-Kosten |  | Link |
| Gemini 1.5 Flash | USD |  | https://ai.google.dev/pricing#1\_5flash |
| Gemini 1.5 Flash-8B | USD |  | https://ai.google.dev/pricing#1\_5flash-8B |
| Claude 3.5 Sonnet |  |  | https://www.anthropic.com/pricing#anthropic-api |
| GPT-4o |  |  | https://openai.com/api/pricing/ |
| GPT-4o mini |  |  | 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 |

1. Evaluationsmetrik wählen
   1. BERTScore: misst semantische Ähnlichkeit. Soll es auf huggingface geben.
   2. Klassische Klassifikationsmetriken: Accuracy, Precision, Recall, F1-Score,…

MindMap zum Thema siehe: <https://miro.com/app/board/uXjVLddp4KY=/>

# Einleitung

1. Worum geht es in deiner Masterarbeit, was ist die Problemstellung

deiner Arbeit und warum / für wen ist das relevant?

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. Dies gilt insbesondere für strategische Dokumente, Wahlprogramme und journalistische Artikel, die oft eine Fülle von Informationen enthalten.

Gute Argumente die Grundlage, um ein jemanden zu überzeugen und für eine Entscheidungsfindung bei verschiedenen Standpunkten (Stab & Gurevych, o. J., S. 1501). Sie führen an, dass die automatisierte Erkennung von Argumenten in Texten dazu beitragen kann, die Plausibilität zu prüfen.

Argumente sind wichtiger Bestandteil in der Menschlichen Kommunikation. Standpunkte werden anhand von Beispielen bestärkt und sollen die andere Seite von dem Standpunkt überzeugen.

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

Der Bereich, welcher sich mit der automatisierten identifikation von Argumenten in Texten und die Beziehung untereinander auseinandersetzt ist Argument Mining.

Dieser Beriech kann in die Teilaufgaben argument detection und argument stance classification unterschieden werden (Marr & Ward, 2019, S. 58). Demnach werden beim ersteren werden die Argumente und Beweise und beim Letzteren die Haltung der Bestandteile der Argumente zueinander identifiziert.

Ziel dieser Masterarbeit ist es, eine Methode zu entwickeln, mit der Kernaussagen aus solchen Texten extrahiert und deren Argumentationsstrukturen in Form von Strukturdiagrammen visualisiert werden können. Dies soll die Verständlichkeit und Nachvollziehbarkeit der Texte erhöhen und die Effizienz der Informationsverarbeitung verbessern.

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

Künstliche Intelligenz kann den Menschen bei seinen täglichen Aufgaben unterstützen.

LLMs liefern dem Stand der Technik entsprechende Ergebnisse bei gängigen Natural Language Processing (NLP) Aufgaben mit weniger gelabelten Daten (Ozdemir, 2024, S. 46). Zu diesen NLP-Aufgaben gehören beispielsweise, maschinelle Übersetzung, Beantwortung von Fragen und Textgenerierung (Kochmar, 2022).

Die herausragende Leistung zeigt sich unter anderem darin, dass es im neuesten ChatGPT verwendet wird. Aber auch andere Unternehmen wie Meta mit Llama, Google mit Geminin und BERT setzen auf die Transformer-Architektur.

„After its inception, transformer soon became the de facto standard for natural language tasks.” (Patil & Gudivada, 2024, S. 6)

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.

N-Shot Learning als Transfer Learning. Transfer Learning = It involves training a model on one task and applying the learned knowledge to a different but related task

LLMs werden mit großen Text-Korpora vortrainiert und können dann mittels fine-tuning für spezifische Aufgaben angepasst werden (Ozdemir, 2024, S. 52).

Die Integration von domänenspezifischen Wissen in LLMs stellt eine Herausforderung dar. Diese Untersuchung soll einen Einblick in die verschiedenen Methoden ermöglichen.

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.

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

Es soll deshalb untersucht werden, ob sich LLMs für Argument Mining unter der Anwendung von Prompt Engineering Techniken eignen.

Es werden pre-trained Large Language Models / Generative Language Models betrachtet

Für weitere Untersuchungen: Die generalistischen Modelle (foundation) können mit der Performance von spezialisierten Modellen (fine-tuned) verglichen werden. Untersuchung bei anderen Sprachen (multilingual)

# Stand der Forschung

2. Auf welchen (wissenschaftlichen) Vorarbeiten willst du aufbauen?

Ruhig auch Verweis auf bestehende Working Paper, sofern diese sehr neu sind und das Thema Prompt Engineering + Argument mining + LLMs gut abbilden.

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.

## Allgemein

Sprachmodelle sind Modelle, welche die Abfolge von Buchstaben oder Wörtern in Texten vorhersagen (Patil & Gudivada, 2024, S. 4).. Eine spezielle Form sind die großen Sprachmodelle, oftmals als Large Language Models (LLMs) bezeichnet. Diese wurden auf einem großen Textkorpus trainiert / besitzen eine wesentlich größere Anzahl an Parametern und können eine Vielzahl von sprachbasierten Aufgaben wie Übersetzungen, Zusammenfassungen und Fragenbeantwortung durchführen (Patil & Gudivada, 2024, S. 4). Ein bekanntes LLM ist Generative Pre-trained Transformer (GPT), welcher sich hinter ChatGPT verbirgt.

Transformer sind eine Art einer Architektur von Neuronalen Netzen. LLMs als fortschrittliche KI-Modelle

Lu et al. (2024) konnten in ihren Untersuchungen feststellen, dass ein LLM, welches im *pretraining* zusätzlich mit domänenspezifischen Informationen aus dem Bereich E-Learning angereichert wurde, eine bessere Leistung erzielen konnte, als das herkömmliche LLM. Dies konnte besonders für nicht weit verbreitete Informationen, wie aus neu veröffentlichen wissenschaftliche Artikel nachgewiesen werden. Darin wird für weitere Untersuchungen auf die Anwendung von Prompting-Techniken zur Verbesserung des LLMS hingewiesen.

