SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Masters Thesis

# Investigating Complex Answer Attribution Approaches with Large Language Models

**Luca Mülln**

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Masters Thesis

# Investigating Complex Answer Attribution Approaches with Large Language Models

# Untersuchung von Ansätzen zur Attribution komplexer Frage- und Antwortszenarien mit großen Sprachmodellen (Large Language Models)

| | |
|---|---|
| Author: | Luca Mülln |
| Supervisor: | Prof. Dr. Florian Matthes |
| Advisor: | Juraj Vladika |
| Submission Date: | 15th March 2024 |

I confirm that this masters thesis is my own work and I have documented all sources and material used.

Munichs, 15th March 2024 Luca Mülln

# Abstract

This master's thesis explores the attribution of answers in complex question-answering scenarios utilizing large language models (LLMs). The research aims to assess and enhance the traceability of answers back to their source documents, a critical aspect of the modern Q&A setting using LLMs. By examining the relationship between the questions posed and the corresponding answers provided by LLMs, the study seeks to determine the extent to which these answers can be attributed to specific source materials and to identify the challenges and limitations inherent in the current attribution processes.

Through a methodical analysis structured around four research questions, the thesis investigates existing taxonomies for user needs and question structures, the creation of a specialized dataset for analysis, and the examination of LLMs' patterns in answering complex questions, weaknesses, and attributing responses. Comparative taxonomies are investigated and evaluated, and a new, two-dimensional taxonomy is proposed, which is adapted to the changed way of interaction with LLMs. Based on this taxonomy, an LLM-focused Q&A Dataset is provided that allows for direct comparison of different attribution methods. The effectiveness of existing methods for answer attribution is scrutinized, and error patterns that result in error propagation are investigated. As an artifact, a generalized framework for evaluating answer attribution is proposed that could potentially be independent of the domain and the complexity of the questions involved.

Based on the findings, the thesis proposes several new methods for evaluating and improving certain subtasks of the attribution process. This includes two methods for creating higher-quality claims for retrieval and attribution evaluation, and a framework for evaluating different retrieval methods in the context of answer attribution. The results are then evaluated in the context of different domains and the taxonomy, showing that the proposed methods are universally applicable and can be adapted to different use cases accordingly.
The findings are expected to contribute to the field by advancing our understanding of LLMs' performance in complex Q&A settings and by proposing refinements to the processes used to ensure accurate and reliable answer attribution.

# Kurzfassung

Diese Masterarbeit erforscht die Attribution von Antworten in komplexen Frage-Antwort-Szenarien (Q&A) unter Verwendung von großen Sprachmodellen (Large Language Models, LLMs). Das Forschungsziel ist es, die Nachverfolgbarkeit von Antworten zu ihren Quelldokumenten zu bewerten und zu verbessern, ein kritischer Aspekt in modernen Q&A Anwendungen unter Einsatz von LLMs. Durch die Untersuchung der Beziehung zwischen den gestellten Fragen und den entsprechenden von LLMs gegebenen Antworten wird angestrebt, das Ausmaß zu bestimmen, zu dem diese Antworten spezifischen Quellenmaterialien zugeordnet werden können. In diesem Kontext werden Herausforderungen und Grenzen identifiziert, welche in aktuellen Attributionsprozessen inhärent sind.

Mittels einer methodischen Analyse, strukturiert um vier Forschungsfragen, werden bestehende Taxonomien für Benutzerbedürfnisse und Fragestrukturen untersucht, die Erstellung eines spezialisierten Datensatzes zur Analyse und die Untersuchung von Mustern in den Antworten der LLMs, Schwächen und der Attribution von Reaktionen. Vergleichende Taxonomien werden untersucht und bewertet, und es wird eine neue, zweidimensionale Taxonomie vorgeschlagen, die an die veränderte Art der Interaktion mit LLMs angepasst ist. Basierend auf dieser Taxonomie wird ein LLM-fokussierter Frage-Antwort-Datensatz bereitgestellt, der einen direkten Vergleich verschiedener Attributionsmethoden ermöglicht. Die Wirksamkeit bestehender Methoden für die Antwortattribution wird hinterfragt, und Fehlermuster, die zu einer Fehlerfortpflanzung führen, werden untersucht. Als Artefakt wird ein generalisiertes Framework für die Bewertung der Antwortattribution vorgeschlagen, das potenziell unabhängig vom Bereich und der Komplexität der beteiligten Fragen sein könnte.

Basierend auf den Erkenntnissen werden mehrere neue Methoden zur Bewertung und Verbesserung bestimmter Teilaufgaben des Attributionsprozesses entwickelt. Dies umfasst zwei Methoden zur Erstellung von hochwertigeren Behauptungen für die Wiederbeschaffung und Bewertung der Attribution sowie ein Framework zur Bewertung verschiedener Wiederbeschaffungsmethoden im Kontext der Antwortattribution. Die Ergebnisse werden im Kontext verschiedener Domänen und der erstellten Taxonomie evaluiert, wobei gezeigt wird, dass die entwickelten Methoden universell anwendbar sind und signifikant bessere Ergebnisse erzeugen als vergleichbare Methoden.
Es wird erwartet, dass die Erkenntnisse zum Forschungsgebiet beitragen, indem sie unser Verständnis der Leistung von LLMs in komplexen Frage-Antwort-Szenarien voranbringen und Verfeinerungen der Prozesse vorschlagen, die verwendet werden, um eine genaue und zuverlässige Antwortattribution zu gewährleisten.

# Contents

# 1. Introduction

This chapter aims to provide a motivation for investigating the attribution of answers in complex Q&A tasks for LLMs. Based on the motivation, a precise problem statement is formulated and the research questions are derived. The chapter concludes with an outline of the research approach and the structure of the thesis.

## 1.1. Motivation

With the ongoing rise in popularity and increase of capabilities of Large Language Models (LLMs) and their various applications, the way information is accessesed and retrieved is noticeably changing. This increase in attention accross industries, research and the public eye, which can be traced back to the publication of the GPT-3 model by OpenAI in 2020 [1] and the respective continuation with ChatGPT (GPT3.5 in the following) [2], has opened up many new use cases and applications for LLMs.

The central importance of attribution in the answers provided by LLMs stems primarily from their propensity to produce "hallucinations," or responses that seem plausible but are either factually incorrect or lack grounding in reliable sources. In combination with the natural and convincing output format in most languages, evaluating truthfulness and sources of answers and claims is essential. This phenomenon especially raises concerns regarding the reliability and trustworthiness of LLMs, in critical applications as well as in day-to-day usage [3]. The challenge is compounded by the inherent "black box" nature of these models, where the internal workings and decision-making processes are opaque and not readily interpretable. This lack of transparency in how answers are derived poses a fundamental barrier to the responsible deployment of LLMs, particularly in domains where accuracy and accountability are paramount.

Furthermore, the absence of clear attribution or explanation for the answers generated by LLMs underscores a critical gap in their design. As these models continue to permeate various sectors — from healthcare [4] and finance [5] to education [6] and customer service — the need for verifiable and accountable information becomes increasingly crucial. The ability of LLMs to convincingly generate text that mimics human expertise, without the capability to provide an audit trail or source validation, raises ethical and practical questions. In scenarios where LLMs provide incorrect information or biased content, the inability to trace back the origin of such responses can lead to misinformation and potential harm.

Methods and frameworks for providing attribution or evaluating the provenance of answers generated by LLMs start to emerge and gain popularity. However, these methods are applications built on top of the LLMs and their output and therefore face the exact same drawbacks as the models themselves: The retrieved sources for specific claims and their respective entailment is frequently hallucinated as well, as current publications and user tests with attributed language model applications depict [7, 8].

The final point underscoring for the motivation of this research is the ongoing discourse surrounding the possible copyright infringement implications for organizations utilizing Large Language Models (LLMs) [9]. Major corporations such as OpenAI and Google, which are involved in training and operating LLMs, have yet to disclose their training data publicly. There is increasing evidence to suggest that this data may include copyrighted content. A notable example of the legal complexities involved is seen in the legal actions taken by news organizations, including The New York Times, against firms like Microsoft and OpenAI. These organizations have raised concerns over the replication of full news articles by GPT models [10]. Such legal developments highlight the importance of ensuring transparency and accurate attribution in the application of LLMs, paralleling the standards of source citation in conventional scholarly work.

In light of these considerations, the motivation for this thesis is to explore and address the challenges associated with the attribution of answers provided by LLMs. By examining the underlying mechanisms that contribute to the "black box" nature of these models, and investigating methodologies to enhance transparency and accountability, this research aims to pave the way for more reliable and attributable applications of LLMs. The goal is to contribute to the development of frameworks and tools that enable users to understand the provenance of information provided by LLMs, thereby fostering trust and promoting responsible usage in diverse domains.

## 1.2. Problem Statement

Based on the above statet motivation for attributing outputs of LLMs in various applications, the problem statement arise naturally and is formulated as follows:

**Overarching Problem Statement** Given a question or query $q$ and a response $r$ from a LLM, how can we identify the source $s$ of the response $r$ and verify it's relation to the response, defined as attribution $A$?
Where:

1. A query $q$ is defined as any natural text requesting an answer in arbitrary informational depth.
2. An response $r$ is the models answer to the given user query.
3. An response $r$ is segmentable into a set of individual (atomic) claims $C$ with $c$ being one claim and $c \in C$.

4. The set of claims $C$ is mutually exclusive and collectively exhaustive to the answer $r$.
5. Each claim $c$ is verifiable without additional necessary context within the text of $c$ itself.
6. The attribution $A$ is the function that evaluates the relation between the source $s$ and the answer $r$. 7. Each claim $c$ can be attributed to a retrieved source by an attribution type $a$ by the function $A$.

### 1.2.1. Objective & Research Questions

The objective of this thesis is to explore and address the challenges associated with the attribution of answers provided by Large Language Models (LLMs), and to improve on weaknesses as necessary or possible. The main goals arise from the depicted problem statements and motivation surrounding the attribution of LLM answers. The first objective is to identify and classify changes in information requests in interactions with LLMs, drawing from various existing and older research in information retrieval. This involves identifying the most common questions and queries posed to LLMs and categorizing them into a taxonomy. Based on this taxonomy, a dataset structure for evaluating attribution is to be developed.
The second objective aligns with an in-depth analysis and evaluation of existing approaches and subtasks of attribution. This includes identifying common patterns of weaknesses or errors in LLM responses and developing a framework to evaluate attribution that is independent of the domain and complexity of the question.
Based on the results of the initial two objectives, the following goal is to improve on weaknesses and subtasks in current attribution approaches or develop new approaches to improve on the weaknesses.
The final objective is to evaluate the developed approaches and frameworks across various domains and use cases to identify commonalities and differences in performance and applicability.
The following research questions are derived from the objectives and problem statements:

**RQ1: Classifying information needs in LLM based Q&A Tasks**

1.1 Can we identify more complex but frequently asked questions and categorize them domain-independently?

1.2 Is it possible to construct a dataset with quadruplets of (category, question, answer, attribution / source) based on previously established categorizations.

**Deliverables:** A taxonomy for question categories for user interactions with large language models.

A formal structure for a dataset which allows for the comparison of different attribution approaches. This stucture aligns alongside current research and unifies different approaches.

**RQ2: Patterns and Weaknesses of LLM Responses in Current Approaches**

2.1 How do current methods for answer attribution perform on the newly created dataset?

2.2 What are the common patterns of weaknesses or errors in LLM responses?

2.3 In what ways can these weaknesses be ameliorated?

**Deliverables:** A comparison of attribution approaches and individual segments of attribution within the previously build dataset.

A modular framework for testing individual attribution components and complete end to end approaches.

**RQ3: Improving Attribution and Attribution Evaluation**

3.1 Can we develop an abstract framework to evaluate attribution that is independent of the domain and complexity of the question?

3.2 Can existing attribution evaluation methodologies be adapted to complex answers?

3.3 How might existing metrics be enhanced to accommodate complex Q&A scenarios?

**Deliverables:** A structural analysis of existing and novel approaches and components for attributing claims.

**RQ4: Cross-Domain Performance**

4.1 Which domains have similarities to the questions cataloged initially?

4.2 Which domains vary significantly from those defined in RQ1?

4.3 How do our attribution and attribution evaluation methods fare across these various domains?

**Deliverables:** A structural analysis of the developed attribution aspects in the context of specific domains.

### 1.2.2. Research Approach

The methodology of this thesis is systematically aligned with its stated research questions and objectives. Each question is designed to sequentially build upon the findings of the previous one, ensuring a consistent research trajectory.

In RQ1, the task is to develop a taxonomy based on existing research datasets and objective analysis of question-and-answer pairs.

RQ2 aims to assess the performance of current attribution methods, utilizing the taxonomy and dataset created in RQ1, in conjunction with recognized benchmarks in the field of attribution research. The focus of RQ3 is to create a framework for attribution evaluation, addressing the weaknesses and patterns identified in existing methods.

This proposed framework is evaluated using the dataset framework from RQ1 and the analytical findings from RQ2. RQ4, the final phase of the research, involves a thorough evaluation of

this framework and taxonomy across a variety of domains, drawing upon the methodologies and findings established in the earlier stages.

Potential future work could involve applying this framework and taxonomy to additional domains and datasets, with the objective of evaluating and possibly improving the versatility and efficacy of these approaches in wider applications.

# 2. Related Work and Background

The motivation and research questions for this thesis open up a variety of topics that form the foundation for understanding the challenges in attributing answers provided by LLMs. This chapter offers an overview of these topics and the current state of research in their respective fields. The chapter is structured to align with the overarching research areas that contribute to the attribution of answers provided by LLMs. Additionally, this research review introduces fundamental definitions and concepts, some of which are novel and specific to this thesis. These definitions are marked as such. This is done to provide a comprehensive overview of the current state of research and to discuss naturally arising questions and challenges in the attribution of answers by LLMs.

## 2.1. Large Language Models

Language models (LMs) have been integral to the field of natural language processing (NLP) for several decades. The evolution of these models has seen significant advancements in architectures and methodologies, yet the core concept remains consistent: utilizing statistical methods for language comprehension and generation. Two primary model types have emerged as prominent in this evolution:

### 2.1.1. Encoder Models

Encoder models are designed to interpret and process language input, converting it into a meaningful, condensed representation. These models function by taking a sequence of words as input and producing an abstract representation of certain aspects of that sequence. The primary utility of encoder models lies in their ability to compress and encode input data for various NLP tasks, including, for example, classification, masked language modelling, sentiment analysis [11], part-of-speech tagging, and named entity recognition [12]. Notable examples of encoder models include BERT [13] and its derivatives such as RoBERTa [14], DeBERTA [15], as well as other architectures like XLNet [16] and ELECTRA [17].

### 2.1.2. Decoder Models

Decoder models are specialized in the domain of language generation, where they process varying scales of input and produce corresponding textual outputs. These models are integral in applications requiring the generation of text with human-like characteristics, such as language translation, content creation, and conversational interfaces. A prominent example of decoder models is the GPT series by OpenAI (e.g., GPT-3) [2, 1]. These models are

distinguished for their ability to generate text that is coherent and contextually appropriate, based on specified prompts. Unlike encoder models that primarily interpret or classify input text, decoder models focus on the generation of new text. The operational goal of decoder language models in the training phase is to minimize the loss in a training corpus $U = u_1, u_2, ..., u_N$, where $u_i$ represents an individual token, as represented in the following equation:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log\left(P(u_i|u_{i-k}, ..., u_{i-1}, \theta)\right) \tag{2.1}$$

In this equation, $\theta$ represents the model parameters, and $k$ is the context window size. The training process involves predicting the next token $u_i$ based on the prior $k$ tokens. As the model's performance improves, its generated text becomes increasingly coherent and contextually relevant, iff the training copora is large enough and coherent itself.
Observations starting from the development of GPT-3 by OpenAI in 2020 [1] show that decoder models possess the capability to perform functions traditionally associated with encoder models. This includes classifying or analyzing input text based solely on language comprehension, especially when a task is framed as a prompt within the model's context window. This advancement suggests a shift in the focus of organizations developing LLMs towards creating models that serve broader reasoning purposes, rather than being limited to text generation, which increases their size accordingly.

Furthermore, while encoder and decoder models are capable of operating independently, there is a noticeable trend in the development of hybrid models that merge both functionalities. These encoder-decoder models are capable of understanding input text and generating relevant output, enhancing their utility in diverse Natural Language Processing (NLP) applications.

### 2.1.3. Contributors of the Performance Leap of LLMs

The advancements in LLM performance are evident across a spectrum of tasks and benchmarks, as illustrated by the logical reasoning tasks benchmark, as referenced in Table 2.1, and the Massively Multitask Text Embedding Benchmark (MTEB), designed for a range of Natural Language Processing (NLP) tasks [18]. This section explores a comprehensive, yet not exhaustive, array of critical factors that have contributed to what appears to be an exponential improvement in the performance of LLMs.

**1. Available Data**: The availability of large scale training corpora has been a key factor in the success of LLMs. Especially with the complex task of understanding natural language, abundant amounts of training data in high quality are key. The internet and the respective digitalisation of books, news articles and other divers text sources allows for an accessible collection of annotated training data, which exceeds almost every other media available media type. Given the loss fucntion of generating LLMs 2.1, every piece of high quality text can be

| Dataset | LogiQA 2.0 test | MNLI dev | ReClor dev | AR-LSAT test |
|---|---|---|---|---|
| **Size** | 1572 | 9815 | 500 | 230 |
| **Human avg.** | 86.00 | 98.00 | 63.00 | 56.00 |
| **human ceiling** | 95.00 | 100.00 | 100.00 | 91.00 |
| **RoBERTa** | 48.76 | 90.02 | 55.01 | 23.14 |
| **ChatGPT (API)** | 52.37 | 55.40 | 57.38 | 20.42 |
| **GPT-4 (Chat UI)** | 75.26(73/97) | 68/100 (68/100) | 92.00 (92/100) | 18.27 (19/104) |
| **GPT-4 (API)** | 72.25 | 64.08 | 87.20 | 33.48 |

Table 2.1.: ChatGPT and GPT-4 performance on the Logical multi-choice machine reading comprehension task (accuracy %), combined with the MNLI dev dataset.[19]

used to train the model.