Zur Überwindung von aufgabenspezifischen Architekturen wurden *pretrained language models* (PLMs) verwendet (Patil & Gudivada, 2024, S. 2). PLMs benötigen einen großen Korpus an unbeschrifteten Daten, um bei verschiedenen Aufgaben effektiv zu sein (Patil & Gudivada, 2024, S. 3). Nach Patil & Gudivada (2024, S. 2) liegt der Fokus von aktuellen Arbeiten auf einem aufgabenunabhängigen Ansatz, bei dem *transfer* und *finetuning* des *self-attention* Blocks ausreichend sind. Diese transformerbasierten Modelle können für verschiedene nachfolgende Aufgaben *finetuned* werden, ohne das eine Anpassung der Modellarchitektur notwendig ist (Patil & Gudivada, 2024, S. 2–3).

Trad und Chehab (2024) haben die Effektivität von Prompt-Engineering und Fine-Tuning bei der Erkennung von phising-URLs untersucht, mit dem Ergebnis, dass Prompt-Engineering eine schlechtere Leistung hervorbringt, als Fine-Tuning.

Pre-training und fine-tuning führen zwar meist zu einer besseren Performance gegenüber den Grundmodellen (foundation-modell), sind jedoch aufgrund der Kosten für Rechenkapazität und der Komplexität der Daten für Personen und Organisationen mit großem Budget vorbehalten.

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.

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)

## Transfer-Learning

Laut Géron (2022, S. 373) ist es weniger sinnvoll, ein eigenes tiefes neuronales Netz zu konstruieren, wenn man stattdessen ein bereits vortrainiertes neuronales Netz für eine ähnliche Aufgabe heranzuziehen kann. Dies hat den Vorteil, dass das Training viel schneller erfolgen kann und weniger Daten benötigt werden (Géron, 2022, S. 350). Diese Vorgehensweise nennt sich Transfer Learning (Géron, 2022, S. 350).

Einige Organisationen sind der Auffassung, dass das Training eines eigenen Modells nicht rentabel ist. Eine Alternative ist fine-tuning (Lu et al., 2024, S. 2).

Im Vergleich zum pretraining ist transfer-learning günstiger und konvergiert schneller, als wenn man das ganze Modell neu trainieren würde (Patil & Gudivada, 2024, S. 18).

Pretraining nutzt ungelabelte Daten, folgt dem self-supervised-Ansatz; Transfer-folgt supervised Ansatz mit für die geplante Nutzung gelabelten Daten. Pretraining-Datensatz von einer generischen Domäne, wohingegen bei transfer-learning die Daten von einer spezifischen Domäne stammen (Patil & Gudivada, 2024, S. 18).

Fine-tuning benötigt eine große Menge von den aufgabenspezifischen Daten (Patil & Gudivada, 2024, S. 18).

Für jede Aufgabe müsste ein eigenes Modell trainiert werden (Patil & Gudivada, 2024, S. 18).

## Infos zu LLMs

Patil & Gudivada (2024, S. 3) unterscheiden die Phasen der transformerbasierten LLMs in *pretraining*, *transfer learning* und *in-context learning*, welche jeweils wiederum unterteilt werden.

Sprachmodelle besitzen zwei Hauptschritte: *pretraining* und *transfer learning* (Patil & Gudivada, 2024, S. 3)*.*

Die *attention layer* hilft dem *decoder* dabei zu verstehen, welcher Input besonders wichtig ist (Patil & Gudivada, 2024, S. 4).

„In general, autoencoding models learn bidirectional contextualized representation suited for NLU tasks, whereas autoregressive models learn to generate the next token and hence are suited for NLG tasks.“ (Patil & Gudivada, 2024, S. 8)

Transformer ermöglichen parallele Verarbeitung, entgegen RNNs welche nur zur sequenziell (nacheinander) Berechnung in der Lage sind, dadurch verringert sich die Trainingszeit (Patil & Gudivada, 2024, S. 6).

Géron geht in seinem Buch auch auf GPT und BERT ein

## Prompting / Prompt Engineering

Es benötigt 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). Zwei Methoden aus dem Prompt Engineering, um damit umzugehen sind Zero-Shot- und Few-Shot-Learning (Tunstall et al., 2023, S. 189).

Prompting bezieht sich auf die Methode, einem LLM bestimmte Eingaben zu machen, um eine gewünschte Reaktion hervorzurufen (Maharjan et al., 2024, S. 8).

LLMs sind in der Lage Aufgaben zu erfüllen, für welche sie während des Trainings nicht trainiert wurden. Wei, J., Tay, Y., Bommasani, R., et al . Emergent abilities of large language models. Trans. Mach. Learn. Res. (2022). https://doi.org/ 10.48550/arXiv.2206.07682.

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.

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

Few-shot Learning erhöht die Leistung des Modells, ohne es feintunen zu müssen. D.h. die Gewichtung des Modells wird nicht verändert.

Few-Shot-Learning ermöglicht es dem LLM sich mit wenigen Trainingsdaten auf die neue Aufgabe anzupassen (Ozdemir, 2024, S. 136). LLMs besitzen die Fähigkeit von den Beispielen aus dem Prompt zu lernen (Tunstall et al., 2023, S. 330).

Das Hauptziel von Few-Shot Learning (FSL) ist die Verallgemeinerung beim Lernen aus einem kleinen Teil der Daten (Yeginbergen et al., 2024, S. 11690). Dabei werden die Daten anhand von zwei Methoden aufgeteilt:

* k-shot: jede der Teilmengen enthält genau k Argumentkomponentenfolgen von Behauptung und Prämisse.
* k-percent: berechnen des k-Anteils für jede Argumentkomponente aus den vollständigen Daten, um die Verteilung widerzuspiegeln.

In dem Zusammenhang ist es wichtig auch Beispiele aufzunehmen, welche keine Argumentationskomponenten beinhalten (Yeginbergen et al., 2024, S. 11690).

„Neuere Untersuchungen der Forscher von Hugging Face legen nahe, dass ein solcher Ansatz dateneffizienter sein kann als einen benutzerdefinierten Head feinzutunen.“ T. Le Scao und A.M. Rush, »How Many Data Points Is a Prompt Worth?« (*https://arxiv.org/abs/2103.08493*), (2021).

In the context of Large Language Models (LLMs), a **primer** refers to an initial input or prompt provided to the model. It serves as the starting point for generating responses. When you interact with chatbot-based LLMs like ChatGPT, priming plays a crucial role in providing dynamic context. [By using primers, these models can generate more accurate, relevant, and contextually appropriate replies](https://promptengineering.org/unlocking-ai-with-priming-enhancing-context-and-conversation-in-llms-like-chatgpt/).

Idee: Zusätzlich zur Variation der übergebenen Beispiele dem Modell testweise mitteilen, was Argumente sind und wie man sie erkennt

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.

Prompt engineering as an important emerging skill for medical professionals: Tutorial. J. Med. Internet Res. 25. Epub ahead of print 4 October 2023. <https://doi.org/10.2196/50638>.

Nori, H., Lee, Y.T., Zhang, S., et al . Can generalist foundation models outcompete special-purpose tuning? Case study in medicine. http://arxiv.org/abs/2311.16452

### Few shot prompting

Brown, T., Mann, B., Ryder, N., et al . Language models are few-shot learners. in Advances in Neural Information Processing Systems. (Curran Associates, Inc.), 1877–1901.

* + Zufällige Auswahl der Beispiele
  + Systematische Auswahl der Beispiele

(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

*“* *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)

## Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) ist eine Technik, welche Suchtechniken zum Informationsabruf mit dem Textgenerierungsmechanismus von LLMs kombiniert (Muludi et al., 2024, S. 1). Es erhöht die Genauigkeit der generierten Antworten indem Informationen von externen Quellen berücksichtigt werden (Muludi et al., 2024).

Lu et al. (2024) konnten an dem Beispiel von spezifischen Informationen aus dem Bereich E-Learning zeigen, dass mit RAG eine bessere Leistung (gemessen am BERTScore) erzielt werden konnte, gegenüber einen herkömmlichen LLM.

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

## Argument Mining /argumentation mining

Stammt aus dem Bereich des Natural Language Processings (NLP).

(Lawrence & Reed, 2020, S. 766) definieren Argument Mining als automatische Identifikation und Extraktion der von Argumentationskomponenten und deren Struktur.

Es verfolgt dem Ziel automatisiert argumentative Strukturen und ihre Beziehungen aus einem Text zu identifizieren, mit dem Endziel, maschinenverarbeitbare strukturierte Daten für computergestützte Argumentationsmodelle bereitzustellen (Cabrio & Villata, 2018, S. 5427).

Argument Minig als automatische Entdeckung von argumentativen Textabschnitten sowie die Identifikation der relevanten Komponenten des Arguments (Peldszus & Stede, 2013, S. 2).

Argument Mining als automatische Identifizierung der Argumentkomponenten und die Klassifizierung der Beziehungen, die zwischen ihnen bestehen können (Yeginbergen et al., 2024, S. 11688). Argument Mining beinhaltet nach (Yeginbergen et al., 2024, S. 11687) in der Regel zwei Teilaufgaben: 1 Erkennung von Argumentkomponenten (Konzentration auf die Verortung von Argumenten und die Identifizierung ihrer Arten (z. B. Behauptungen und Prämissen) und 2 Klassifikation von Argumentationsbeziehungen, bei der die Beziehung zwischen zwei Argumentationskomponenten als unterstützend oder angreifend eingestuft wird.

Es gibt verschiedene Definitionen. Sie haben gemeinsam: automatisiert, Extraktion von argumentativen Textbereichen und deren Struktur bzw. Beziehung. Hierbei wird auch von Komponenten gesprochen

(Lawrence & Reed, 2020) geben in Ihrer Arbeit eine Übersicht über den damaligen Forschungsstand zum Thema Argument Mining. Demnach wurden *„Argument Mining-Techniken wurden erfolgreich entwickelt, um Details der argumentativen Struktur eines Textes zu extrahieren, wobei der Schwerpunkt auf verschiedenen Ebenen der argumentativen Komplexität liegt, wie es der Bereich und die Aufgabe erfordern. Für jede Aufgabe haben wir Arbeiten berücksichtigt, die unter Verwendung eines breiten Spektrums von Techniken, einschließlich statistischer und linguistischer Methoden, durchgeführt wurden. Wir haben eine Hierarchie von Aufgabentypen vorgestellt, die auf zunehmender argumentativer Komplexität basiert.“*

*„Während sich das Opinion Mining darauf konzentriert, zu verstehen, was die Nutzer über ein bestimmtes Thema oder Produkt denken, dreht sich das Argument Mining darum, warum die Nutzer eine bestimmte Meinung zu einem Thema oder Produkt haben.“* (Cabrio & Villata, 2018, S. 5428)

**Ein Bild, das Text, Diagramm, Schrift, Reihe enthält.

Automatisch generierte Beschreibung**

„ Einer der zentralen Aspekte der menschlichen Kommunikation ist die Argumentation: der Prozess der Vermittlung von Neigungen, Einstellungen oder Meinungen und der Versuch, den Partner dazu zu bringen, diese zu akzeptieren oder sogar zu übernehmen.“ (Peldszus & Stede, 2013, S. 1)

Doch was genau ist ein Argument?

## Argument

Menschen tauschen ihre Argumente über Sprache und Text aus in natürlicher Sprache (NL). Da Problem ist, dass Argumente nicht immer eindeutig im Text erkennbar sind.

Theorie zu Argumenten: Freeman, James B. 2011. Argument Structure: Representation and Theory. Springer

Das verstehen der argumentativen Struktur macht es nachvollziehbar, warum Menschen eine gewisse Meinung haben (Lawrence & Reed, 2020, S. 765)

Nach Peldszus & Stede (2013) besteht ein Argument aus Komponenten wie Behauptungen (claim) und Prämissen (premises), welche eine bestimmte Struktur durch die Beziehungen zwischen den Komponenten aufweisen.