**2. Model Architecture and Size**: Decoder-only transformer architectures, as described in 2.1.2, are a key aspect in explaining the leap in performance of LLMs. Multiple studies across various speech and language applications have demonstrated that transformer architectures are superior to the currently available alternatives for language tasks, such as RNNs or CNNs [20, 21, 22]. The superiority of transformer architectures is generally attributed to several factors, including their ability to process long sequences of text, parallelize the training process, and learn long-term dependencies through the self-attention mechanism [23]. In addition to these advantages, the size and scalability of current state-of-the-art transformer models play a crucial role as well. The number of trainable model parameters began to increase in 2018 from around 100 million [24] to up to approximately 100 billion for popular models. Although the increase in model size is correlated with improved model performance [1], it is important to note that size is not the sole determining factor. Previous approaches with significantly greater capacity have not achieved comparable results to current models.
**3. Training and Fine-Tuning**: A fine tuning strategy deployed with most state of the art LLMs is reinforcement learning from human feedback (RLHF) [25]. This approach differs from traditional supervised learning by incorporating iterative refinements based on human feedback. In RLHF, the LLM is initially trained with a supervised learning model, often leveraging substantial datasets to grasp basic language structures and contextual understanding. Subsequently, the model undergoes a fine-tuning process using reinforcement learning, where human feedback plays a pivotal role. In this phase, the model is presented with specific scenarios or tasks, and human trainers provide feedback on the model's outputs. This feedback is not merely binary (correct or incorrect) but often nuanced, guiding the model to understand and replicate human preferences in responses, tone, and complexity. This iterative process of training, feedback, and adjustment leads to a significant enhancement in the model's ability to generate relevant, coherent, and contextually appropriate responses. RLHF has been instrumental in bridging the gap between purely data-driven learning and the incorporation of qualitative, human-centric aspects of language understanding [26].

**4. Resources and Computational Power**: In conclusion, the previously discussed elements pertaining to the scale of the model architecture and the size of the training dataset require substantial resources to manage these demands effectively. This necessity is evident in the parallel advancements in the performance of processing units such as TPUs (Tensor Processing Units) and GPUs (Graphics Processing Units). More importantly, substantial financial investments from large corporations like Microsoft, NVIDIA, Google, Amazon, among others, are the key in achieving the high-level performances exemplified by models such as GPT-4. Notably, the training process for GPT-4 alone incurred an estimated cost of approximately $100 million for OpenAI [27].

### 2.1.4. Mixture of Experts

As mentioned above, model capacity plays a key role in the increase in output performance of LLMs. But current developements hint towards model size and performance correlation arriving at a plateau, while other strategies show more promising results. The most prominent examples for this are the architectures GPT4 by OpenAI [2] and Mixtral by Mistral AI [28]. Both models utilize the "Mixture of Experts" (MoE) architecture [29].
Fundamentally, MoE involves a system architecture wherein multiple specialized sub-models, termed as 'experts', are orchestrated to handle specific types of tasks or data segments. Each expert is typically a neural network trained on a subset of data or specific tasks, enabling it to develop a deep proficiency in that area. The core idea is to leverage this specialized expertise by dynamically routing different parts of an input, such as a query or a textual segment, to the most relevant experts. This routing is typically managed by a trainable gating mechanism, that decides which expert(s) should be engaged for a given input. This mechanism allows MoE-based LLMs to not only improve their overall performance by capitalizing on the specialized knowledge of each expert but also enhances computational efficiency [30]. By selectively activating only relevant experts for a given task, MoE models can manage resource allocation more effectively than monolithic models. This aspect of MoE is particularly advantageous in scaling LLMs, where managing computational resources and response accuracy are paramount. However, the complexity of managing multiple experts and the routing mechanism introduces challenges in training stability and model interpretability, which are critical areas of ongoing research.

## 2.2. Natural Language Inference

The task commonly referred to as "Natural Language Inference" (NLI) involves a classification problem consisting of a premise $p$ and a hypothesis $h$. The goal is to determine the relation between the two. The most commonly used relation types are *Entailment*, *Neutral*, and

*Contradiction*. A typical representation of this task is analogous to equation 2.4.1 with:

$$\text{NLI}_{\text{original}}(p,h) = \begin{cases} \text{Entailment} & \text{if } h \text{ is inferred from } p \\ \text{Contradicting} & \text{if } p \text{ contradicts } h \\ \text{Neutral} & \text{if there exists no clear relation between } p \text{ and } h \end{cases}$$

Multiple large-scale training datasets and benchmarks exist for this task, such as the SNLI [31] (Stanford NLI), MultiNLI [32], and ANLI [33] (Adversarial NLI) datasets. Each dataset consists of triplets of elements $p$, $h$, and $y$ as $D = (p,h,y)_i$, where $y$ is the label of the relation between $p$ and $h$. Each dataset has a specific focus, with SNLI being the first large-scale dataset comprising more than 500k labeled pairs, MultiNLI offering a more diverse and broader coverage, and ANLI being an adversarial dataset where the hypothesis is generated by an adversarial 'human and model in the loop' approach. The following are two examples from ANLI [33] to showcase the different relation types:

1. **Premise:** "The 1947 Washington State Cougars football team was an American football team that represented Washington State College in the Pacific Coast Conference (PCC) during the 1947 college football season. Phil Sarboe, in his third of five seasons as head coach at Washington State, led the team to a 2–5 mark in the PCC and 3–7 overall."
   **Hypothesis:** "Sarbe coached more than 4 seasons"
   **Label:** *Entailment*

2. **Premise:** "John Thomas Harris (May 8, 1823 – October 14, 1899) was a nineteenth-century politician, lawyer and judge from Virginia. He often referred to after the American Civil War as "Judge Harris", even after his election to Congress. He was the first cousin of John Hill."
   **Hypothesis:** "Harris was a judge from Maryland"
   **Label:** *Contradicting*

This dataset is particularly interesting for the task of attribution, as it represents the contextual length of sources with the *premise* and a claim as the *hypothesis*. In ANLI, the premises are significantly longer than the hypotheses, which corresponds to the setting of attributing a claim to a source. In other datasets, such as SNLI, the premises and hypotheses are of similar length, typically one sentence each, which is not representative of the task of attribution.
Each dataset is still used to benchmark different models and approaches, with most training cases combining multiple datasets to achieve optimal performance [34]. Prominent models for this task are transformer-based models, such as BERT [11], RoBERTa [14], and T5 [35]. Notably, DeBerta NLI [34] performs exceptionally well on ANLI, and current LLM models have reportedly been outperforming other models on the SNLI dataset with and "Entailment as Few-Shot Learners" (EFL) approach [36]. This again suggests the overall general capacity of LLMs in most NLP tasks, as discussed in the previous section on different dimensions of attribution.
It is important to note that data leakage as a reason for the superior performance of current LLMs on specific tasks like NLI cannot be ruled out, since the training datasets for these

models are not fully disclosed. It is reasonable to assume that some of the most prominent NLP-based datasets were used to fine-tune LLMs like GPT3.5 or GPT4.

## 2.3. Question Answering

Question Answering is the task of automatically providing a response to a question posed by a human in natural language [37]. This task is closely related to Information Retrieval (IR), as outlined in Section 2.6. The field of Question Answering (Q&A) can be categorized into two main types: **Closed Domain** Question Answering and Open Domain Question Answering. In Closed Domain Q&A, the answering system is confined to a specific domain and ontology, retrieving and responding with information from a predetermined set of documents or knowledge bases. In contrast, **Open Domain Question Answering (ODQA)** involves developing systems and models capable of answering questions across various domains without domain-specific constraints. In most instances, ODQA utilizes a broad collection of topics as a reference corpus [38]. While these two systems may share similar architectures, their performance and challenges differ markedly.

A Q&A system traditionally comprises three distinct subsystems, each responsible for a specific subtask [37]. These subsystems are:

1. **Question Analysis**: This subsystem analyzes the question and classifies the user's intent. Earlier systems, not based on LLMs, primarily focused on straightforward closed questions, where the answer is a single entity or a list of entities. Contemporary systems, however, are equipped to handle more complex queries, as detailed in Section 2.6.2.

2. **Information Retrieval**: This subsystem is tasked with retrieving relevant documents or passages from a reference corpus, typically employing a search engine or database. Further information on this task is provided in Section 2.6.

3. **Answer Extraction**: This subsystem aims to extract the answer from the retrieved documents or passages. Traditional methods involve using Named Entity Recognition (NER) or Information Extraction (IE) techniques [37], while modern approaches leverage LLMs to generate answers directly from the sourced material.

Note that these components align with those found in current "Retrieve then Read" (RTR) based systems like "Retrieval Augmented Generation" (RAG), as discussed in section 2.5.1. This similarity arises naturally, as the task of attributing an answer to a source initially requires an answering system.

With the enhanced performance of LLMs, as discussed in the preceding section, the domain of Q&A has undergone significant improvements as well. LLMs have not only proven to be effective in information retrieval [39], but they have also greatly advanced answer extraction through improved natural language understanding. Furthermore, LLMs possess the capability to generate answers directly, eliminating the need for a separate retrieval step

[1]. This advancement is particularly evident in current publicly available systems such as GPT-4 [2] and Mixtral [28]. The performance of these models is evaluated not only based on logical inference but also on their knowledge of the world. GPT-4, in particular, has shown remarkable results in knowledge-based and real-world tests, such as LSAT scores and the Uniform Bar exam, where it achieves above 88th percentile performance in both cases [2].

Being one of the most significant tasks in NLP and a key component in the lives of many people through search engines on the web, there are numerous benchmarks for Q&A systems, especially in the open domain. Two of the most notable datasets include the Stanford Question Answering Dataset (SQuAD) [40] and the Natural Questions Dataset by Google [41]. These datasets generally use the F1 score and exact matches to evaluate the output data.
For SQuAD, the authors selected a diverse range of Wikipedia articles and generated questions from the content of these articles using crowdworkers. The answer is expected to be in the corresponding paragraph of the article. Models are then fine-tuned on a training split of the SQuAD dataset after being pre-trained on large text corpora. SQuAD is a reading comprehension dataset, meaning that the context, which should contain the answer, is provided as input. The model predicts the start and end indices that are most likely to contain the answer. One of the best-performing models on SQuAD is the "Ensemble ALBERT" [42] architecture, which achieves an F1 score of 90.12.

## 2.4. Faithfulness and Factuality

With the increasing spread of Large Language Models across various domains and applications, the faithfulness and corresponding factuality of their outputs become critical factors. This section provides an overview of different factors contributing to a lack of factuality in LLMs, defines key terms in the context of factuality, and summarizes current surveys and research in this domain.

### 2.4.1. Definitions

In the assessment of "factuality" within Large Language Model outputs, distinct challenges are encountered. This concept can be contextualized either in relation to the training data (closed domain) or in reference to existing world knowledge (open domain). Despite stemming from similar foundational principles, these approaches significantly diverge in terms of their technical and practical implications, effectively constituting two completely separate tasks that require seperate methodologies.

From a technical perspective, factuality in LLMs is frequently defined by the content of the training data. In this context, an output is considered "factual" if it is derived directly from the training corpus. This approach parallels traditional objectives in Natural Language Processing (NLP), focusing on enhancing model performance and reducing loss. Enhancing factuality, in this sense, becomes an extension of the conventional model training methodolo-

gies. This more technical definition allows for a formal definition of factuality which, in this thesis, is based on Natural Language Inference (NLI):
Starting with the optimal scenario for a definition of factuality based on Natural Language Inference (NLI), the output $a$ of a LLM $L$ is either $true$, $false$, or $not\ referenced$ given the training data set $T$ as a reference corpus with $t \in T$ and $t$ being a sequence of tokens. Within this setting, the three states for the answer of a model are defined as follows:

1. $a$ is $true$ if $\exists t \in T : \text{NLI}(a, t) = true$
2. $a$ is $false$ if $\exists t \in T : \text{NLI}(a, t) = false$
3. $a$ is $not\ referenced$ if $\exists t \in T : \text{NLI}(a, t) = neutral$

where

$$\text{NLI}(a,t) = \begin{cases} \text{true} & \text{if } a \text{ is inferred from } t \\ \text{false} & \text{if } a \text{ contradicts } t \\ \text{neutral} & \text{if there is no clear relation between } a \text{ and } t \end{cases}$$

For this optimal definition to be applicable in real-world scenarios, the training dataset $T$ would need to meet a list of requirements that are almost unachievable in large training sets. These requirements include, but are not limited to, comprehensive coverage of all factual domains, up-to-date information, and the absence of any form of biases, inaccuracies, and contradictions. Given the dynamic and expansive nature of human knowledge, creating or maintaining such a dataset is impractical. Thus, assessing the factuality of LLM outputs based solely on the training data poses challenges, even though this assessment method is technically the only precise and 'fair' approach for both the model and its training process. In addition to the real-world based impracticalities, training procedures like Reinforcement Learning from Human Feedback (RLHF) complicate this further. With RLHF, even if a relevant training set exists for output reference, the model's state is influenced by more than just the training corpus. It is also defined by the human feedback within the training dataset. This factor makes referencing the output of a model even more complex. This complexity is compounded by the fact that most of the training data for currently available foundation LLMs, such as GPT-4 or LLamA, is not publicly accessible.

In contrast, a practical approach to factuality involves aligning outputs with real-world accuracy. Here, a LLM's output is deemed "factual" if it is consistent with verifiable real-world information. This perspective introduces more complex challenges, as it requires the development of new methods to assess the factuality of a LLM's output beyond its training data. Given that no training corpus can fully represent the entirety of real-world knowledge, such a corpus cannot serve as a definitive standard for factuality assessment. Additionally, the dynamic and broad scope of human knowledge surpasses the capability of a static training framework. Consequently, in this approach, the evaluation of a LLM's factuality necessitates consideration of external factors, including temporal context and the model's recognition of its own knowledge limitations. This necessitates the creation of more adaptable and

model-agnostic methods, metrics, and approaches for determining factuality and reliability. It is important to note that both definitions would converge in an ideal world, where the training data is comprehensive, up-to-date, and free from biases and inaccuracies. In that case, the model could be traditionally trained and regularly updated to maintain its alignment with real-world knowledge, and would, in this setting, face the same and more traditional deep learning challenges as in the first definition.

In light of these challenges and current use cases, factuality is often not defined as an intrinsic function of the training dataset and model output. Instead, it is considered a function agnostic of training data, with a reference to "real-world facts", as the second definition suggests. These "facts" are drawn from more or less trusted sources, such as Wikipedia or research papers. This approach can lead to the model producing technically "factual" output, in the sense that the output is directly inferred from the training corpus (as described in Section 2.4.1), but application-wise "un-factual" output, in the sense that the output does not align with real-world facts.
In the context of these definitions, factuality turns out to be more of a task of **attributability** than a task of providing factual outputs by a LLM. This perspecitve is even more supported by the fact that current research in LLM outputs suggests that hallucinations, which inherently contain false claims (as described in Section 2.4.2), are inevitable for any existing and computable LLM [43].

### 2.4.2. Hallucinations

Hallucinations are commonly referred to as outputs or completions by language models which are not factual, faithful to the provided source content, or overall nonsensical. More precisely, current research suggests that the definition of hallucination has considerably expanded due to the versatility of LLMs [44].
Zhang et al.[44] proposes three sub categories for hallucinations in LLM text generation:

1. **Input-conflicting hallucination**, where LLMs generate content that deviates from the source input provided by users;

2. **Context-conflicting hallucination**, where LLMs generate content that conflicts with information previously generated by itself;

3. **Fact-conflicting hallucination**, where LLMs generate content that is not faithful to established world knowledge.

While there exist some slight variations of these definitions and other categorizations in comparable publications, most are grounded in the same fundamentals and trains of thought. Other papers, such as Friel et al. [45], only differentiate between two categories of hallucinations, namely *Closed-domain halluciantions* and *Open-domain hallucinations*. The former refers to hallucinations that are not faithful to the source content, while the latter refers to hallucinations that are not faithful to established world knowledge. This pair of definitions is

known as the *intrinsic-extrinsic dichotomy* [43, 45, 46, 47].
Examples of input-output pairs for each of these categories are as follows:

1. **Input-conflicting hallucination / Closed-domain hallucinations**:
   *Input / User Query*: "'The quick brown fox jumps over the lazy dog' - Which color has the fox?"
   *Output*: "The color of the fox is orange."

2. **Context-conflicting hallucination**:
   *Input / User Query*: "What is a famous dish from china?"
   *Output*: "Fried rice is one of the most popular dishes in China. It is made from rice, eggs, and vegetables. Have fun eating your steak!"

3. **Fact-conflicting hallucination / Open-domain hallucinations**:
   *Input / User Query*: "What is the capital of Germany?"
   *Output*: "The capital of Germany is Berlin. It is located in the south of the country."

In this thesis, the focus lies heavily on *Input-conflicting hallucinations* in "Retrieve then Read" (RTR) Q&A settings, and *Fact-conflicting hallucinations* in "Post Hoc Retrieval (PHR) Q&A settings. *Context-conflicting hallucinations* are not heavily focused on, as the main task of "complex Q&A" which is analysed in this thesis is limited to single-turn Q&A settings and does not involve a context window of previous answers. It is important to note that *Input-conflicting hallucinations* and the model training specific definition of factuality (Section 2.4.1) differ. The former refers to hallucinations with reference to the directly provided input source content or user query, the latter refers to the factuality of the output with reference to the training data.