(Peldszus & Stede, 2013, S. 6) definieren ein Argument als eine Reihe von Prämissen, die eine Schlussfolgerung unterstützen. Ein Argument ist somit ein Komplex von einer oder mehreren Prämissen, die für eine Behauptung angeführt werden. Prämissen und Schlussfolgerungen sind in dem Text ausgedrückte Behauptungen.

## Aufgaben des Argument Minings

Nach (Stab & Gurevych, 2017, S. 620) bestehen Argumente aus mehreren Argumentationskomponenten. Dies beinhaltet eine Behauptung und eine oder mehrere Prämissen. Eine Behauptung ist eine kontroverse Aussage und der zentrale Bestandteil eines Arguments, und die Prämissen sind Gründe für die Rechtfertigung (oder Widerlegung) der Behauptung. Darüber hinaus haben Argumente gerichtete argumentative Beziehungen, die die Beziehungen einer Komponente zu einer anderen beschreiben. Jede dieser Beziehungen zeigt an, dass die Ausgangskomponente entweder eine Rechtfertigung oder eine Widerlegung der Zielkomponente ist. Die Identifikation solcher Argumentationsstrukturen beinhaltet mehrere Teilaufgaben (Aufgabenbezeichnung auf S.621): 1) Trennung des argumentativen Texts von nicht-argumentativen Text (component identification), 2) Argumentationskomponenten nach Behauptungen und Prämissen klassifizieren (component classification) und 3) Identifizierung der argumentativen Beziehungen support vs. attack (structure identification). Laut den Autoren fehlt ein Ansatz, welcher alle diese Teilaufgaben abdeckt. 🡪 In (Stab & Gurevych, 2017) gibt es zu den Teilaufgaben in den Unterkapiteln detaillierte Erläuterungen. Es wurden für jede Teilaufgabe verschiedene Feature aus dem Text gebildet.

Das Modell zur Identifizierung der argumentativen Beziehung in (Stab & Gurevych, 2017, S. 645) kann zwischen support und attack unterscheiden. Es ist somit ein binärer Klassifikator. Der Standpunkt jeder Behauptung und Prämisse wird in support oder attack klassifiziert. Der Standpunkt jeder Prämisse ist im Typ ihrer ausgehenden Relation kodiert, während die Haltung jeder Behauptung in ihrem Attribut „Haltung“ kodiert ist.

Das in (Stab & Gurevych, 2017, S. 646) vorgestellte Modell erreicht nahezu menschliche Performance mit 95,2 % der menschlichen Leistung bei der Identifizierung von Komponenten, 87,9 % bei der Identifizierung von Beziehungen und 80,5 % bei der Erkennung von Haltungen.

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

Automatisch generierte Beschreibung

(Stab & Gurevych, 2017, S. 648)

Der Argumentationsstruktur-Parser ist eine Pipeline aus mehreren Schritten, weshalb sich Fehler am Anfang der Pipeline negativ auf die nachfolgenden Modelle / Schritte auswirkt. (Stab & Gurevych, 2017, S. 648–649)

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, o. J., S. 1501)

(Cabrio & Villata, 2018, S. 5428) benennen zwei wichtige Schritte im argument mining framework:

1. Argument extraktion: Identifikation von Argumenten im Text. Dieser Schritt kann wiederum in zwei weitere Schritte unterteilt werden.
   1. Erkennung von Argumentkomponenten: Behauptungen und Prämissen
   2. Identifizierung ihrer textlichen Grenzen
2. Vorhersage der Beziehungen: bezogen auf die Argumente aus Schritt 1. Die Beziehungen zwischen den Argumenten können Angriffe oder Unterstützungen sein. Angriffe sind negative Beziehungen zwischen zwei Argumenten, wohingegen eine Unterstützung eine Positive Beziehung zwischen zwei Argumenten ist. Anhand dieser Beziehungen können Argumentationsgraphen gebaut werden.

Argument Mining nach Peldszus & Stede (2013, S. 20): Die Art und Weise, wie die Schritte 2 bis 4 ausgeführt werden, hängt von der Art der zu erstellenden Strukturbeschreibung ab 🡪Wie komplex sollen die identifizierten Konfigurationen sein?

1. Für Argumentation relevanten Text extrahieren.
2. Segmentation: Text in Argumentative Diskurs Einheiten (ADUs) unterteilen. Bei einer mininmalen argumentations-analyse werden lediglich „Prämissen“ und „Schlussfolgerungen“ ermittelt.
3. Segment Classification: Rolle von jedem ADU bestimmen
4. Relation Identification: Herstellung von Beziehungen zwischen einzelnen ADUs, die möglicherweise zu einer vollständigen Baum- oder Graphenstruktur zusammengeführt werden können.
5. Argument Completion: unverständlich…

Es gibt verschiedene Theorien zu den Strukturen von Argumenten mit verschiedener Komplexität (Peldszus & Stede, 2013, S. 3–14)

Argumentation als Komplex aus mehreren zusammenhängenden Argumenten, wobei sich die Argumente untereinander unterstützen (support), angreifen (attack) oder kontern können (counter-attack) (Peldszus & Stede, 2013, S. 7)

* Support (Peldszus & Stede, 2013, S. 7): Schlussfolgerungen können unterstützt werden, Ein Bild, das Text, Reihe, Entwurf, weiß enthält.

  Automatisch generierte Beschreibung
  + indem ein weiteres Argument aufgeführt wird (Kannst du mir weitere Argumente für die Schlussfolgerung geben?) (multiple Support)
  + oder das bereits bestehende Argument weitereinwickelt/ergänzt wird (Warum sollte ich die Prämisse akzeptieren? (serial support)
  + Eine Spezialform ist die Nennung von Beispielen (Kannst du mir ein Beispiel nennen ?) (example support)
* Attack (Peldszus & Stede, 2013, S. 8–10): Ein Bild, das Text, Screenshot, Schrift, Diagramm enthält.