With the previously established definition of "factuality" and the precessing arguments in real world scenarios, it becomes clear that the definition of *Fact-conflicting hallucination* comes very close to the real-world based definition of factuality. Preventing these type of hallucinations is, as mentioned above, not an easy task, since real world knowledge is constantly changing and the training data for publicly available LLMs is iterating in much slower cycles.
Additionally, it has been indicated that hallucinations are inevitable for any existing and computable LLM, even in a formal setting where all knowledge was provided to the LLM in the training process [43]. In their publication, the authors create a formal world for LLMs and their output, in which a hallucination is defined as any deviation from a ground truth function for a given input string. In this construction, the authors formally prove that any LLM will hallucinate for every function that it can not compute, and that all computable LLMs will hallucinate for infinitely many inputs [43].

The concept of hallucinations in LLMs varies considerably between formal definitions and those inspired by real-world applications and use cases. For the purposes of this thesis, particularly in the context of attribution, emphasis is placed on definitions that are inspired by real-world scenarios and practical use cases. Typically, contemporary applications of LLMs

operate in less formalized environments, and as a result, they often encounter challenges related to fact-conflicting hallucinations. This focus acknowledges the prevalent need for addressing the complexities that arise when LLM outputs do not align with factual information in real-world settings.

### 2.4.3. Measurability and Detection of Hallucinations and Faithfulness

A key challenge in this domain is quantifying the factuality and incidence of hallucinations in LLM outputs. In line with the established definitions for 'factuality' and 'hallucinations', the process of evaluating these aspects in LLM outputs encompasses specific facets and challenges. For the purposes of this analysis, it is assumed that access to, or the ability to retrieve, every pertinent source content for each fact is feasible.

Recent studies highlight the emerging difficulties in detecting and measuring hallucinations in large language models. Traditional metrics demonstrate a weak correlation with human judgment [46], particularly in identifying hallucinations [47]. This observation is logical, given the complex and realistic nature of text generated by LLMs, and when considering the 'training data-agnostic' definition of *Fact-conflicting hallucination*. An exhaustive comparison of current static metrics, such as BLEU, ROUGE, and METEOR, and their correlation with human assessment of output faithfulness is presented in Pagnoni et al. [48]. These publications elucidate the challenge in reliably detecting faithful output, as indicated by the low correlation coefficients of these metrics. Static metrics primarily rely on N-grams, entities, or relationships between tuples of entities and generally compute some form of overlap metric to assess output faithfulness. This assessment is conducted in relation to a database of trusted sources, with each output compared against this database to determine a corresponding score. It is important to note that this method depends on the availability of a comprehensive source database for comparison against all potential claims.

More advanced and increasingly adopted methods are **classifier-based**, such as entailment or Natural Language Inference (NLI) models. As defined in Section 2.4.2, hallucinations are identifiable when an output does not correspond to its source content, or more precisely, is not inferred. While these methods offer improved performance over static metrics, their effectiveness is contingent on the performance of the underlying NLI models. Since NLI is focused on in this thesis, more information is provided in the respective Section 2.2.

Another noteworthy approach is uncertainty-based detection. This technique, rooted in Bayesian deep learning, utilizes the predictive entropy of the output distribution to characterize the total uncertainty of a prediction. Xiao and Wang (2021)[49] and Guerreiro et al. (2023a) [50] have demonstrated the efficacy of this method. They observed a positive correlation between the likelihood of hallucinations in data-to-text generation and predictive uncertainty, measured using deep ensembles or the variance in hypotheses generated by Monte Carlo

Dropout in Neural Machine Translation (NMT). Additionally, Van der Poel et al. (2022) [51] employed conditional entropy to assess model uncertainty in abstractive summarization. Another dimension of this approach is the log-probability-based method, where Guerreiro et al. (2023a) [50] use length-normalized sequence log-probability to gauge model confidence. Further, Miao et al. (2023) [52] introduced a model-based strategy, employing a system named SelfCheck for error detection in complex reasoning within LLMs, aggregating confidence scores through a systematic evaluation of each reasoning step [47]. While these methods have shown promise, they are not without limitations. The most prominent one being that especially for closed source LLMs, the complete model state is not available for the user and thus the uncertainty of the model is not directly accessible in some cases.

A final approach to measure and detect hallucinations involves leveraging the capabilities of LLMs themselves. This method centers around two primary strategies: direct faithfulness scoring and output-source pair evaluation. In the first strategy, the LLM is tasked with directly assessing a faithfulness score within its prompt, thereby enabling it to evaluate the reliability and accuracy of its generated text in real-time. This approach capitalizes on the model's inherent understanding of context and semantics, offering a self-referential method of quality control. The second strategy involves using a LLM to analyze an output-source pair. Here, the model scrutinizes the alignment between the generated content and the original source material. By evaluating the consistency and congruence between the output and the source, the LLM can effectively identify instances of hallucination, distinguishing between accurate representations and distortions or fabrications in the generated text. Both approaches signify a shift towards more autonomous and sophisticated methods of ensuring the veracity and reliability of LLM outputs.

Multiple benchmark datasets exist to evaluate factuality and hallucinations of models. They vary based on their underlying task, their generation type, their length and domains. Table 2.2 provides an overview of some of the most prominent datasets from Chen et al. [53].
Methods for **mitigating hallucinations** in LLMs can be broadly classified into two categories:

| Datasets | Response | | Granularity | Evidence | Scenario |
|---|---|---|---|---|---|
| | Length | Generated by | | Provided | Domain |
| FEVER | 7.3 | Human | Claim | ✓ | Wikipedia |
| FactCC | 20.8 | Synthetic | Sentence | ✓ | Newswire |
| QAGS | 16.1 | Model | Summary | ✓ | Newswire |
| WICE | 24.2 | Human | Claim | ✓ | Wikipedia |
| HaluEval | 36.9 | ChatGPT | Response | X | QA/Newswire |
| FELM | 89.1 | ChatGPT | Segment | ✓ | Five domains |

Table 2.2.: A comparison of published factuality benchmarks w.r.t model generated responses to be verified based on collected evidence [53]

**Data-Related Methods** and **Modeling and Inference Methods** [46]. The former encompasses techniques designed to refine the input dataset for the model. This refinement process aims to reduce noise, enhance data fidelity, and ensure greater coherence within the dataset. Such improvements in data quality inherently contribute to mitigating hallucinations by reducing the likelihood of the model encountering irrelevant or contradictory information [54]. One recent example is the construction of counterfactual examples, where the model is finetuned on counterfactual data to it's original training data in order to enforce it to rely on source information provided in the context window, instead of referencing learned information [55, 56]. Methods like these specifically focus on reducing *Input-conflicting hallucinations*.

In contrast, Modeling and Inference Methods encompass a wider range of techniques, focusing on modifications to the model's architecture, its information processing mechanisms, and output generation procedures. This includes alterations in the model's architecture, adjustments in the training and inference processes, and the implementation of post-processing techniques for the generated outputs. An exemplar of this category is Reinforcement Learning from Human Feedback (RLHF), a method that further fine-tunes the model to enhance its fidelity and reliability. Other examples are Multi-Task Learning and Controllable Generation. In **Multi-Task Learning**, a model is concurrently trained on multiple tasks, enabling it to grasp commonalities across these tasks and thereby reducing hallucinations. This approach, demonstrated by Weng et al. [57] and Li et al. [58], integrates additional tasks (like word alignment and rationale extraction) with the primary task, enhancing model robustness and faithfulness. The benefits of this method include improved data efficiency and reduced overfitting, although it also presents challenges in terms of task selection and model optimization [46, 54].
**Controllable Generation** addresses hallucinations by treating their severity as an adjustable attribute. Techniques such as controlled re-sampling and control codes, either manual or automatic, are applied to maintain low hallucination levels and improve content faithfulness. While requiring annotated datasets, this method allows for the adjustment of hallucination degrees to suit diverse application needs.

**Attribution**, in the context of mitigating hallucinations in LLMs, can be characterized as an inference-level approach. Unlike direct adaptations to the generated text for enhanced faithfulness, attribution primarily focuses on aligning user goals with the source of information. This method involves providing users with pertinent source documents and an automated assessment that correlates the LLM's response with these source documents. The objective is not to modify the generated text directly, but rather to offer transparency regarding the information basis of the LLM's output.

## 2.5. Answer Attribution

The preceding section outlines the characteristics, origins, and challenges associated with hallucinations and faithfulness in the context of language models. As stated in the final

paragraph of that section, *answer attribution* can be conceptualized as a sub-component in the broader framework of hallucination mitigation. Distinct from conventional methods of hallucination mitigation, which primarily focus on altering or enhancing the model's output, answer attribution is oriented towards end-users. This approach aims to equip users with a compilation of potential sources that inform the output of the language model. Such a strategy enhances transparency, allowing users to discern the faithfulness of the information provided by the model and to independently assess its reliability. In its fundamental implementation, answer attribution does not modify the model's response to elevate its quality; instead, it transfers the responsibility of quality assurance to the users themselves. In the light of the previously discussed challenges of faithfulness and hallucinations, this makes sense inherently. Since hallucinations are inevitable [43] and human knowledge is dynamic and ever changing, attributing the output of LLMs will most likely stay a crucial part of the deployment in real-world and open domain applications.
We provide a precise description for the task of attribution in the context of LLMs for this thesis as follows:

> **Answer Attribution** is the task of providing a set of sources that inform the output of a language model. These sources must be relevant to the model's response and should contain information that substantiates the respective sections of the response. The task is to be performed in real-time and is aimed at enhancing the transparency and faithfulness of the model's outputs.

This definition provides a comprehensive overview of the task and makes its inherent incapsulation of a list of subtasks transparent. These subtasks are:

1. **Answer Segmentation**: Segmenting the answer $r$ into individual claims $c_i$.
2. **Claim Relevance Determination**: Determining the relevance of each claim $c_i$ for the need of attribution.
3. **Finding Relevant Sources**: Retrieving a list of relevant sources $s_i$ for each claim $c_i$.
4. **Source Claim Inference**: Determining wether the list of sources $s_i$ actually supports a claim $c_i$.

The subsequent section aims to delineate the current advancements in answer attribution, emphasizing the methodologies and techniques employed for attributing the outputs of LLMs.

### 2.5.1. Answer Attribution Systems

Different approaches to answer attribution in LLMs can be categorized based on their integration into applications or systems. The primary distinction lies between **Post-Hoc Retrieval** (PHR) and **Retrieval Then Response** (RTR) systems. The category often referred to as

"Retrieval-Augmented Generation" (RAG) systems represents a specialized subset of RTR systems, equipped with additional functionalities.
The distinction between these systems hinges on the sequence of retrieval and response generation. In PHR systems, the answer generation process is entirely independent of the retrieval process, necessitating separate identification of relevant sources for each assertion. Conversely, RTR systems commence with a comprehensive retrieval of sources, and the subsequent answers are formulated based on these pre-retrieved documents. An effective RAG system, ideally, ensures that each assertion can be traced back to the sources initially retrieved. Figures 2.1 and 2.2 illustrate the differences between these systems.
Although these systems may appear similar, their differences are crucial for the task of an-

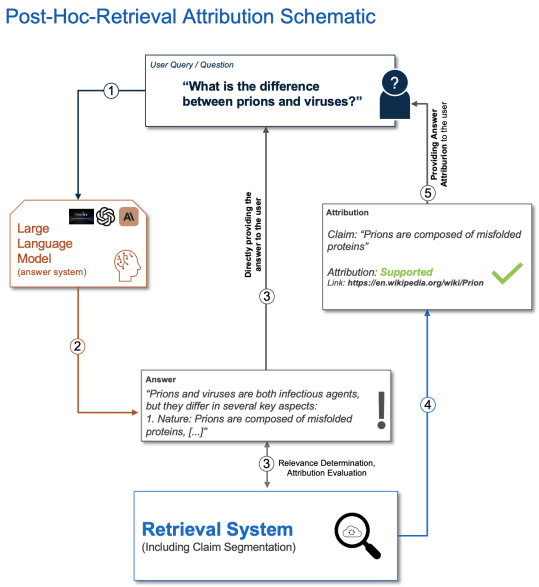


Figure 2.1.: PHR attribution schematic

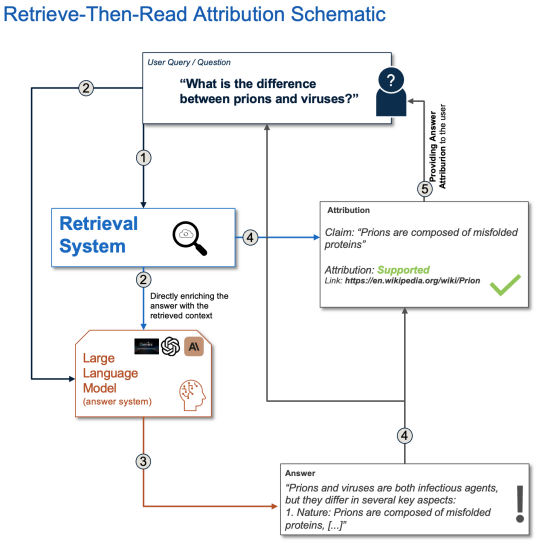


Figure 2.2.: RTR attribution schematic

swer attribution and the quality of the attributions [59]. This distinction is significant because RTR systems, relying on pre-retrieved documents, are less prone to generating hallucinations, since all the context is pre-retrieved [43, 54].
Nonetheless, specific subtasks, such as attribution evaluation, can be conducted independently of the system type, offering flexibility in research methodologies.

### 2.5.2. Prominent Platforms - Case Study

When examining the landscape of publicly accessible state-of-the-art applications, we can see that platforms such as OpenAI, BingSearch, and PerplexityAI, which allow users to directly interact with their respective LLMs through chatbot interfaces, have incorporated answer attribution features in their systems. Taking the popularity of both systems into account, they provide an interesting case study and starting point to research state of the art and deployed

applications with attribution.

Figures 2.3 and 2.4 illustrate the answer attribution mechanisms employed by OpenAI's

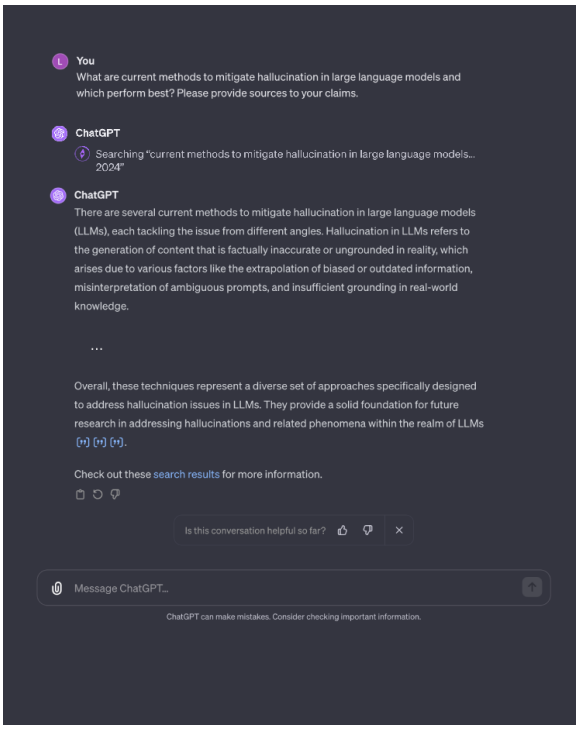


Figure 2.3.: OpenAI GPT4 attribution showcase using Bing Search

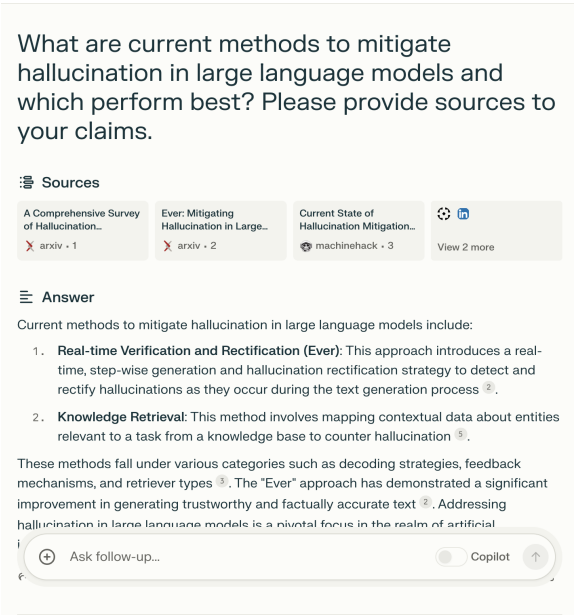


Figure 2.4.: PerplexityAI attribution showcase

GPT4 and PerplexityAI, respectively. Both systems enumerate a set of sources that informed the model's output. Users have the option to delve into these sources for further details. GPT4 appends a list of Bing search-derived sources subsequent to its complete response, whereas PerplexityAI integrates such a list following each section. However, it should be noted that there is no explicit verification indicating the support of these sources for the model's outputs. Recent publications suggest that even when certain segments of an answer are supported by cited sources within the application, these sources may not substantiate all the corresponding parts of the response [8]. Even though the exact background implementation and prompts for providing both answers are unknown, it is likely that both include a subset of the search results into the context window of the LLMs and answer the query using retrieval augmented generation (RAG). If the model hallucinates in those cases, these hallucinations can be characterized as *Input-conflicting hallucinations* 2.4.2.

The comparison between the attribution methodologies of OpenAI and PerplexityAI reveals significant differences. PerplexityAI employs a detailed attribution system where each sentence or semantic assertion in an academic response is individually cited, enabling clear

tracing of each claim to its source. Conversely, OpenAI's method typically aggregates sources at the end of a paragraph or the entire response, potentially obscuring the direct correlation between specific claims and their sources, and leaving ambiguity in instances where multiple sources might support a single claim or whether certain claims are substantiated.