  Automatisch generierte Beschreibung
  + Argument gegen die Schlussfolgerung (Was macht dich so sicher in deiner Behauptung, in Anbetracht des Gegenarguments?) (rebut conclusion)
  + Stichhaltigkeit des gegebenen Arguments angreifen, indem man die Prämisse angreift. Ob die angegriffene Behauptung als Prämisse oder Schlussfolgerung dient ist für die Widerlegung unerheblich.
* Counter-Attack (Peldszus & Stede, 2013, S. 10–11): Als Möglichkeit des Befürworters auf die Attacken des Angreifers zu reagieren. M.E. ist das nur eine Verschachtelung von Attacken. Für die Untersuchung sollte eine einfache Unterscheidung in Support/Attack ausreichend sein.Ein Bild, das Diagramm, Text, Reihe, technische Zeichnung enthält.

  Automatisch generierte Beschreibung

Argumente bestehen aus **elementary discourse units (EDUs)** (Lawrence & Reed, 2020, S. 775). Das sind sich nicht überlappende Textabschnitte, die den atomaren Einheiten eines Diskurses entsprechen (Lawrence & Reed, 2020, S. 776). **Diskurs = strukturierte Art der Kommunikation**

*„Der zentrale Begriff für jede Theorie der Diskursstruktur ist der der Kohärenzrelation, d.h. die Vorstellung, dass benachbarte Textabschnitte in einer semantischen oder pragmatischen Beziehung zueinander stehen, wie z.B. Kausalität oder Kontrast*.“ (Peldszus & Stede, 2013, S. 15)

In diesem Zusammenhang wird auchvonArgumentative Discource Units (ADUs) gesprochen, welche minimale Analyseeinheiten sind (Lawrence & Reed, 2020, S. 776).

*„Die Identifizierung von ADUs ist jedoch wesentlich schwieriger als die Identifizierung von EDUs, da sie ein Verständnis für die argumentative Funktion der einzelnen Spannen erfordert.*“ (Lawrence & Reed, 2020, S. 790)

Extraktion der Argument Komponenten als sequence labeling task? (2282 Cheng et al., 2022). Auch (Stab & Gurevych, 2017, S. 636) und (Yeginbergen et al., 2024, S. 11688) bezeichnen Argument Mining als eine Sequenzetikettierungsaufgabe (sequence-labeling task).

POS-Tagging und NER sind auch sequence labeling tasks

Die größte Herausforderung ist die Identifikation der zentralen These des Textes, da die anderen Analyseschritte (support, attack) darauf basieren.(Peldszus & Stede, 2013, S. 27). Nach Peldszus & Stede (2013) unterscheidet sich die Schwierigkeit zwischen den Genres, weshalb ein Trainingskorpus mit unterschiedlichen Genres angestrebt werden sollte

Bei(Cheng et al., 2022) handelt es sich um ein Arbeitspapier! Dabei wird ein Klassifikationsmodell basierend auf den vortrainierten Modellen BERT und RoBERTa trainiert. Sie unterschieden in drei Hauptaufgaben und zwei weiterführende Aufgaben. Die Hauptaufgaben sind Claim extraction, Stance Classification und Evidence extraction. Claim extraction ist die automatische Extraktion von Behauptungen aus einem Text (Cheng et al., 2022, S. 2281). Bei Stance classification ist für jede Behauptung zu bestimmen, ob sie das Thema unterstützt oder bestreitet. (Cheng et al., 2022, S. 2281) Evidence Extraction: Extraktion der Beweise (Cheng et al., 2022, S. 2281). Die weiterführenden Aufgaben sind Claim extraction with stance classification – CESC: Behauptungen extrahieren und die Haltung zum Thema bestimmen (Cheng et al., 2022, S. 2281) und Claim-evidence pair extraction – CEPE: Behauptungen mit dazugehörigen Beweisen extrahieren. (Cheng et al., 2022, S. 2281).

Argument Mining Aufgaben nach (Cheng et al., 2022, S. 2277–2288)

* Context dependent claim detection – CDCD
* Claim stance classification – CSC
* Context dependent evidence detection – CDED

Anstatt Modelle für jeden einzelnen Schritt zu trainieren, soll sich die Lernfähigkeit von LLMs zu Nutze gemacht werden. (Cheng et al., 2022) haben mit CESC und CEPE bereits mehrere Einzelaufgaben miteinander kombiniert.

Für argument mining gibt es nicht ausreichend Trainingsdaten (Morio et al., 2022)

Die Ergebnisse von (Morio et al., 2022) zeigen, dass multi-task argument mining Modelle übertrifft, die auf einem einzigen Korpus trainiert wurden.

*„Unser Ansatz zum Parsen von Argumentationsstrukturen besteht aus fünf aufeinanderfolgenden Teilaufgaben, die in Abbildung 3 dargestellt sind. Das Identifikationsmodell trennt argumentative von nicht-argumentativen Texteinheiten und erkennt die Grenzen der Argumentationskomponenten. Die nächsten drei Modelle bilden ein gemeinsames Modell zur Erkennung der Argumentationsstruktur. Wir trainieren zwei Basisklassifikatoren. Das Modell zur Klassifizierung von Argumentkomponenten kennzeichnet jede Argumentkomponente als Hauptbehauptung, Behauptung oder Prämisse, und das Modell zur Identifizierung argumentativer Beziehungen erkennt, ob zwei Argumentkomponenten argumentativ miteinander verbunden sind oder nicht. Das Baumgenerierungsmodell optimiert global die Ergebnisse der beiden Basisklassifikatoren, um einen Baum (oder mehrere) in jedem Absatz zu finden. Schließlich unterscheidet das Modell zur Erkennung von Haltungen zwischen Unterstützungs- und Angriffsbeziehungen.*“ (Stab & Gurevych, 2017, S. 634) Bei den fünf Teilaufgaben werden diverse Features berücksichtigt, wodurch die Komplexität zunimmt (strukurell, syntaktisch,…). Das LLM hat bereits Erfahrung im Umgang mit Sprache. Es zieht sich aus den Beispielen die Muster.