In terms of response length and content, GPT4's answers are generally more extensive, averaging 328 words, compared to the shorter responses from PerplexityAI, averaging 127 words. GPT4's responses are characterized by comprehensive elaboration and contextual explanations, whereas PerplexityAI appears to rely more on direct citations without further exposition. Intriguingly, despite PerplexityAI utilizing GPT3.5, a predecessor to GPT4, the disparity in response quality seems to stem more from the differences in the attribution systems rather than the capabilities of the underlying LLMs.
PerplexityAI provides a functionality to focus on or limit the sources to a specific domain, such as academic papers, news articles, or general web sources. This feature is not available in GPT4 without additional plugins. GPT uses general BingSearch to draw documents, where Perplexity AI most likely uses their own indexing, even though definite answer on which search method they use is not publicly available.
User testing reveals that PerplexityAI draws information from up to 5 different sources / websites, whereas GPT4 typically cites 3 sources.

### 2.5.3. Claims - Answer Segmentation

The initial phase of answer attribution involves breaking down the response into manageable segments requiring individual attribution. This process is vital for accurate attribution, as highlighted in the case study preceding this section. Lengthy paragraphs or responses often encompass various assertions, each potentially resting on different sources or requiring separate validation. Consider the instance illustrated in Figure 2.3, which underscores this point:

> **User Query:** "What are current methods to mitigate hallucination in large language models and which perform best? Please provide sources to your claims."
> **GPT4 Answer:** "[...] Another method is the use of iterative natural language feedback or self-reasoning, enabling LLMs to refine their initial outputs, thereby reducing hallucinations. Techniques like CoVe employ a chain of verification, where the LLM first drafts a response and then generates questions for self-fact-checking. DRESS aims at tuning LLMs to align with human preferences through natural language feedback, permitting non-expert users to critique model outputs. [...]"
> **GPT4 cited sources:** Tonmoy et al., (2024) [54], Mittal, Unite.AI, (2023) [60]

In this example, the entire answer is attributed to two sources, leaving the user uncertain about which portion of the response each source supports. More critically, the answer includes several distinct assertions, some of which may necessitate further verification, particularly if

the source is not entirely reliable or if the claim is presented without additional evidence.

Optimal segmentation of answers involves breaking them down into individual atomic claims, as delineated in 1.2. This level of segmentation is more detailed than mere sentence-level analysis, since a single sentence can encompass multiple assertions. Particularly, conjunctions and disjunctions within a sentence can lead to several claims, as each component of the list may represent an individual assertion. The majority of attribution research targets this level of segmentation due to its granularity and the comprehensive information it offers to users [8, 59, 61, 62]. Min et al., (2023) [8] argues that utilizing the most granular information segments aids in minimizing ambiguity and subjective interpretation. They define an "atomic claim" as a unit containing a single piece of information. Based on this definition and the preceding example, the following claims could be delineated:

1. "A method for mitigating hallucinations in LLMs involves leveraging iterative natural language feedback."
2. "Leveraging iterative natural language feedback allows LLMs to refine their initial outputs."
3. "CoVe uses a chain of verification approach to fact-check its own response."
4. ...

Note that the following example would **not** be an atomic claim, since it contains multiple pieces of information:

> "Techniques like CoVe use a chain of verification approach, where the LLM first drafts a response and then generates potential verification questions to fact-check its own response."

In practical settings, the level of segmentation often exceeds that of atomic claims, primarily due to constraints like the feasible number of sources for referencing, as observed in the preceding case study.
The methodologies for segmenting answers into atomic claims range from token-labeling techniques to approaches leveraging transformer models or generative methods. Chen et al., (2023) [61] describes this process as "propositional segmentation" and evaluates various transformer architectures for this purpose. The highest performing model in their study was T5-Large [63, 35], achieving a Jaccard $\theta = 0.8$ F1 score of 55.5, compared to the human benchmark of 67.1. Other research, such as Min et al., (2023) [8], employs a combination of human annotators and LLM-generated segments (InstructGPT) to create atomic claims. However, they do not provide specific performance metrics for these tasks. This research suggests that the mere question of "what needs to be attributed" is not trivial and requires further research on its own.

### 2.5.4. Finding Relevant Sources

In the context of answer attribution, the process of finding relevant sources is intrinsically linked to the principles of information retrieval (as described in Serction 2.6). This task, however, involves certain specific considerations, particularly when applied to answer attribution. A primary distinction emerges between **open domain** and **closed domain** attribution tasks. Open domain tasks are characterized by the absence of restrictions on the nature or origin of source documents. A notable example within this domain is the attribution of answers derived from LLMs to source documents available on the internet. In this scenario, search engines play a pivotal role in generating a comprehensive or selective list of potential sources in response to user queries.
Conversely, closed domain attribution involves specific constraints on the source documents, which could be limited to a particular database or a predefined collection of documents. The majority of research focuses on closed domain tasks due to their more manageable and quantifiable nature.

A further distinction is observed between PHR (Post-Hoc Retrieval) and RTR (Retrieval Then Response) systems, as described in 2.5.1. In PHR systems, the generation of answers is entirely decoupled from the retrieval process. Consequently, relevant sources might need to be identified separately for each assertion made.
In contrast, RTR systems involve an initial comprehensive retrieval of sources, with subsequent answers being formulated based on these pre-retrieved documents. Ideally, in an effective RAG (Retrieval-Augmented Generation) system, every assertion can be traced back to the initially gathered sources.

The challenge of source attribution and discovery is particularly pronounced in open domain PHR systems, which, given the focus of this thesis, represent the area of utmost interest.

### 2.5.5. Source Claim Inference

Attribution evaluation describes the task of evaluating the relation between a given answer of a model and the proposed source documents. This task can span over multiple dimensions and categories. The most prominent one is strongly connected to Natural Language Inference (NLI, Section 2.2) and related to the previously provided definition of factuality for large language models, as given in Section 2.4.1. The most commonly used relation types are *Entailment*, *Neutral*, and *Contradiction* [8]. Other approaches only differentiate between *Hallucinated* and *Not Hallucinated* categories, which makes sense especially in RAG systems where the answer is supposed to be built solely on the retrieved resources [61].

Other dimensions for attribution aim to expand the complex nature of attributing a claim to multiple use case-specific metrics, as in Malaviya et al., 2023 [59]. Here, claims and attributions are not solely evaluated based on the factuality of a claim in relation to the source document, but also on additional dimensions. These dimensions not only capture the quality

of the attribution, meaning the relation between the source document and claim, but also the quality of the claim itself or the relation between claim and answer. The following dimensions are proposed, including but not limited to:

1. *Claim Informativeness*: This describes the informativeness of the claim in relation to the user question to judge the relevance of specific sections of the answer to the user query.
2. *Reliability of Source Evidence*: This dimension captures the reliability of the source documents themselves.
3. *Worthiness of Claim Citation*: This dimension assesses whether certain claims need to be cited at all, or if they can be classified as common knowledge.

Malaviya et al. [59] evaluate the responses of large language models to domain-specific questions requiring expert knowledge and assess the answers across the above dimensions of different systems not automatically, but with actual domain experts. This approach is particularly interesting, as it captures the real-world use case of large language models in a professional setting.
It also highlights some of the core challenges of attribution evaluation, which include the subjectivity of the evaluation and the lack of a clear ground truth. While completely reasonable for the use case of expert evaluation, the proposed dimensions are challenging to assess using automatic methods of Natural Language Processing (NLP).

A naturally arising approach for evaluating the responses of LLMs across complex dimensions is, again, using the LLMs themselves, similar to the approach mentioned in Section 2.4.3 regarding the detectability of hallucinations. In this scenario, the enhanced reasoning capabilities of LLMs are utilized in combination with carefully constructed prompts (user queries) that define the task at hand to evaluate dimensions where direct ground truth training data is not available. For instance, the dimension of *Worthiness of Claim Citation* would be presented as a prompt to the LLM, asking it to assess the worthiness of a specific claim in a given context. For this particular dimension, it is reasonable to assume that the LLM can provide a reliable answer, as it is trained on a vast amount of data and can be expected to have a good understanding of what constitutes common knowledge. Multiple approaches, such as those by Chern et al. [64] and Wang et al. [65], utilize this assumption. The evaluation of the quality of this specific step is not provided, as it would need to be conducted by humans and therefore be expensive. The same assumption can be brought up for *Claim Informativeness*, as this task is closely related to NLI and solely solvable by natural language understanding. Other dimensions, such as *Reliability of Source Evidence* are more challenging to evaluate, as they require a deep understanding of the source documents and additional meta information about sources and publishers of reference documents.

However, this approach has notable limitations. Primarily, self-referencing and self-correction within LLMs do not inherently mitigate bias, which can lead to significant error propagation. The underlying training data of LLMs often contains biases, and relying on the same models

for evaluation can perpetuate these biases in the assessment process. This is particularly concerning in dimensions like *Worthiness of Claim Citation* and *Claim Informativeness*, where subjective judgments are involved. For example, an LLM's concept of 'common knowledge' may reflect the biases present in its training data, leading to skewed evaluations. Especially the problem of potential benchmark leakage to closed sourced models is made transparent in current research [66].

## 2.6. Information Retrieval

Information Retrieval (IR) is a fundamental aspect of computer science, focusing on acquiring relevant information from a large repository in response to a specific query. The evolution of IR has closely paralleled advancements in technology and the changing nature of information needs and requests. This field is expansive and diverse, covering various subfields including document retrieval, web search, structured data retrieval, and multimedia retrieval. The following section concentrates on information retrieval within the context of **natural language processing**. It excludes query languages like SQL or information retrieval on structured knowledge bases such as knowledge graphs.
The core elements of information retrieval include the user query $q$ (analogous to the problem statement definition in 1.2), the reference corpus $C$, and the scoring function $f$. The objective is to retrieve a set of documents $D$ from $C$ that are relevant to the query $q$. A document $d$'s relevance to a query $q$ is typically determined by a scoring function $f(q, d)$, which assigns a score to each document in $C$. The documents are then ranked based on their scores, and the top $k$ documents are returned as the result. The scoring function $f(q, d)$ may be influenced by various factors, such as the frequency of query terms in the document, the position of query terms, or the query and document's similarity. The most widely used scoring function is the *TF-IDF* (Term Frequency-Inverse Document Frequency), which considers the frequency of query terms in the document and across the entire corpus. TF-IDF is an example of a vector space model [67], which represents documents and queries as vectors in a high-dimensional space. The cosine similarity between the query and document vectors is then used to determine the relevance of the document to the query [68].

### 2.6.1. Web Information Retrieval

The most common and well known use case for a system of information retrieval is searching for specific information from the web. Web Information Retrieval (Web IR) differs significantly from traditional information retrieval due to the unique characteristics and complexities of the web as a dynamic and vast information repository [69].

The primary challenges in Web IR stem from the sheer scale, heterogeneity, and dynamic nature of web content. Unlike structured databases, the web comprises a vast array of unstructured or semi-structured data, including text, images, videos, and interactive content. This diversity necessitates sophisticated processing techniques for effective retrieval. Additionally,

the web's dynamic nature, with constant additions, updates, and deletions, poses a challenge for maintaining up-to-date and relevant search results. Another critical challenge is the issue of relevance and ranking. Given the enormous volume of available data, determining the relevance of web pages to a user query is a non-trivial task. Search engines must employ complex algorithms to rank pages not only by their content's relevance but also by their authority, quality, and user engagement metrics.

Search engines are the primary means of web information retrieval. They employ web crawlers to traverse the web and index pages, which are then processed and stored in databases. When a user submits a query, the search engine retrieves and ranks the results based on its indexing and ranking algorithms [70].
The effectiveness of a search engine in Web IR is largely determined by its indexing comprehensiveness and the sophistication of its ranking algorithms. These algorithms often involve a combination of keyword matching, link analysis (such as PageRank), and personalization factors [69].

Indexing the web presents several challenges, primarily due to its size and rate of change. Efficiently indexing billions of web pages and keeping the index current with frequently updated content is a significant technical challenge. Additionally, the index must be structured in a way that allows for rapid retrieval of relevant pages, which is essential for the performance of a search engine.

The web, and search engines in particular, being the most prominent and widely used sources of information today, have significantly shaped the way we access and process information. Common search engines, such as Google, Bing, and Yahoo, typically do not provide direct answers in the form of natural language to user queries. Instead, they present a list of web pages that may contain the sought-after information. It is important to note, however, that there are approaches to directly answer user queries, such as the 'featured snippets' in Google or in Bing search results. But these snippets, in most cases, contain only a single atomic piece of information, such as a definition or a fact, and are not suitable for the task of attribution.
The manner and language in which users formulate their queries are adapted to the capabilities of search engines. Most search engine users do not pose complex questions, but rather, they reduce their queries to keywords and short phrases [71, 41]. This is a significant limitation, as it restricts the complexity and depth of information that can be retrieved, while in addition, making most user queries ambiguous and more general.

### 2.6.2. User Goal Classification

Classifying the user goal, or more precisely, the information need of a user query, is a crucial aspect of information retrieval since the information need is highly use-case specific and can vary significantly. This categorization aims to identify the type of query or question posed by the user in terms of language style, intent, formulation, and expected answer type. Such an

abstraction can also be referred to as 'intent recognition', which is related to the Mixture of Experts approach as described in 2.1.4. A classification like this is particularly interesting for the task of answer attribution as it defines the type of answer the user is requesting. In the context of web searches, a user goal classification was proposed by Rose et al., 2004 [71] as follows:

1. **Navigational Queries**: These queries are aimed at finding a specific web page or website which is already known to the user. These queries are typically short and concise and easier as to type in the URL directly.
2. **Informational Queries**: These queries are aimed at finding information on a specific topic. Informational queries are the focus of this thesis and the most common type of user goal. Subcategories of informational queries include:
   a) **Directed Queries**: The user has a specific topic in mind and is looking for something particular about that topic.
      i. **Closed**: The user is looking for a specific and unambiguous piece of information to a question. An example would be "What is the capital of France?".
      ii. **Open**: The users goal is to get an answer to a query with unconstrained depth, in an open ended setting. An example would be "Tell me about the history of the internet".
   b) **Undirected Queries**: The user wants to learn anything about a specific topic without specifying a particular aspect. An example would be "Tell me about the history of the internet".
   c) **Advice**: Queries where the user is looking for advice, ideas, suggestions or instructions. An example would be "Help quitting smoking".
   d) **Locate**: Queries where the user is looking for a specific place or location. An example would be "Find the nearest gas station".
   e) **List**: Queries where the user is looking for a list of plausible items. An example would be "List of universities in Amsterdam".
3. **Resource Queries**: The user goal of these queries is to obtain a resource which is available on the web. This can be a file, a document, a video or an image.

The sections about "Navigational Queries" and "Resources Queries" are summarized to shorter versions since the focus of this thesis lies heavily on information need.
Malaviya et al., 2023 [59] propose a framework to answer expert questions with LLMs and also classify the user queries into different categories. They build on the work of Rose et al., 2004 [71] and propose a differentiation of the informational query categorization as follows:

1. **Directed Questions**: Questions that have a single and unambiguous answer.
2. **Open Ended Questions**: Questions that are open ended and are potentially ambiguous.

3. **Summarization**: Summarization of information on a topic.
4. **Advice**: Questions that are asking for advice or suggestions on how to approach a problem.
5. **Hypothetical**: Questions that describe a hypothetical scenario and ask a question based on that scenario.
6. **List of Resources**: Questions that ask for a list of resources on a specific topic.
7. **Opinion**: Questions that ask for an opinion on a specific topic.

Note that, although there is clear overlap between the two categorizations, their goals and use cases differ significantly. The categorization proposed by Malaviya et al., 2023 [59] is specifically designed for expert questions and aims to categorize a **question** independently. In contrast, the categorization by Rose et al., 2004 [71] focuses on categorizing the **user goal** of a query within the context of web searches. The implications and differences between these categorizations are discussed in the section about classifying user needs for LLMs 3.1.

### 2.6.3. Information Retrieval Systems

Information retrieval systems form the foundation of information retrieval, encompassing the algorithms and methods used to respond to user queries. These systems boast a significant history, extending beyond mere technological advancements, and have become increasingly vital as data and information grow more accessible and abundant. Early information retrieval systems enabled users to articulate their information needs using combinations of Boolean logic and terms. These were then matched against a database of documents [72]. Such systems offered complete user control but lacked in aspects like user-friendliness and handling ambiguous queries. There are three primary categories of modern information retrieval systems: **Vector Space**, **Probabilistic**, and **Network Inference Models** [72].

**Vector Space Models** represent both documents and queries as vectors in a multidimensional space. The relevance of a document to a query is determined by the cosine similarity between their vectors. This model is advantageous for its simplicity and effectiveness in capturing the importance of terms within documents.
**Probabilistic Models**, on the other hand, are based on the principle of probabilistic relevance. They estimate the probability that a given document is relevant to a user's query. This approach incorporates uncertainty and variability in information retrieval, providing a more dynamic response to user queries.
Lastly, **Network Inference Models** utilize graph-based structures to represent relationships between documents and queries. Nodes in these networks represent documents or terms, while edges signify the relationships or associations between them. These models are particularly effective in leveraging the interconnectivity and contextual relationships within the information.

Each of these models offers distinct advantages and operates on different principles, reflecting the diversity and complexity inherent in information retrieval systems [72, 73].

### 2.6.4. Vector Databases

Vector databases, or vector database management systems (VDBMS), are Information Retrieval (IR) systems that fall into the category of *Vector Space Models*. They are used to store and retrieve vectors based on similarity searches between the query vector and the stored vectors. Specifically designed to handle high-dimensional vector data and scalability, VDBMS can be tuned for specific use cases [74]. VDBMS experienced a rise in popularity analogous to that of artificial intelligence solutions, as vector-based representations of specific input structures are a naturally occurring byproduct of neural network applications.
Particularly, data structures that are difficult to describe, lack meta-information, or are inherently high-dimensional, such as images, audio, or text, are well-suited for vector databases. In this thesis, we focus on VDBMS for text data, as the task of information retrieval and attribution is inherently text-based.