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

Automatisch generierte Beschreibung

(Stab & Gurevych, 2017, S. 634)

Neben den 5 Modellen werden zum Proprocessing unter anderem lemmatizer und Part-of-speech-Tagger, constituent and dependency parser verwendet (Stab & Gurevych, 2017, S. 635). 🡪 Sehr aufwändig und benötigt viel Fachwissen. Alternativ über Sprachmodelle, welche gute Performance in NLP-Aufgaben haben. Die Idee dahinter ist, dass Modelle komplexe Strukturen eigenständig erkennen🡪 ggf. Abgleich, ob es besser ist die Aufgaben der 5 Modelle vom LLM nacheinander oder auf einmal bearbeiten zu lassen.

(Stab & Gurevych, 2017, S. 636) betrachten die Identifizierung von Argumentkomponenten als eine Sequenzetikettierungsaufgabe (sequence-labeling task) auf Token-Ebene, wobei das vollständige Essay als eine einzige Sequenz betrachtet wird.

* Dementsprechend werden das erste Token jeder Argumentkomponente als „Arg-B“,
* die Token, die von einer Argumentkomponente abgedeckt werden, als „Arg-I“
* und die nicht-argumentativen Token als „O“. bezeichnet

*„In Abbildung 3 wird gezeigt, wie diese automatischen Aufgaben miteinander verknüpft sind. Ausgehend von der Identifizierung von Argumentkomponenten durch Segmentierung und Klassifizierung als Teil des vorgebrachten Arguments oder nicht (diese Aufgaben werden manchmal gleichzeitig, manchmal getrennt durchgeführt, und manchmal wird die letztere Aufgabe ganz weggelassen), bewegen wir uns durch Ebenen zunehmender Komplexität: Erstens wird die Rolle einzelner Klauseln betrachtet (sowohl intrinsisch, z. B. ob es sich bei der Klausel um berichtete Rede handelt, als auch kontextuell, z. B. ob die Klausel die Schlussfolgerung eines Arguments ist); zweitens werden argumentative Beziehungen von einfachen Prämissen/Schlussfolgerungs-Beziehungen betrachtet; und drittens wird betrachtet, ob ein Satz von Klauseln eine komplexe argumentative Beziehung bildet, z. B. ein Beispiel für ein Argumentationsschema.“* (Lawrence & Reed, 2020, S. 787–788)

*aEin Bild, das Text, Screenshot, Diagramm, parallel enthält.

Automatisch generierte Beschreibung*

1. Identifizierung von Argumentkomponenten: Automatische Identifikation von argumentativen Sektionen im Text entspricht einer argument/non-argument klassifikation (Lawrence & Reed, 2020, S. 788). Der Kontext des Textes kann entscheidend sein, um zu bestimmen, ob es sich bei einem Satz um ein Teil eines Arguments handelt oder nicht –> siehe hierzu die Querverweise von (Lawrence & Reed, 2020, S. 789)
2. Rolle der Klauseln: Es geht darum, die Funktion der einzelnen Textabschnitte zu identifizieren, a) wobei zunächst die intrinsischen Eigenschaften betrachtet werden (z. B. ob ein Textabschnitt ein Beweis für eine Behauptung ist) und b) dann untersucht wird, wie ein Textabschnitt im Argument als Ganzes verwendet wird (z. B. Prämisse vs. Schlussfolgerung). (Lawrence & Reed, 2020, S. 791)
3. Identifizierung relationaler Eigenschaften: Generelle Beziehungen und Argumentative Beziehungen. Dies ist eine komplexere Aufgabe, als die Rolle der jeweiligen Komponenten zu bestimmen. Die Ansätze zur Identifizierung solcher Beziehungen bauen entweder auf der vorherigen Klassifikation / den Vorherigen Teilaufgaben auf oder versuchen die Beziehungen direkt zu extrahieren (Lawrence & Reed, 2020, S. 799)

Um die Argumente zu extrahieren wird Prompt Engineering verwendet.

*„Zur Veranschaulichung: Ein vollständiges Argument-Mining-System würde für den in Abbildung 9 angegebenen Text (zusammen mit seiner RST-Analyse) den oben in Abbildung 11 dargestellten Graphen erstellen. In der Praxis bauen die derzeitigen Systeme jedoch wesentlich einfachere Strukturen auf - was für bestimmte Zwecke ausreichend ist.*“ (Peldszus & Stede, 2013, S. 25)

Problem, dass auch bei menschlicher Extraktion der Argumente (und dazugehöriger Teile) Uneinigkeit hinsichtlich der Beschriftung besteht, da eine pragmatische Annotation eine subjektive Einschätzung beinhaltet (Peldszus & Stede, 2013, S. 27)

Während die Theorie von einer sauberen Auflistung der Argumente ausgeht, muss sich in der Praxis die segmentbasierte Annotation an den sprachlichem Stil des Autoren anpassen (Peldszus & Stede, 2013, S. 11). Die Pro- und Kontra-Argumente sind nicht immer direkt miteinander verbunden, sondern können im gesamten Text verteilt sein .

Aufgrund des zeitlich Begrenzten Rahmens der Untersuchung, wurde ein bereits annotierter Datensatz herangezogen. Eine Unterteilung in EDUs, ADUs, und die Bestimmung der Beziehungen war somit nicht notwendig. Neben dem geringeren zeitlichen Aufwand ist auch die eine höhere Qualität der Annotation von Vorteil. Die Annotation bei dem verwendeten Datensatz erfolgte von …

*Wörterbuch*

Claim – Behauptungen (Kann als Prämisse oder Schlussfolgerung dienen (Peldszus & Stede, 2013, S. 9))

Stances – Standpunkt / Haltung

Evidence – Beweise

Premise – Prämisse (Voraussetzung, Annahme)

Conclusion – Schlussfolgerung

## 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

Man könnte für jede Teilaufgabe einen eigenen Datensatz verwenden. Besser wäre es jedoch einen Datensatz zu verwenden, welcher sich für alle drei Teilaufgaben verwenden lässt. Damit bedarf eines annotierten Datensatzes, in dem die argument-Komponenten und den argumentativen Beziehungen ausgewiesen werden. Anhand der Grundwahrheit kann die Ausgabe des LLMs evaluiert werden.

*„Für beide Hauptaufgaben des Argument Mining werden qualitativ hochwertige annotierte Korpora benötigt, um die Leistung automatisierter Ansätze zu trainieren und zu bewerten. 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.

Es gibt verschiedene Datensätze, welche sich in ihrem Schwerpunkt und den Annotationen unterscheiden (Lawrence & Reed, 2020, S. 780–786).

*„Zwei der größten Probleme bei den derzeit verfügbaren Daten sind das Fehlen einer standardisierten Methodik für die Kommentierung und eines zentralen Ortes für die Speicherung und den Abruf von einheitlich formatiertem kommentiertem Material“* (Lawrence & Reed, 2020, S. 786)

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

Es wurden verschiedene Datensätze betrachtet. Um möglichst realitätsnah zu sein, sollte der Datensatz vorab nicht bereits von dem nicht-argumentativen Text befreit worden sein. (Stab & Gurevych, 2017, S. 620) weisen auf dieses und weitere Probleme mit bestehenden Datensätzen hin.

Die ersten beiden Datensätze stammen von UKP, TU-Darmstadt als starker Forschungsstandort. Dazu gibt es Paper

1: annotating argument components and relations in persuasive essays

* Frei zugänglicher Datensatz
* annotation of claims, premises, support and attack relation (argument components and argumentative relations) (Stab & Gurevych, o. J., S. 1501)
* 3 annotators, 90 persuasive essays; inter-rater agreement of 0.72 for argument components and 0,81 for argumentative relations 🡪 annotation scheme guides annotators to substantial agreement
* Essays stammen von essayforum.com
* Der finale Korpus enthält 1.673 Sätze mit 34.917 Token.
* Das Ziel des von uns vorgeschlagenen Annotationsschemas ist es, sowohl Argumentkomponenten als auch argumentative Beziehungen zu modellieren, die die argumentative Diskursstruktur in persuasiven Aufsätzen ausmachen. (Stab & Gurevych, o. J., S. 1503)

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

Automatisch generierte Beschreibung

2: Argument Annotated Essay Corpus (version 2) AAEC (Stab & Gurevych, 2017)

* Überarbeitung von Datensatz 1
* 420 von essayforum.com zufällig ausgewählte persuasive essays (S.630)
* Der Korpus umfasst 7.116 sätze mit 147,271 token.
* 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
* Baut auf dem vorherigen Datensatz auf, welcher als zu klein bezeichnet wird (S.630)
* Annotation Guidelines umfassen 31 Seiten
* Verwendung des brat rapid annotation tools <https://aclanthology.org/E12-2021.pdf> // <http://brat.nlplab.org/>
* Essays annotated by the expert annotator were used as training data (80%), essays annotated by the other annotators used as test data (20%)
* Wird von (Yeginbergen et al., 2024, S. 11688) als “perhaps the most popular NLP dataset manually annotated with argument structures” bezeichnet
* *„In einem überzeugenden Aufsatz wird ein bestimmtes Thema erläutert und versucht, das Publikum davon zu überzeugen, dass der Standpunkt des Sprechers die sachkundigste, logischste und gültigste Sichtweise des Themas ist. Weshalb sie sich gut für AM Aufgaben eignen”* (Cabrio & Villata, 2018, S. 5429)

3: IAM :(Cheng et al., 2022, S. 2279)

* <https://github.com/LiyingCheng95/IAM/tree/main>
* 123 debating topics
* Für jedes Thema wurden ca. 10 Artikel in englisch untersucht
* Insgesamt 1010 Artikel gesammelt
* Artikel wurden mittels NLTK-Packet in 69.666 Sätze unterteilt und anschließend annotiert. Somit Unterscheidung nur zwischen Sätzen und nicht innerhalb der Sätze möglich
* Für die Kontext abhängigen Beweise wurden 10 bis 15 Sätze vor und nach dem Satz mit der Behauptung durchsucht. Dabei ist auch möglich das die Behauptung selbst ein Beweis sein kann.
* Die Daten wurden von Professionellen Data-Annotators von einem darauf spezialisierten Unternehmen beschriftet. Diese wurden für ihre Tätigkeit bezahlt.
* Jeder Satz wurde von 2 professionellen „Annotators“ (Kommentatoren) unabhängig voneinander beschriftet. Dabei wurde ein Cohen´s kappa von 0,44 erreicht. Dies wird von den Autoren im Hinblick auf die Komplexität der Beschriftung als angemessen bewertet. Hierdurch wird auch die Komplexität des Argument Minings deutlich. Bei Unstimmigkeiten wurde ein dritter professioneller Kommentator hinzugezogen, um die Unstimmigkeiten zu beseitigen.
* Bereits bestehende Datensätze haben einen individuellen Fokus auf einzelne Aufgaben, wohingegen der IAM-Datensatz für die alle Komponenten für Argument mining beschriftet ist: claims, stances, evidence and relations among them.

4: Argument Mining ECHR

* https://github.com/PLN-FaMAF/ArgumentMiningECHR

## Evaluation / Evaluationsmetrik

“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).

# Fragestellung

Was ist deine Forschungsfrage (und ggf. Hypothesen)?