The components of a text-based VDBMS are listed as follows:

1. **Embedding Engine**: An engine that processes raw text data and converts it into high-dimensional vector representations. This can be achieved using traditional vector representations, such as Bag-of-Words (BoW), specialized vector representations like Word2Vec or Doc2Vec, or modern transformer-based models, such as BERT or GPT.

2. **Chunking Engine**: The system that divides the *storage input text or document data* into smaller, meaningful segments, like sentences or paragraphs. This is crucial for attribution, enabling the system to attribute specific claims to their respective sources accurately.

3. **Indexing Engine**: The engine responsible for indexing the vector representations of input data. This enhances the efficiency of data storage and retrieval. Common indexing systems are categorized as *Table Based*, *Tree Based*, or *Graph Based* [74], each differing in their time and space complexity.

4. **Storage**: The component that stores the indexed vector representations. In large-scale systems, this usually involves a distributed storage system, like Hadoop [75].

5. **Similarity Search Algorithm**: The algorithm employed for identifying the vectors most similar to a given query vector. This is typically a nearest neighbor search algorithm, such as k-Nearest Neighbors (k-NN) or Locality Sensitive Hashing (LSH) [74].

There exists a multitude of vector databases, each with its unique features and capabilities. Some of the most prominent vector databases include Faiss [76], ChromaDB [77], and Milvus [78]. These databases are designed to handle high-dimensional vector data and are optimized for similarity searches, making them particularly well-suited for tasks of information retrieval

and attribution. The choice of vector database highly depends on the specific use case and the nature of the input data. For instance, Faiss is particularly well-suited for large-scale, high-dimensional data, whereas ChromaDB is designed for multimedia data, such as images and audio. On the other hand, Milvus is a general-purpose vector database that is highly scalable and optimized for similarity searches. The most referenced metric to benchmark is not retrieval quality, but the execution and retrieval time for large scale systems. An open source benchmarking tool for VDBMS is *VectorDBBench* [79] where a version of Milvus ranks the highest in terms of execution time and retrieval time for large scale systems.
The ANN-Benchmark [80] provides an overview of the performance of different approximate nearest neighbor search (ANN) algorithms, which are crucial for similarity searches in VDBMS. The most prominent algorithms are HNSW [81], IVFADC [82], and ANNOY [83]. The benchmarking tool provides a comprehensive overview of the performance of these algorithms on different datasets, embedding types and distance metrics and is a valuable resource for selecting the most suitable algorithm for a specific use case.

# 3. Main Part

The primary objective of this thesis is to explore the issue of complex answer attribution by delivering structured insights into the predefined research questions. This section, the main part, adopts a more informative and analytical approach to the provided research question, whereas an in-depth discussion on the implications of the findings is presented in chapter 4, titled "Discussion".

Each subsection within this chapter focuses on a distinct research question and includes a detailed analysis of the issue as well as a methodological strategy for addressing it. At the outset of each subsection, a summary of the findings is presented. References to definitions and related work, covered in the preceding chapter, are extensively utilized in this chapter.

## 3.1. Building a Taxonomy and Dataset

The goal of this sections is to provide a structured and analytical approach for answering the research question 1, as stated in Section 1.2.1:

**RQ1: Classifying information needs in LLM based Q&A Tasks**
1.1 Can we identify more complex but frequently asked questions and categorize them domain-independently?
1.2 Is it possible to construct a dataset with quadruplets of (category, question, answer, attribution / source) based on previously established categorizations.
**Deliverables:** A taxonomy for question categories for user interactions with large language models.
A formal structure for a dataset which allows for the comparison of different attribution approaches. This stucture aligns alongside current research and unifies different approaches.

This research question is motivated mainly by the changing nature of accessing information in the context of LLMs and the lack of a comprehensive taxonomy for classifying information needs.

### 3.1.1. Methodology

The primary objective of this section is to provide a quantified response to the assertion regarding the evolving nature of interaction with information in the context of LLMs. This goal is accomplished by examining existing Question and Answer (Q&A) datasets and

observing how the formulation of questions and expectations for answers have changed over time.
Drawing from the insights of this analysis, we review existing taxonomies for categorizing information needs and, if necessary, propose a new taxonomy. A baseline for understanding information needs has already been established in the related work (Section 2.6.2).
By integrating this newly proposed taxonomy with the findings from the first part, a novel dataset is constructed. This dataset is then utilized to assess the efficacy of current LLMs in addressing various information needs.

### 3.1.2. Analysis of Existing Datasets

In multiple previous sections of this thesis, the assertion was made that the manner in which we interact with information is undergoing significant changes, particularly through large language models. To support and attribute this assertion, we analyze established Q&A datasets and observe how the formulation of questions and expectations for answers have evolved over time.
For an overview of existing datasets, we utilized this list of Q&A datasets [84] and selected the following datasets for analysis:
These datasets were chosen to represent a variety of open domain Q&A dataset types,

| Dataset | Year | # Questions | Answer Type | Context | Evidence |
|---|---|---|---|---|---|
| WebQuestions [85] | 2013 | 5,810 | Entities | No | No |
| MSMarco [86] | 2016 | ca. 1,000,000 | Human generated | Yes | No |
| SQuAD [40] | 2016/2018 | 107,785 | Span of words | Yes | Yes |
| TriviaQA [87] | 2017 | ca. 95,000 | Single Entities | No | Yes |
| Natural Questions [41] | 2019 | 307,373 | Entities & Paragraphs | Yes | No |
| ExpertQA [59] | 2023 | 2,177 | Full Paragraphs | Yes | Yes |

Table 3.1.: High level comparison of selected Open Domain Q&A datasets

question types, and answer types comprehensively. A brief description of the characteristics of each dataset, including an example, is provided in the following list:

1. WebQuestions: Questions generated using the "Google Suggest API", all starting with a wh-word and designed to be answerable by a single entity.
   *Question*: "What is the political system of Nigeria?"
   *Answer*: "Federal republic"

2. MSMarco: A dataset of real user queries from Bing and Cortana search engines, where for each query, text passages from the retrieved webpages are provided to human annotators to generate answers.
   *Question*: "What is a corporation?"
   *Answer*: "A corporation is a company or group of people authorized to act as a single entity and recognized as such in law."

3. SQuAD: A dataset of human-generated (crowdworkers) questions based on Wikipedia articles, where the answer is a span of words from the article.
   *Question*: "What is essential for the successful execution of a project?"
   *Answer*: "Effective planning" *Context*: "[...] For the successful execution of a project, effective planning is essential. [...]"

4. TriviaQA: A dataset of questions generated from trivia games, where the answer is a single entity.
   *Question*: "Who was the man behind The Chipmunks?"
   *Answer*: "David Seville"

5. Natural Questions: A dataset of questions generated from real user queries, where the answer is a single entity or a paragraph from a Wikipedia article.
   *Question*: "When are hops added to the brewing process?"
   *Short Answer*: "During the boiling process" *Long Answer*: "After mashing, the beer wort is boiled with hops (and other flavorings if used) in a large tank known as a 'copper' or brew kettle – though historically [...]"

6. ExpertQA: A dataset of complex questions generated by experts in different domains, where the ground truth answer is a complete paragraph provided by an LLM and revised by the expert who asked the question. This dataset is built to evaluate LLMs.
   *Question*: "What is the sunny 16 rule and how do I apply it with exposure compensation?"
   *Answer*: "In a studio setup where a photo of a model is overexposed with the light at 45 degrees and two strobes, you can decrease the [...]"

An initial qualifying analysis of these datasets reveals significant differences between previous datasets and ExpertQA [59], which focuses on answers from LLMs. The majority of datasets are designed to be answered by a single entity or a span of words, extracted from a context document. Most are centered around reading comprehension tasks, where the context, which should contain the answer, is provided. This creates a similarity to RAG and RTR-Systems, as described in Section 2.5.1. Additionally, the sources for questions and answers are predominantly web-based, such as Bing, Cortana, or Wikipedia. This influences the language style of questions, as evident in the examples from Natural Questions and SQuAD.
In contrast, ExpertQA questions are phrased in a more complex manner, often including follow-up questions. This prevents simple entity extraction and necessitates a more intricate understanding of the question and the domain, including reasoning and inference.

To quantify these differences, we compare the datasets in terms of average question length, average answer length of the answers provided in the ground truth, and average context length if applicable. The length is measured in the average number of characters. If there are pre-defined splits of datasets, we only use "train". Furthermore, we have a random sample of 100 questions per dataset answered by GPT-3.5 (Microsoft Azure API version) [6], without specifying the desired answer length in the model's prompt. This aims to highlight the divergence between LLM answers and traditional Q&A system answers. The results of this

analysis are presented in Table 3.2.

If a question in a dataset lacks an answer, it is labeled with "NaN" (Not a Number) and

| **Dataset** | ∅ len($Q$) | ∅ len($A_1$) | ∅ len($A_2$) | ∅len($A_{GPT3.5}$) |
|---|---|---|---|---|
| WebQuestions [85] | 37.2 | 44.7 | - | 234.9 |
| MSMarco [86] | 35.2 | 78.0 | 81.0 | 570.1 |
| SQuAD [40] | 58.5 | 20.1 | - | 337.6 |
| TriviaQA [87] | 79.3 | 10.4 | 14.5 | 120.1 |
| NaturalQuestions [41] | 48.7 | 25.4 | 1375.0 | 414.3 |
| ExpertQA [59] | 112.6 | 982.3 | 1084.8 | 1367.0 |

Table 3.2.: Low level comparison of selected Open Domain Q&A datasets based on average length of characters of questions and answers. $A_1$ and $A_2$ are the ground truth answers, $A_{GPT3.5}$ is the answer from GPT-3.5.

excluded from the average calculation. The distinction between answer $A_1$ and $A_2$ is made only if the dataset contains two structurally different answer systems or if the dataset's design inherently aims for two different types of answers to the same question. An example of this is the Natural Questions dataset, which aims to include both a short answer and a long answer for each question. In cases where a single question has multiple answers from the same system within a dataset, only the first answer is considered. For Natural Questions, the short answers are designated as $A_1$ and the long answers as $A_2$. In the case of MSMarco, the first answer is categorized as $A_1$ and the "well-formed" answer as $A_2$. For TriviaQA, $A_2$ represents the first normalized alias of the ground truth answer, $A_1$. In ExpertQA, $A_1$ is the revised answer the "Retrieve then Read Google Search GPT4" system, while $A_2$ is the "Post Hoc Google Search GPT4" systems answer. Note that not every question has a revised answer of those two systems, but for the sake of comparability, only those systems where used. Additionally, the answer prompt of ExpertQA asks the model to limit the answer to 500 words.

Every filler or padding character (e. g. from HTTP) is removed from the answers such that they remain as close to written natural language as possible. Spaces, commas, full stops and other punctuation marks are included in the character count. The same applies to the questions.

The statistics in Table 3.2 are visualized in Figure 3.1 for enhanced comparative analysis. Several key aspects of these statistics merit attention. The first notable point is the considerable difference in the average character length of questions and answers between ExpertQA, which targets complex queries for LLMs, and other datasets. ExpertQA displays the longest average question length, as well as the longest average answer length for $A_1$. The only dataset with longer answers than ExpertQA is NaturalQuestion, specifically the $A_2$ answers. A critical distinction here is that the $A_2$ answers from NaturalQuestion do not directly address the question but rather present a complete paragraph from the related

Wikipedia article that includes the "short answer".

Another significant difference is observed in the average answer length of GPT-3.5 compared to the ground truth answers across all datasets. GPT-3.5's average answer length is markedly longer than that of the ground truth in every dataset. This disparity is especially pronounced in the WebQuestions and MSMarco datasets, where GPT-3.5's average answer length exceeds that of the ground truth by more than fivefold. This discrepancy underscores the distinct nature of responses generated by LLMs in contrast to those from conventional Q&A systems. In the case of complex queries (ExpertQA), GPT-3.5, when not restricted by a specified answer length in the prompt, tends to provide answers comparable in length to complete Wikipedia paragraphs, akin to the Natural Questions $A_2$ responses. Because

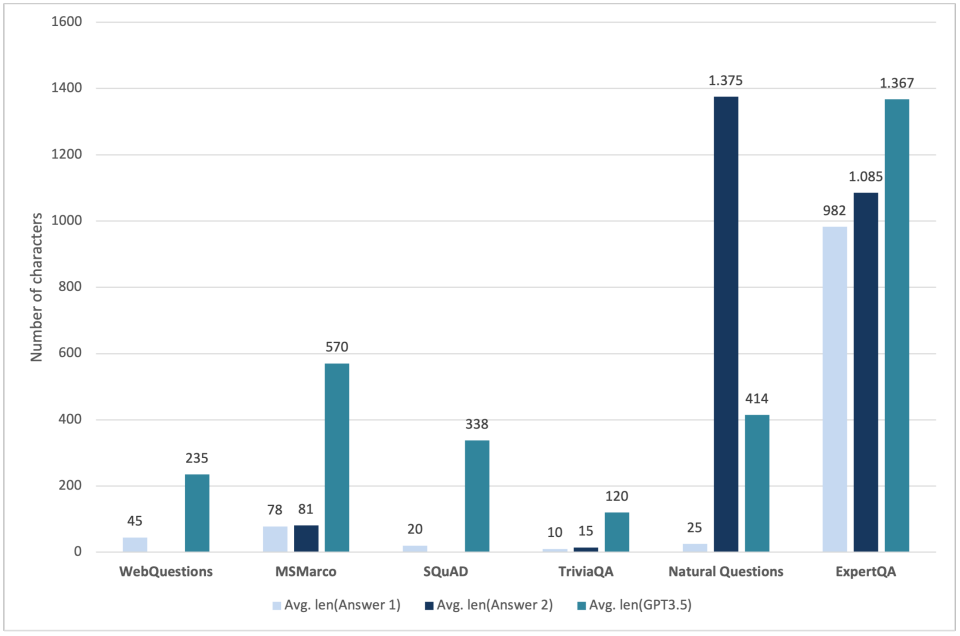


Figure 3.1.: Visual comparison of average character length answers in selected Q&A datasets.

analyzing answers and questions based solely on character length is insufficient to capture their complexity, we also analyze the embedding space of 250 randomly sampled questions and answers ($A_1$) from each dataset. For this purpose, we use OpenAI's GPT-3.5 (Microsoft Azure API version) "Ada-002" model as an embedding instance. The embeddings of each question and answer are reduced to two dimensions using Principal Component Analysis (PCA) from the Scikit-Learn library [88], and visualized in a scatter plot. The color of the points indicates the dataset, and the distance between points represents the similarity of the embeddings. The results of this analysis is presented in Figure 3.2.

It is evident that for both embeddings, questions and answers, the ExpertQA dataset stands out and is less intersected with other datasets. In the answer embedding space, the most distinct group of outliers belongs to the WebQuestions embeddings. These differences

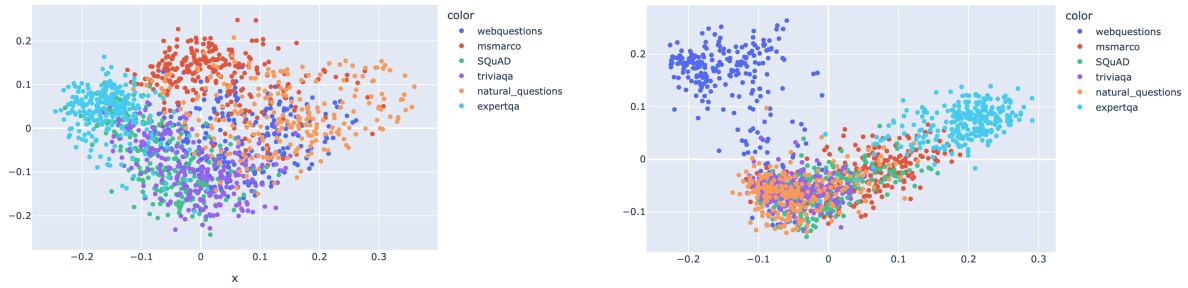


Figure 3.2.: Embedding PCA visualization of 250 random questions (left) and corresponding answers ($A_1$, right) from selected Q&A datasets.

are also visible in the silhoutte score of the respective clusters, which is put in numbers in Table 3.3. The silhouette score is a metric used to calculate the goodness of a clustering technique, defined as the mean distance between a sample and all other points in the same class (cohesion) and the mean distance between a sample and all other points in the next nearest cluster (separation). The scores ranges from -1 for incorrect clustering to +1 for highly dense clustering. For each category, the silhouette score for ExpertQA is the highest, indicating the largest distinction from other Q&A datasets.
It is important to note that the original embedding has a dimensionality of over 1500 dimensions, which is substantially reduced by PCA, leading to a loss of information.
The results of the PCA analysis align with the findings from the character length analysis. The ExpertQA dataset is distinctly separable from the other datasets, especially in the answer embedding space. This distinction reflects the complexity and unique nature of the questions and answers in ExpertQA, where the focus is on posing questions to LLMs. We contend that this is representative of the new opportunities LLMs provide for interacting with information, while also delivering valuable responses to traditional, entity-focused queries.

| Dataset | Questions | Ground Truth Answers | GPT3.5 Answers |
|---|---|---|---|
| WebQuestions | -0.076357 | 0.450837 | -0.143352 |
| MSMarco | 0.204624 | -0.110372 | -0.194305 |
| SQuAD | -0.143934 | -0.178242 | -0.212656 |
| TriviaQA | 0.039180 | 0.141169 | 0.026112 |
| Natural_questions | -0.101483 | -0.201564 | -0.151824 |
| ExpertQA | **0.398834** | **0.572056** | **0.299889** |

Table 3.3.: Silhouette scores of the clusters in PCA embedding for Questions, Ground Truth Answers and GPT3.5 Answers as in Figure 3.2 and Figure 3.2.