**Approach for Argument Mining using LLMs in Few-Shot and Zero-Shot Learning**

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

# Methodisches Vorgehen

4. Wie willst du vorgehen (Forschungsdesign /-methode, Gliederung, etc.)?

* Definition von Argument Mining, Argumenten, LLMs, Prompt Engineering (Abgrenzung zum fine-tuning und ggf. auch RAG)
* Vergleich von Zero-Shot und FewShot: Anzahl der Beispiele Variieren und Performance messen (Abweichung zur Grundgesamtheit). 0, 1, 2, 5, 10, 20, 50 Beispiele. Anschließend Evaluation an X Beispielen
  + Hypothese: Nutzen pro Anzahl der Beispiele verhält sich wie Grenznutzen-Theorem
  + Ggf. Ergänzung von weiterem Experiment bei der Ergänzung von Chain of Thought
* Verwendung eines Gelabelten Datensatzes zur Evaluation der Performance. Klärung Aufteilung Train-Test-Split
* Bestimmung geeigneter Metrik zur Evaluation
* Erläuterung, warum sich für ausgewähltes Modell entschieden wurde.
* Abwägung von Kosten und Nutzen pro Beispiel. Was kostet ein weiteres Beispiel. Kosten ergeben sich aus Anzahl Tokens des Prompts Multipliziert mit den Kosten pro Token.
* Speichern der Kombination aus Input Prompt und Output in einer strukturierten Datenbank und Berechnung der Evaluationsmetrik.
* Was sind beste Ansätze um argumente aus unstrukturiertem Text in strukturierte Form zu bringen (Graphviz als Visualisierungstool für Argumentationsstruktur?)
* Explorativer Use Case. Mögliche Methode identifizieren = Literaturrecherche + Ausprobieren. Auf Metaebene Problemstellung, Explorativ das Vorgehen weiter einzugrenzen
* Text to argument graphs
* Prompt Engineering Vorlage erstellen, mit welcher eine Überführung der Argumente in eine strukturierte Form möglich ist. Handlungsempfehlungen und Stolpersteine identifizieren.
* Experimente mit unterschiedlichen Labels? 2-label task: argument vs. No argument // 3-label-task: pro argument, con argument, no argument.
* Gegenüberstellung der Kosten eines Prompts über die API mit der daraus resultierten Genauigkeit.

1. Literaturrecherche: Untersuchung der bestehenden Methoden
2. Transfer Learning: Fine-Tuning eines GPT-Modells
   1. Beim Fine-Tuning wird ein vortrainiertes LLM auf einem spezifischen, kleineren Datensatz mit spezialisierten Inhalten trainiert. Hier könnte untersucht werden, welche Layer des Modells trainiert werden sollten, um die besten Ergebnisse zu erzielen, ohne die allgemeine Sprachkompetenz des Modells zu verlieren.
3. Alternativen zum Fine-Tuning: Adapter Layers und Prompt Engineering
   1. Adapter Layers: Anstatt das gesamte Modell zu trainieren, können spezielle Layer in das Modell integriert werden, um domänenspezifische Aufgaben zu unterstützen. Diese Ansatzweise spart Rechenleistung und vermeidet Overfitting.
   2. Prompt Engineering: Diese Technik konzentriert sich darauf, wie das Modell durch spezielle Eingabeanweisungen (Prompts) dazu gebracht werden kann, domänenspezifische Aufgaben besser zu lösen. Dies könnte eine interessante Methode sein, um die Anpassung ohne explizites Fine-Tuning zu untersuchen.
4. Evaluation
   1. Die Evaluation der angepassten Modelle erfolgt durch spezifische NLP-Aufgaben, die relevant für die gewählte Domäne sind. In der Medizin könnten dies Aufgaben wie Textklassifikation, Named Entity Recognition (z. B. Identifikation von Medikamenten oder Symptomen) oder Frage-Antwort-Systeme (QA) sein.
   2. Performanzmetriken wie Accuracy, F1-Score, Precision, Recall und Perplexity könnten herangezogen werden, um die Qualität des angepassten Modells zu messen.

Vergleich der Performance von Zero-Shot-Learning, Few-Shot-Learning und fine-tuning? 🡪 Erhöhung der Anzahl der Beispiele im Few-Shot-Prompting und die damit einhergehende Performance messen.

# Erwarteter praktischer Mehrwert

Wem nützt deine Arbeit, wer hat praktisch was davon und was?

Einblick in die Möglichkeiten zur Verwendung von Sprachmodellen in eigenen Anwendungsbereiche.

Strukturierter Überblick über die Argumentation (=Argument-Komplex) eines Textes und somit einfachere Plausibilisierung der Inhalte und Argumentationskette.

Als Entscheidungshilfe? Pro/Contra?

# Gliederung

1. Einleitung
2. Theoretische Grundlagen
   1. Argumente
   2. Argument Mining
   3. LLMs
   4. Prompting / Prompt Engineering
      1. Zero-shot
      2. One-Shot
      3. Few-Shot
      4. Chain-of-thought?
3. Datensatz
4. Methodik
5. Ergebnisse
6. Diskussion

# Zeitplan

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

Automatisch generierte Beschreibung

# 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

Marr, B., & Ward, M. (2019). *Artificial intelligence in practice: How 50 successful companies used artificial intelligence to solve problems*. Wiley.

Morio, G., Ozaki, H., Morishita, T., & Yanai, K. (2022). End-to-end Argument Mining with Cross-corpora Multi-task Learning. *Transactions of the Association for Computational Linguistics*, *10*, 639–658. https://doi.org/10.1162/tacl\_a\_00481

Muludi, K., Fitria, K. M., Triloka, J., & -, S. (2024). Retrieval-Augmented Generation Approach: Document Question Answering using Large Language Model. *International Journal of Advanced Computer Science and Applications*, *15*(3). https://doi.org/10.14569/IJACSA.2024.0150379

Ozdemir, S. (2024). *Praxiseinstieg Large Language Models: Strategien und Best Practices für den Einsatz von ChatGPT und anderen LLMs* (F. Langenau, Übers.; 1. Auflage, 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. (o. J.). *Annotating Argument Components and Relations in Persuasive Essays*.

Stab, C., & Gurevych, I. (2017). 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.; 1. Auflage, Deutsche Ausgabe). 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