### 3.1.3. Analysis of Existing Taxonomies

In addition to the quantitative analyses of the previous section, we suggest a more qualitative and example-based examination of existing taxonomies and question classifications. This analysis extends the discussion in Section 2.6.2, building upon the question classifications by Malaviya et al. [59] and the user goal classifications by Rose et al. [71].
The distinction between question classification and user goal classification is crucial, leading to the first observation: It is imperative to specify what exactly is being classified. In addition to user goal and question classification, answers can also be categorized, as demonstrated by Rajpurkar et al. [40]. Here, the extracted word spans are classified by the type of information they contain, such as *Date*, *Numeric*, *Person*, *Entity*, etc. Since in this publication, the user goal of every question is limited to **Closed, Directed, and Informational**, as described by Rose et al. [71], this classification is exhaustive for the specific dataset but not for the general case.

Other classification criteria of question types strictly focus of the formulation of the question itself and the keywords it contains, such as "Who", "What", "Where", "When", "Why", "How", etc. WebQuestions [85] is an example of a dataset that is based on this classification, whereas here all questions start with a wh-word. This classification is useful for understanding the structure of the question, but it does not provide any information about the user's information need.

In order to analyze the pros and cons of existing taxonomies, we first need to formulate the specific goal and use case of a taxonomy or classification. In the context of this thesis, our goal for developing a Q&A taxonomy is as follows:

> **Goal:** The taxonomy aims to provide a valuable distinction between different information needs of users in complex Q&A settings, impacting the quality of answers provided by LLMs and, consequently, the requirements for answer attribution.

For instance, a question such as "What is the capital of Germany?" can be answered through simple entity extraction and has significantly different attribution requirements compared to a query like "Explain the political system of Germany in detail." The former can be addressed through straightforward entity extraction or as an atomic claim, whereas the latter requires a more complex understanding of both the question and the domain, including reasoning and inference.

Taking this goal into consideration, we analyze existing taxonomies and classifications, referencing specific queries from ExpertQA. These examples are selected to represent the complexity and diversity of questions in ExpertQA and those posed to LLMs:

1. "A patient with a history of heart failure now presents with newly diagnosed metastatic HER2+ breast cancer. What is her recommended first line of treatment and what additional information should be discussed with the patient given her history of heart failure?"

2. "How will the estate of an individual who dies without a will be distributed?"
3. "Explain the concept of the Rainbow nation, referring to the event of human evolution and genetics studies."

It becomes clear that classifying questions based solely on the words they contain is not feasible in the context of LLMs and queries like those in ExpertQA. Firstly, questions can contain multiple keywords, as demonstrated in example (1), and secondly, these queries might not be formulated as traditional questions, as example (3) demonstrates.

A natural extension of this is the classification of questions based on a more **abstract grouping**, such as the categorization proposed in ExpertQA [59], which is described in Section 2.6.2. In a taxonomy like this, questions are grouped together if they appear to ask for the same type of answer or are structured similarly. For example, in this categorization, questions (2) and (3) could be grouped together, as they both seek an explanation of a concept. However, this example also highlights the limitations of a concise taxonomy. Question (2) could be considered to be answerable by simple, entity-like answer, and therefore fall under the question type of *Directed Questions*. But the complexity of the domain prevents a straightforward response, making the answer ambiguous and the question therefore *Open Ended*. This indicates that a taxonomy should consider not only the question itself but also the context in which it is asked, and inherently, that the question categorization system requires knowledge about the domain in order to correctly classify the question.

The second dimension in this taxonomy pertains to the structure of the posed question, a category novel to other Q&A tasks. Malaviya et al. [59] instruct annotators to create questions *"that describe a hypothetical scenario and ask a question based on this scenario"*. Question (1) exemplifies this category. This category presents several intriguing implications. Firstly, no other Q&A setting or dataset accommodates this question type, as it introduces significant complexity, particularly for approaches focused on entities or specific word spans. Secondly, this category does not address the user's goal or information need directly, as the other question types of ExpertQA do. It represents a purely structural classification, which contrasts with user-goal-oriented classifications such as Closed or Directed. Malaviya et al. [59] address the conceptual inconsistency between user goals and question structure by allowing multiple categories for a single question. While the rationale for this specific classification is understandable, given that these types of questions are common in interactions with LLMs and were not answerable by automatic systems previously, it could also lead to an excessive number of categories when attempting to build a comprehensive, mutually exclusive and collectively exhaustive (MECE) taxonomy.

The third possible dimension for classifying questions is the user goal behind the question. While arguably the most abstract and valuable for a Q&A system, as it would provide clear indication if the response actually satisfies the query, it is also the most complex to construct and implement. The key challenge in an approach like this is dealing with language ambiguity and the diversity of user goals. When classifying the user goal, one implicitly

classifies the expected answer. Some categories of Malaviya et al. [59] are based on the user goal, such as *Directed Questions*, *Open Ended Questions*, or *Request for Resources*. Classifying the user goal based on a question requires analyzing what type of information and in what structure the user's informational need would be fulfilled. Taken this into account, the task of classifying the user goal behind a question can be reformulated as follows:

> **Task:** Classifying the user goal in a question or query involves determining the specific structure and depth of information required in the answer to adequately meet the user's informational needs.

This precise formulation leads to the fourth possible dimension of a taxonomy, which is the classification of the answer types. Key challenge for this dimension is finding an adequate abstraction level to cover the diversity of possible answers while not letting the taxonomy become too complex. For example, we argue that all directed questions that are answerable by a single and unambiguous answer should be summarized as *Closed Directed Questions*. Other Q&A dataset which focus on entity extraction differentiate questions in this category further by describing the entity type of the answer, like *Date*, *Numeric*, *Person*, *Entity*, etc. This can be a valuable distinction for some datasets, but not for a high level taxonomy.

The above points can be summed up in the following observations:

1. Existing taxonomies and classifications are not designed to address the complexity and diversity of questions posed to LLMs, particularly in the context of ExpertQA.
2. The classification of questions based solely on the words they contain is not feasible in the context of LLMs and queries like those in ExpertQA.
3. In complex domains, a question categorization system requires knowledge about the domain and the answer of the question in order to correctly classify the question.
4. A taxonomy should differentiate between the structure of a question and the user goal behind a question.
5. Classifying the user goal means describing what informational structure is needed to fulfill the user's informational need.

### 3.1.4. New Taxonomy

Based on the previously described findings, we propose a new taxonomy for **classifying information needs (user goals)** in the context of Large Language Models. This taxonomy aims to address the complexity and diversity of questions presented to LLMs, especially along the example of ExpertQA. Basing on Malaviya et al. [59] and Rose et al. [71], our taxonomy extends their classification systems by introducing more detailed categories. It distinguishes between high-level user goals and the more granular categories within these

goals. The defined taxonomy is as follows:

**Directed Question**

- **1.1 Factual / Atomar Information**: Users requesting verifiable information.
  Example: Who wrote the play "Romeo and Juliet"?
- **1.2 Definition**: Seeking a precise meaning of a term or phrase.
  Example: What is photosynthesis?

**Open Ended**

- **2.1 Elaboration**: Elaboration on more complex topics.
  Example: How does Machine Learning work?
- **2.2 Comparison**: Comparison of two or more different concepts or sources.
  Example: How do reptiles differ from amphibians?
- **2.3 Cause and Effect / Explanation**: Explaining logical reasoning or causal chains.
  Example: What led to the fall of the Roman Empire?

**Summarization**

- **3.1 Brief Overview**: Users seeking a concise explanation of a broad topic.
  Example: Can you summarize the events of World War II?
- **3.2 Complex Definition**: Defining longer and more complex concepts.
  Example: What is the pressure and release model?

**Advice / Suggestion**

- **4.1 Methodology**: Users seeking methods to tackle a problem.
  Example: How should I start when I want to learn programming?
- **4.2 Resource Recommendation**: Seeking resources related to a specific topic.
  Example: What are the best fantasy books in recent years?
- **4.3 Strategy / What to Do / Procedure**: Inquiring about step-by-step plans or procedures.
  Example: How do I exchange a car engine?

**Opinion**

- **5.1 Evaluation**: Judgment/Assessment.
  Example: What do you think about AI's impact on job markets?

- **5.2 Preference**: Requesting an evaluation between options.
  Example: Between deep learning and classical ML, which is better for small datasets?
- **5.3 Prediction / Consequence Analysis**: User requesting a prediction for a certain scenario.
  Example: If the sun disappeared, what would be the immediate effects on earth?

In this taxonomy, a user goal can be classified into multiple low-level categories. For example, the user need behind question (2) from Section 2.6.2, *"How will the estate of an individual who dies without a will be distributed?"*, would be classified as *Factual / Atomic Information* if applicable, and as *Elaboration*. This taxonomy is designed to be exhaustive yet not mutually exclusive, meaning that every question can be classified into one or more low-level categories, and that no question can be classified into more than one category of the same high-level category.

In addition to classifying user needs, we propose a taxonomy for classifying the structure of the question. This taxonomy does not aim to classify all possible question formulations in detail, but focuses on question structures novel to Q&A systems due to LLMs. All other question structures are classified as "Other Questions".

**Question Structure**

- **Hypothetical Set-Up**: Questions that formulate a hypothetical setting and ask a question based on this setting.
  Example: A patient with Ewing sarcoma of the femur is due for restaging scans while on VDC/IE after undergoing surgery which included the placement of hardware. Should the clinician order a CT, MRI, PET/CT, or a combination of these scans?
- **Follow-Up / Multiple Questions**: Queries that may contain co-dependent follow-up questions or simply contain multiple chained questions.
  Example: How does PCA differ from LDA, and when should I use one over the other?
- **Other Questions**: Questions that do not fall into the above categories.
  Example: What is the capital of Germany?

This duality of taxonomies, one for classifying the user need and one for classifying the structure of the question, is designed to provide a comprehensive and detailed understanding of information needs in regards to LLMs. It deals with the previous inconsistencies and lack of detail of other taxonomies.

### 3.1.5. Evaluation of the Taxonomy

To evaluate this taxonomy, we randomly sample 70 questions from ExpertQA and 30 questions from the Natural Questions [41] dataset. This list is given to annotators who are asked to

classify the user goal of each question while allowing multiclass classifications. In addition, the questions are categorized by GPT4 with a few-shot example prompt that also contains the above description of the taxonomies. The taxonomy is evaluated using the Cohen Kappa Score on a binary classification of each low-level category. That means that for each low level category $c$, there exist a list of length 100 of binary values per annotator $an$ that indicates if the annotator classified the user need of question $i$ as category $c$. The Cohen Kappa Score is a measure of inter-annotator agreement, which is adjusted for the possibility of the agreement occurring by chance. The range lies between -1 and 1, where 1 indicates complete agreement and 0 indicates that the agreement is not more than chance. Each annotator is compared to both other annotators, resulting in a total of 3 annotator comparisons with 12 Cohen Kappa scores each, one for every class. The results of this evaluation are presented in Figure 3.3 and Figure 3.4.

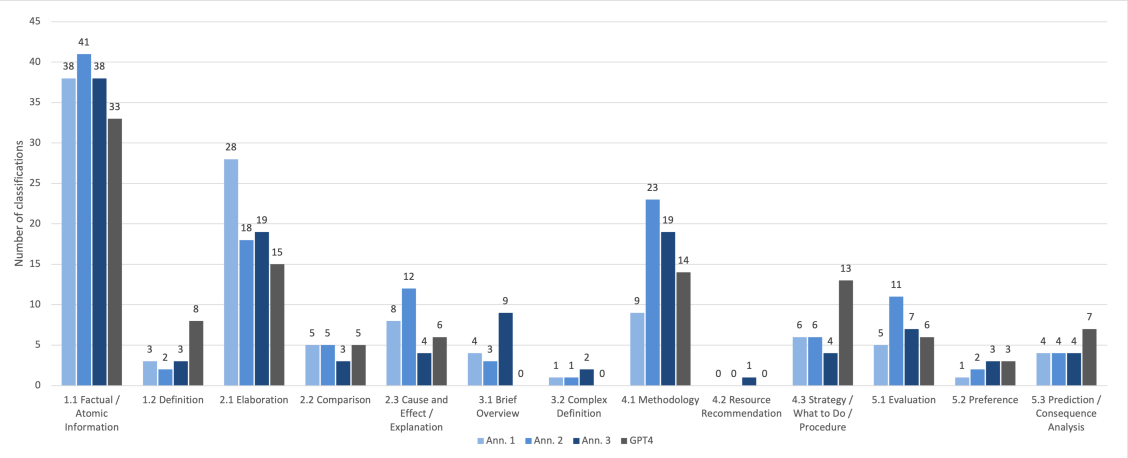


Figure 3.3.: Classification support for each annotator and GPT4 with multilabel classification

The evaluation results indicate that the taxonomy is generally well-supported, though significant ambiguity persists for some categories. Notably, GPT4 did not classify any of the 100 user queries under the high-level category **Summarization**, resulting in a lack of consensus between GPT4 and the annotators for these categories. In the category **Resource Recommendation**, only one annotator classified a user query, suggesting that this category may be either poorly defined or inherently absent in the sampled 100 questions. A similar argument can be made for the **Complex Definition** category, which had at most two supporting instances. The most support was observed in the class **Factual / Atomic Information**, likely due to the dataset containing 30 questions from Google's natural questions, which primarily seek atomic and factual responses.
The Cohen Kappa scores reveal that, on average, the agreement between annotators is higher than that between GPT4 and the annotators. However, the agreement for those categories actually supported by both GPT4 and the annotators is higher on average. This finding supports the assertion that the taxonomy is robust and that GPT4 is effective in classifying

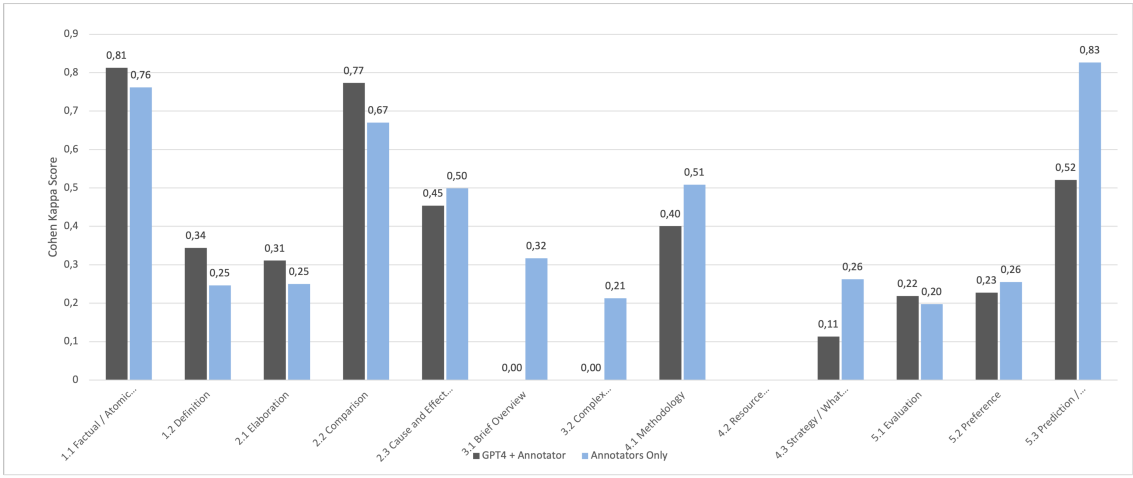


Figure 3.4.: Cohen Kappa Scores for each low level category of the taxonomy. Grey represents the average scores for GPT4 compared to the annotators, while blue shows the average inner-annotator scores.

the user need in a question.

To analyze which taxonomy classes are most interchangeable and similar, we constructed the following confusion matrix. It is important to note that due to the multi-label and multi-annotator classification, the depicted weights are not normalized and can exceed 1. The outcomes of this analysis are presented in Figure 3.5.
It is important to note that due to the methodology used in creating this confusion matrix, multi-label classifications are also marked as "confusion," although they do not necessarily indicate disagreement. Based on this confusion matrix, the following categories appear to need additional clarification and may be the most interchangeable:

- **Definition** and **Complex Definition**
  Explanation: The ambiguity in this category can be directly attributed to the domain understanding of the annotators, as noted above. Additionally, this distinction has become obsolete, as it can be depicted by the multi-label classification of "Elaboration" and "Definition" as well.

- **Elaboration** and **Brief Overview**
  Explanation: This confusion arises because both answer types generally satisfy most user needs similarly, as they both provide a more detailed explanation of a topic. Although the answer types are distinctly different, this distinction may not be apparent from the question itself. This issue can be addressed by instructing annotators to classify questions under both categories if they fulfill the user need.

- **Comparison** and **Preference**
  Explanation: Similar to the above, even though the answer types are distinct, the


Figure 3.5.: Weighted confusion matrix for the user need classification of the taxonomy for annotators and GPT4. The weights are not normalized and can be higher than 1.

difference may not be clear from the question itself, and both types provide viable answers that fulfill the user need.

- **Methodology**, **Strategy**, and **Evaluation**
  Explanation: This confusion arises because all three categories relate to the user's need for advice or suggestions. The distinctions between these categories are not always evident from the question itself, and the user need could be fulfilled by any of the three categories.

### 3.1.6. Revised Taxonomy Structure

Based on the taxonomy created and the findings from previous sections, we aim to refine the classification of user needs to reduce the ambiguity within the taxonomy. To accomplish this, we consolidate certain categories that show significant overlap and interchangeability, and recalculate the Cohen Kappa scores for these combined categories.

The first and most evident consolidation stems from the low Cohen Kappa scores, indicating confusion within the high-level user need category "Summarization", which encompasses "Brief Overview" and "Complex Definition". These categories are deemed redundant, especially when considering the multi-label classification of "Elaboration" and "Definition". Moreover, the essence of "Summarization" is adequately covered by "Elaboration", allowing for the removal and merging of these categories as previously mentioned.
The second consolidation involves combining "Comparison" and "Preference" into a single category, due to the blurred lines between them in certain queries, where either category could address the user need. The aspects previously captured by "Preference" are now encompassed within "Comparison", along with "Evaluation" or "Elaboration".
The third consolidation merges "Methodology" and "Strategy" into one category, reflecting their significant overlap. "Evaluation" remains distinct, catering to a unique user need for assessed opinions.
The recalculated Cohen-Kappa Scores for the revised taxonomy are illustrated in Figure 3.6.

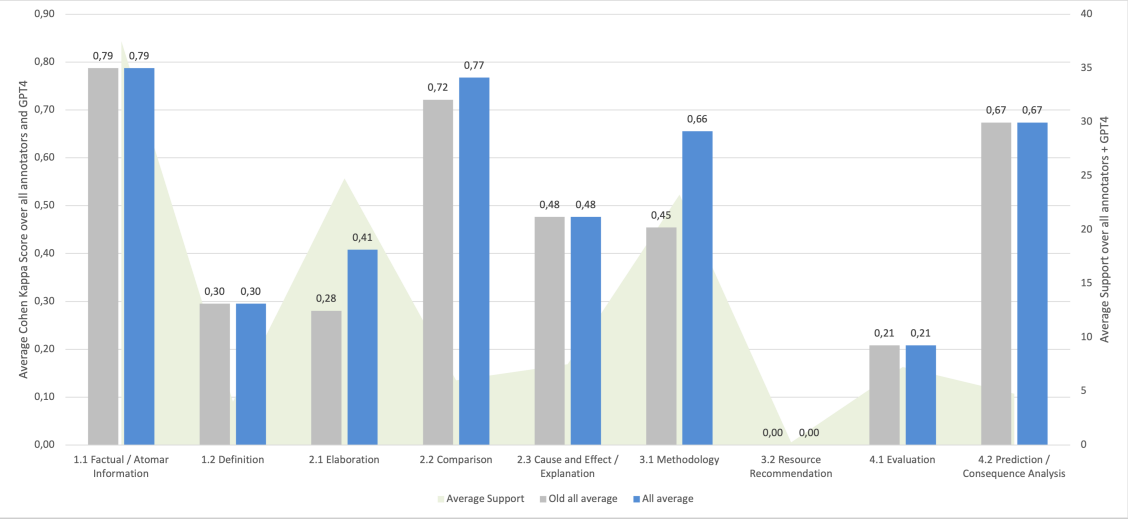


Figure 3.6.: Cohen Kappa Scores and support for each low level category of the revised (merged) taxonomy. Grey represents the average scores the respective categories of the original taxonomy, while blue shows the Cohen Kappa scores of the above described revision.

### 3.1.7. Dataset Construction

Based on the created taxonomy and the previous work in the Section 2, "Related Work and Background", we propose a dataset structure for a new Q&A dataset with a focus on attribution. The structure aims to provide an abstract and standardized way to present attribution needs for a given question that allows analyses and comparisons of different attribution methods. The dataset structure is as follows (Class structure):

**ComplexQADataset**: This class represents a dataset for complex questions and answers, including claims and sources. It is structured to handle multiple answers per question, multiple claims per answer, and multiple sources per claim.

`Attributes:`

- `questions`: A list that stores all question instances.
- `stat_dict`: A dictionary for storing statistical information about the dataset.
- `stat_dict`: A dictionary for storing statistical information about the dataset.
- `cb_info`: A list for callback information.
- `base_path`: A string indicating the base path of the dataset

**Question**: Nested within the `ComplexQADataset` class, this class represents individual questions in the dataset.

`Attributes:`

- `question_text`: The text of the question.
- `user_need`: A list of user needs associated with the question.
- `question_structure`: The structure of the question.
- `source_dataset`: The source dataset from which the question is derived.
- `answers`: A list to store instances of answers related to the question.
- `retrieved_context`: A list to store contextual information retrieved for the question.

**Answer**: Nested within the `Question` class, this class represents the answers associated with each question.

`Attributes:`

- `answer_text`: The text of the answer.
- `answer_system`: The system that provided the answer.

- `claims`: A list to store instances of claims related to the answer.

**Claim**: Nested within the `Answer` class, this class represents the claims made within each answer.

`Attributes:`

- `claim_text`: The text of the claim.
- `sources`: A list to store instances of sources supporting the claim.
- `retrieved_context`: A list of additional contextual information related to the claim.
- `claim_relevance_classification`: A classification of the relevance /cite-worthiness of the claim.

**Source**: Nested within the `Claim` class, this class represents the sources that support each claim.

`Attributes:`

- `source_text`: The text of the source.
- `source_link`: The link(s) to the source, which can be a single link or a list of links.
- `direct_quote`: A boolean indicating whether the source is a direct quote.
- `attribution_type`: The type of attribution for the source.
- `attribution_system`: The system responsible for the attribution of the source.

This hierarchical structure allows for structured analyses of each hierarchy level and the dependencies of different attributes.

As an initial list of question, we use the same reference list as in the previous section. This dataset contains of 100 questions, which are separated into 70 questions from ExpertQA and 30 questions from the Natural Questions dataset. For the classification of each question, we utilize the revised taxonomy introduced in the previous section, alongside classifications from both annotators and GPT-4. The categorization of user needs and questions encompasses **every possible user need**, ensuring that differing classifications by annotators are considered valid. This approach is similarly applied to the analysis of question structure. The structure of this dataset is elaborated upon in subsequent sections.

The following figures present statistics for the 100 questions based on the revised taxonomy, as well as classifications from the annotators and GPT-4. These statistics are visualized in Figure 3.7. This graph reveals several notable insights. Firstly, over half of the questions

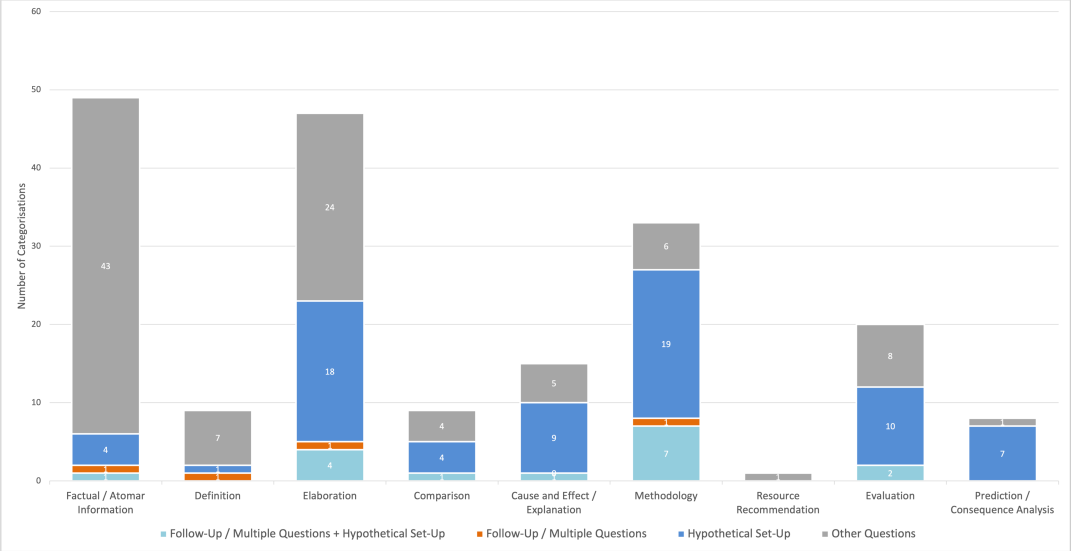


Figure 3.7.: Classification support for the revised taxonomy, incorporating classifications from each annotator and GPT-4. The diagram further divides each user need by question structure.

categorized as "Methodology" are structured around a hypothetical scenario, as are questions labeled as "Predictions / Consequence Analysis". Additionally, although not directly illustrated in this graph, it is noteworthy that 29 of the questions classified under "Factual / Atomic Information" originate from the "Google Natural Questions" dataset. This observation underscores the argument that complex datasets and their questions are framed significantly differently from those in traditional Q&A datasets.

Figure 3.8 illustrates the co-occurrence of user needs within the dataset, clearly showing that the "Elaboration" user need frequently intersects with multiple other categories. This pattern strongly suggests the necessity for a multi-label classification approach to accurately represent user needs. The co-occurrence matrix displays the proportion of questions on the y-axis that are classified under the same user need on the x-axis.

### 3.1.8. Results & Summary

In this section, we demonstrate the significant differences between traditional Q&A datasets and the ExpertQA dataset, which specifically targets questions addressed to experts and answered by Large Language Models. Upon analyzing existing taxonomies, we concluded that most lack sufficient depth and breadth or exhibit structural inconsistencies. To address these inconsistencies, we developed a new taxonomy comprising two dimensions: one focusing on user goals and the other on the structure of the question. Our evaluation of this taxonomy revealed general robustness, though some categories require further clarification. Additionally, we proposed a dataset structure for a new Q&A dataset, emphasizing attribution

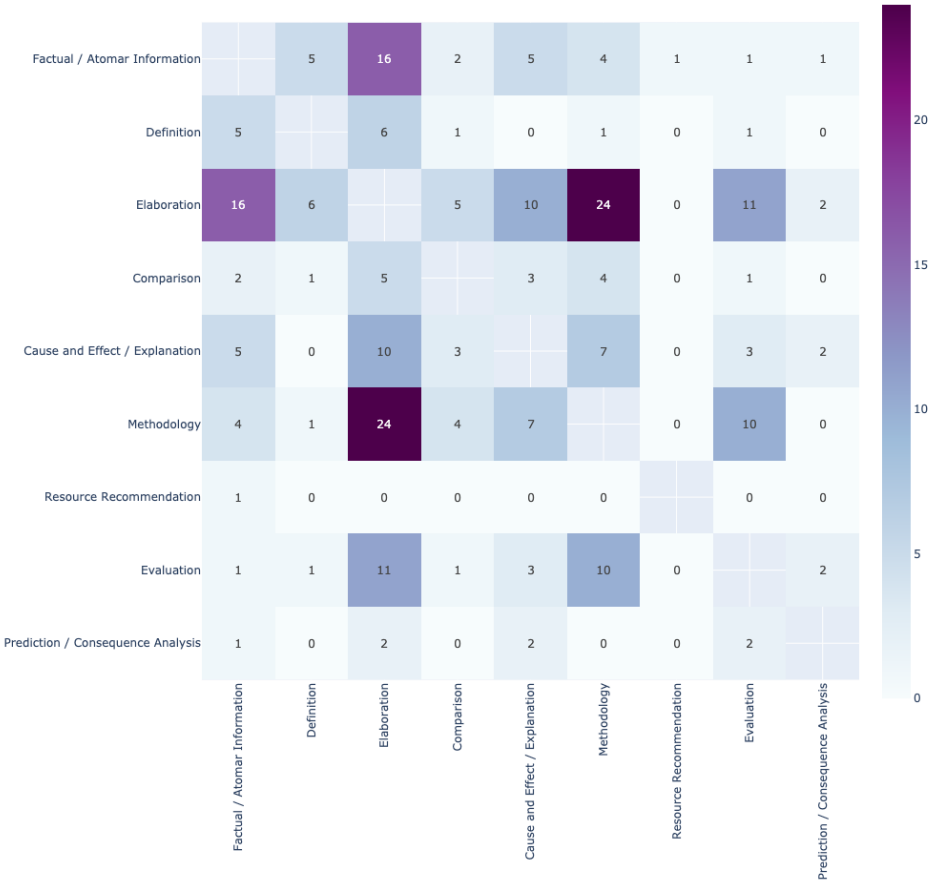


Figure 3.8.: Co-occurrence of user needs within the revised taxonomy in the dataset.

and based on our newly created taxonomy. This dataset encompasses 100 questions, divided into 70 from ExpertQA and 30 from the Natural Questions dataset. In the following, we use this exact dataset and its structure for investigating attribution.

## 3.2. Comparing Existing Answer Attribution Approaches in the Context of LLMs

The goal of this section is to provide an analytical overview of existing attribution approaches and extract weaknesses and strengths of each approach. For that, we utilize the taxonomy created in Section 3.1 and compare those approaches in the context of large language models. This section aims to answer Research Question 2, as stated in Section 1.2.1:

**RQ2: Patterns and Weaknesses of LLM Responses in Current Approaches**

2.1 How do current methods for answer attribution perform on the newly created

dataset?
2.2 What are the common patterns of weaknesses or errors in LLM responses?
2.3 In what ways can these weaknesses be ameliorated?
**Deliverables:** A comparison of attribution approaches and individual segments of attribution within the prebuild dataset.
A modular framework for testing individual attirbution components and complete end to end approaches.

### 3.2.1. Methodology

To provide a comprehensive comparison of existing approaches, we utilize the points and analysis from the related work described in Section 2.5. We implement modules for each step of the attribution process and compare the outcomes of different approaches to identify weaknesses. Our analysis is grounded in the taxonomy and categories for user needs that we have established, as well as the various steps involved in the attribution process.

### 3.2.2. Steps of Attribution Systems

The primary distinction in attribution systems lies in whether they are based on RTR (Retrieve-Then-Read) or PHR (Post-Hoc Retrieval) setups. For PHR systems, we examine the following steps:

1. **Answer Formulation**: The answer system (LLM) formulates a response $r$ based on the input question.
2. **Claim Segmentation**: Segmenting the answer $r$ into individual claims $c_i$.
3. **Claim Relevance Determination**: Assessing the relevance of each claim $c_i$ for attribution needs.
4. **Information Retrieval**: Retrieving a list of relevant sources $s_i$.
5. **Source-Claim Inference**: Determining whether a list of sources $s_i$ substantiates each claim $c_i$.

For RTR systems, the sequence of steps is reordered, with information retrieval preceding answer formulation:

1. **Information Retrieval**: Retrieving a list of relevant sources $s$ for the question $q$.
2. **Answer Formulation**: The answer system (LLM) creates a response $r$ based on the question $q$ and the retrieved source list $s$.
3. **Claim Segmentation**: Segmenting the answer $r$ into individual claims $c_i$.
4. **Claim Relevance Determination**: Assessing the relevance of each claim $c_i$ for attribution needs.

5. **Source-Claim Inference**: Determining whether a list of sources $s_i$ substantiates each claim $c_i$.

Different methods for each step were outlined in the related work and background section. The answer formulation is based on different LLM systems, such as GPT3.5, GPT4, Gemini and Mixtral. Since the goal of this thesis is not the comparison of different models, but the comparison on how to attribute the answers of models for complex questions, different model behaviors are only covered if they explicitly influence the attribution process.

As an initial benchmark, we use answers created by GPT3.5 (`gpt-35-turbo`, Microsoft Azure API, model version 0613). The prompt for answering questions can be found in the appendix. This model is chosen precisely because it has a known tendency to hallucinations, enabling us to compare the attribution approaches and attribution category without only having true answers.

### 3.2.3. PHR System

As described above, the first step for attribution in PHR attribution systems is to split the provided answer into claims. We define a claim as follows:

> **Claim Definition**: A claim is a statement or a group of statements that can be attributed to a source. The claim is either an word-by-word segment of a previous provided answer or semantically entailed by the answer.

There exist multiple ways to define and structure a claim, each resulting in different technical approaches for claim segmentation and each having different implications for the subsequent steps of the attribution process.

The first and most obvious step for segmenting an answer into claims is to use the syntactical structure of the answer, segmenting it into sentences, paragraphs, or other syntactical units. Sentence segmentation serves as a baseline for comparing different approaches. Approaches like Malaviya et al. [59] use this level of granularity to define claims. This segmentation is executed using the `spaCy` library. To validate the segmentation, we sample 20 random questions from the ComplexQADataset. The different segmentation systems are evaluated based on the number of atomic facts each claim contains and the number of claims they generate. An atomic fact is defined as in the related work and background Section 2.5.3.

An additional model for claim segmentation is based on the work of Chen et al. [61], where text is segmented into propositions, which aim to represent atomic claims. The precise definition is as follows:

> **Proposition Definition**: "Given a sentence $S$ with tokens $[t_0, t_1, \ldots, t_n]$ from a document $D$, a system is expected to identify the set of propositions $P \subseteq 2^S$, where each proposi-

tion $p \in P$ is represented by a unique subset of tokens from sentence $S$." [61]

The best-performing model from the paper is used for comparison. It is accessible via the `huggingface` library [89]. The model is a fine-tuned version based on the T5 checkpoint 1.1 [63]. Importantly, the model is designed to work on sentences, not complete paragraphs. This has multiple implications for performance and cross-sentence references, which is discussed later. Initially, this segmentation is based on the previous sentence segmentation performed by the spaCy library. If the model is prompted to segment complete paragraphs, the output changes form completely and turns out to be mostly nonsensical for the task, as the following example shows:

**Example Question**: "How to track the interface between two fluids?"
**Example Answer**: "To track the interface between two fluids, you can use various techniques depending on the specific situation and the properties of the fluids. Here are [...]"
**Output SegmenT5 - Paragraph Level**:

1. "There is a sentence."
2. "The sentence is about something."
3. ...

**Output SegmenT5 - Sentence Level**:

1. "Tracking the interface between two fluids involves using various techniques."
2. "The techniques used to track the interface depend on the specific situation."
3. ...

Note that the example provided above is cherry-picked and illustrates a notable performance difference but does not offer a comprehensive overview of the model's performance on paragraphs. The model is designed to work with sentences, and while the output for paragraphs may be effective in some instances, it is generally not meaningful.

The final variation of segmenting an answer into its individual claims utilizes pretrained LLMs and prompting to segment sentences into individual claims, as demonstrated in Min et al. [8]. In this approach, the model is prompted to segment the answer into claims, and the resulting output is subsequently revised by human annotators. The model operates on a sentence level, leading to similar issues encountered with the T5 model, including the lack of context, among others. The model used for this approach, GPT-3.5 Turbo, is the same as that used for formulating answers. We replicate this method using the same prompt (*"Please breakdown the following sentence into independent facts:"*), amended with meta-information and instructions for the model on formatting the output. This adjustment is crucial for enabling post-processing of the output and its evaluation in a consistent manner. The prompt is

provided in the appendix. Given that the approach was applied at the sentence level, we again use sentence segmentation as a baseline for the purpose of comparison
The tables and figures illustrate this approach. Table 3.4 shows the high level differences between the three claim segmentation approaches.

As expected, the average number of characters of the "atomic" facts created by GPT-3.5 and

| **Segmentation System** | **Number of *c*** | **Unique #*c*** | **avg. len(*c*)** | ***c* / Sentence** |
|---|---|---|---|---|
| spaCy_sentences | 938 | 855 | 103.2 | 1.00 |
| gpt35_factscore | 3016 | 2684 | 61.4 | 3.2 |
| segment5_propsegment | 2676 | 2232 | 54.2 | 2.85 |

Table 3.4.: High Level comparison of the different claim segmentation systems.

T5 is significantly smaller than the original sentence length. It is also noteworthy that the atomic facts/claims generated by GPT-3.5 are longer in characters and more numerous per sentence as well. In addition, the number of unique claims per answer and the number of claims per answer differ significantly by on average 12% and up to 16.5% for SegmenT5. The segmentation systems create duplicated claims for the same answer, which is a sign of an error prone system. Alternatively, these claims are created by the existing repetitions in the answer itself, where two independent sentences are so similar that they create exact string matches.
Figure 3.9 displays the average number of claims per classified user need: It is noteworthy

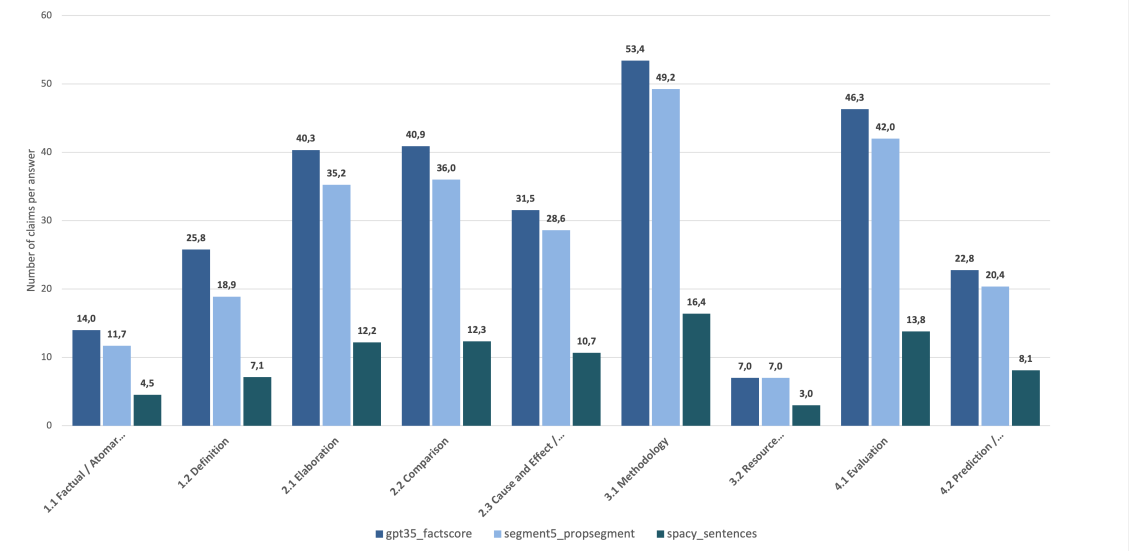


Figure 3.9.: Average number of claims per classified user need. Claims are used multiple times due to user need being a multi-label classification.

that the different classes of user needs exhibit significantly different numbers of claims per

answer, even though the answers are provided by the same system for all questions. If one disregards the "Resource Recommendation" class, due to its support of only one, "Factual and Atomic" questions have the lowest number of claims per answer. This outcome is logical since these types of questions should, in theory, be answerable by an unambiguous and single atomic claim. However, GPT-3.5, as the answering system, responds with an average of 4.5 sentences and between 11.7 and 14.0 atomic claims per answer. Conversely, questions requesting a methodology (user need category 3.1) have the highest number of claims per answer, averaging around 50.

**Qualitative Analysis**

For a qualitative analysis of these segmented claims we annotate 122 claims for a randomly selected question by hand. The categories for annotations are aligned with multiple reference papers, such as Malaviya et al. [59] and Chen et al. [61]. The categories are as follows:

1. **Atomic**: The claim contains a single atomic fact.
2. **Independent**: The claim can be verified without additional context.
3. **Useful**: The claim is useful for the question.
4. **Without Error**: The claim does not contain structural errors, e. g. being an empty string.
5. **Repetition**: The claim is a repetition of another claim from the same segmentation system.

Each category is binary, meaning that a claim can be annotated with multiple categories. In addition, we provide a binary category per answer which should give an indication about the exhaustiveness of the segmentation in relation to the claim. The question answer which was analyzed and segmented was given to the following question: *"A 55 year old male patient describes the sudden appearance of a slight tremor and having noticed his handwriting getting smaller, what are the possible ways you'd find a diagnosis?"*. Given that this question results from the medical domain, the claims are expected to be more complex and the segmenting model would need to have medical domain knowledge as well. The following table (Table 3.5) shows the result of the qualitative analysis.

The most noticeable outcome is that the `spaCy_sentences` segmentation system performs

| | Atomic | Independent | Useful | Without Error | Repetition |
|---|---|---|---|---|---|
| gpt35_factscore | 53/56 | 8/56 | 44/56 | 48/56 | 13/56 |
| segment5_propsegment | 40/53 | 6/53 | 28/53 | 34/53 | 18/53 |
| spaCy_sentences | 3/15 | 3/15 | 10/15 | 11/15 | 3/15 |

Table 3.5.: Comparison of the claim quality for the different segmentation systems.

significantly differently compared to other systems. It is structurally distinct and targets a different level of claim segmentation. Given that answer sentences, particularly in the example domain, can be lengthy and intricate, there is no inherent objective to limit each sentence to a single atomic fact. Consequently, the score for "Atomic" claims stands at 20%. Intriguingly, even as the context window extends with longer sentences, only 20% of these are independently verifiable. This means they do not require additional context from the question or answer for validation. Due to the complexity of the answers, most sentences reference a preceding sentence in the paragraph, often mentioning "the patient" or "the symptoms." This approach does not facilitate verification, as a reader possessing only this claim cannot possibly confirm its contents. The usefulness of the claims is comparatively high for sentence segmentation and GPT-3.5-based segmentation but diminishes for the SegmenT5 segmentation. Although most claims are error-free, it is notable that all systems produce erroneous outputs. Specifically, for this question, `spaCy` segments four empty strings as individual sentences. It is plausible that errors in other segmentation systems stem from this issue, as they rely on `spaCy` segmented sentences as input. This dependency also results in repetitions, primarily based on incorrect claim segmentation. The following list provides a positive and negative example claim for each category.

1. **Atomic**
   *Positive:* "Seeking a second opinion helps" (`gpt35_factscore`)
   *Negative:* "Brain tumors or structural abnormalities are among the possible causes that these tests aim to rule out." (`gpt35_factscore`)

2. **Independent**
   *Positive*: "Parkinson's disease is a cause of changes in handwriting." (`segment5_propsegment`)
   *Negative*: "Imaging tests may be ordered." (`segment5_propsegment`)

3. **Useful**:
   *Positive*: "There are several possible diagnoses that could explain the sudden appearance of a slight tremor and smaller handwriting." (`gpt35_factscore`)
   *Negative*: "The patient is a 55-year-old male." (`segment5_propsegment`)

4. **Without Error**
   *Positive*: "The patient is experiencing smaller handwriting." (`gpt35_factscore`)
   *Negative*: "The sentence is about something." (`segment5_propsegment`)

Based on these findings, we conclude that automatic claim segmentation faces three main challenges: First, to provide independently verifiable claims, the segmentation system requires more context than just the sentence. It might be necessary to include the entire paragraph and the question in the model. Second, the segmentation system needs to be capable of handling domain-specific language. The medical domain serves as a good example, as the language used in medical texts is often complex and necessitates extensive domain knowledge. Third, if the goal is to identify individual atomic facts, the segmentation system needs to operate at a more granular level than sentences.

**Claim Relevance Determination**

The usefulness (relevance) of a claim is evaluated based on its relation to the question. The precise formulation for this step is provided in the context of this thesis as follows:

> **Usefulness Definition**: Given a question or query $q$ and an corresponding answer $a$, a claim $c$ with $c \in a$ is useful if it provides relevant information to satisfy the user's information need.

Based on this definition, multiple underlying results are made transparent. First, the usefulness of a claim is inherently tied to the user need. In addition, usefulness can give a direct indication on wether it is worth to check the claim. We use the table of claims referenced above (3.5) to evaluate automatic systems for classifying the usefulness and check-worthiness of claims. Since this task is very specific, there are few direct comparisons for this step in existing literature. Most publications in the realm of attribution evaluate the answers and individual claims for their relation to the question, but they do not perform this evaluation automatically, relying instead on annotators [59, 8].
Wang et al. [65] directly implement this step by using ChatGPT (GPT-3.5) and comparing it against the baseline assumption that every claim is check-worthy. They divide the evaluation of claim check-worthiness into two distinct subtasks, where the first one determines whether a claim contains a factual statement and the second one classifies each claim into four categories of check-worthiness: `factual claim, opinion, not a claim, other`. The majority vote outperforms ChatGPT in terms of accuracy. The exact prompt for the model is not provided in the paper itself but in the corresponding GitHub repository [90]. It is not explicitly clear if this exact prompt was used in the paper itself. We assume so and reproduce results on our data. The prompt is provided in the appendix, the following is a cut out:
*"You are a factchecker assistant with task to identify a sentence, whether it is 1. a factual claim; 2. an opinion; 3. not a claim (like a question or a imperative sentence); 4. other categories. Let's define a function named checkworthy(input: str).[...].*

The first task itself is not reproduced as it appears redundant with the categorization into "factual claims" and "not a claim". The results are provided in the table below, using the above prompt and the same 122 claims as for the previous task. Again, for the purpose of postprocessing, meta-information is added to the prompt. Figure 3.10 shows the confusion matrix of the human annotator vs. the claims labeled by GPT3.5.

It is evident that, although most claims are identified by both the human annotator and GPT-3.5 as "factual claims," there is a discrepancy in the classification of "not a claim" and "other categories." The annotator predominantly categorized claims that were not "factual claims" as "other category," whereas GPT-3.5 labeled these as "not a claim." While this difference is notable, the crucial distinction lies between "factual claims" and the other categories, as this differentiation forms the basis upon which the rest of the pipeline relies.

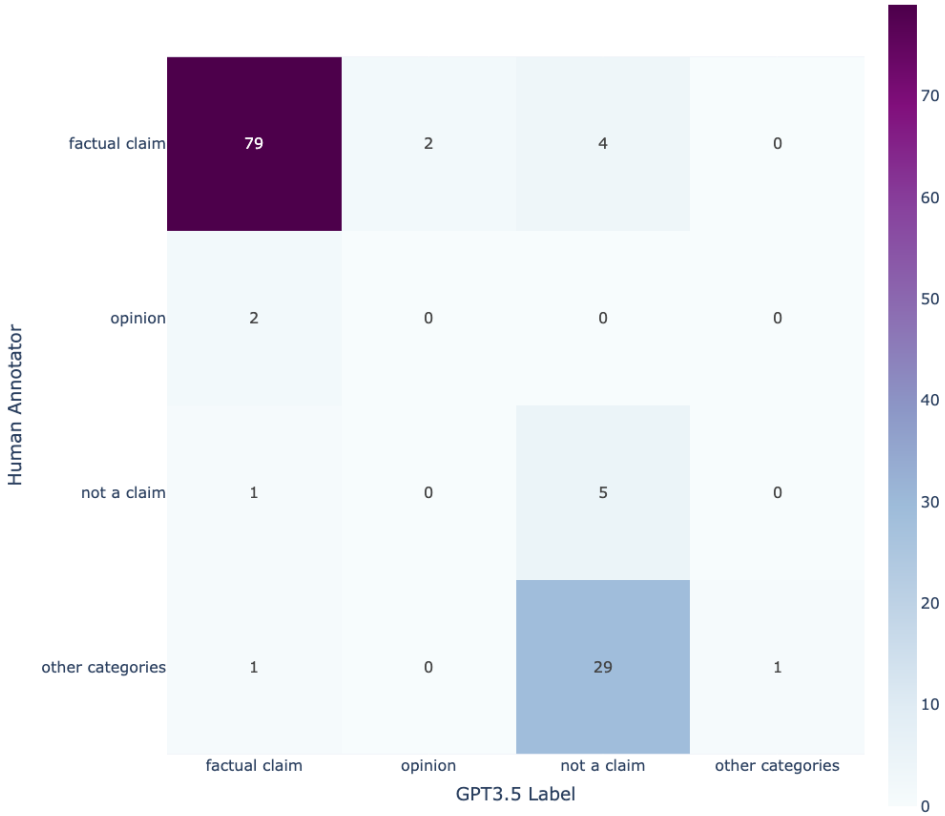


Figure 3.10.: Confusion Matrix for the classification of claims into different categories to evaluate their check-worthiness.

For the analysis of both subtasks, claim segmentation quality and claim checkworthiness, the 122 classified claims are used to examine the Spearman correlation coefficient between the initially provided categories and the categories provided by Wang et al. [65]. It is significant that the "factual claim" category shows a positive correlation with every category in the qualitative claim analysis, as outlined in Section 3.2.3. This finding is particularly intriguing for "Independent Claims," which are considered valid and factual only when sufficient context is provided, rather than when they are viewed in isolation. The following example, which was classified as non factual because of its non-independence, illustrates this point:

**Claim**: "Researching symptoms and medical conditions online"
*Categorization*: "Not a claim" by GPT-3.5
**Example Question**: "How would you diagnose a 55-year-old male patient who reports the sudden onset of a slight tremor and notices his handwriting becoming smaller?"

**(Human) Correction**: "Researching symptoms and medical conditions online is a valid method for diagnosing a patient who reports the sudden onset of a slight tremor."
*Categorization*: "Factual Claim" by GPT-3.5

This example highlights that a claim, while not inherently useful on its own, gains relevance within the context of a specific question. This underlines the necessity for claims to be independent to be considered valid, useful, and worth checking, pinpointing a limitation in current methodologies. The complete table of correlations is presented in the appendix.

We conclude that assessing the usefulness, relevance, and check-worthiness of a claim presents a challenging task for current systems, as they fail to provide the necessary context for evaluating the claims. Despite LLMs like GPT-3.5 not perfectly aligning with human annotation, there is a significant overlap that could be leveraged to achieve more closely aligned results.

We use the system proposed in FactScore [8] with GPT3.5 to classify each claim into one of the above described categories. The results are presented in the table below (Table 3.6).

| **Segmentation System** | **Unique #*c*** | **# factual** | **# not a claim** | **# opinion** | **# other** |
|---|---|---|---|---|---|
| spaCy_sentences | 855 | 550 | 244 | 26 | 35 |
| gpt35_factscore | 2684 | 2317 | 258 | 68 | 41 |
| segment5_propsegment | 2232 | 1878 | 290 | 36 | 28 |

Table 3.6.: Classification distribution of the different segmentation systems and claim classification for the system proposed in Factscore [8].

**Information Retrieval**

The information retrieval step in the attribution process is arguably the most critical, especially for a post hoc retrieval process. Although a significant body of literature on the outputs of fact-checking by LLMs addresses steps including the natural language inference of answers and context, there is noticeably less focus on the retrieval aspect in most publications. Without retrieved content for a specific claim, verification in subsequent steps becomes unfeasible. Even when a publication mentions a Post-Hoc Retrieval system, it often lacks specific details about the retrieval system used or its impact on performance. In ExpertQA [59], two retrieval systems and corpora are utilized, employing a bag-of-words retrieval function (BM25, [91]): first, Sphere [92], a static dump of CommonCrawl, and second, Google Search results, without additional details on the precise method for incorporating Google Search results.
Wang et al. [65] adopt a similar strategy, generating search queries for each claim using GPT-3.5 and retrieving content from relevant web pages through Google Search. However, the paper does not detail the methodology for utilizing and extracting content from Google

Search.
Generally, there are four different approaches and specifications for retrieval:

1. **Static Corpus**: A static corpus comprises a predefined set of documents utilized for retrieval, as seen in ExpertQA [59], which relies on the CommonCrawl dataset. Its main advantage lies in reproducibility and suitability for benchmarking. The primary limitation, however, is the corpus's static nature, failing to mirror the evolving state of knowledge and proving inadequate for updating claims that become outdated.

2. **Dynamic Corpus**: Contrasting with a static corpus, a dynamic corpus is regularly updated to reflect the latest state of knowledge, epitomized by the web. This approach, employed by Wang et al. [65] and Malaviya et al. [59], utilizes Google Search results. Its principal advantage is its currentness, applicable to any claim. The main challenges, however, stem from difficulties in reproducibility and accurately accessing relevant information for a specific claim.

3. **Retrieval Query - Question**: Employing the question as the retrieval query represents the most straightforward method. Although intuitively appealing from a linguistic standpoint, this method proves less effective in practice for web queries and other retrieval systems. Complex, fully formulated questions often fail to produce direct answers in web searches, as search engines primarily rely on keyword matching rather than semantic comprehension. Furthermore, the answers generated by an LLM may contain claims not readily found by searching the web or a reference corpus using the question.

4. **Retrieval Query - Claim**: Utilizing claims as retrieval queries allows the retrieval system to directly provide sources for the claim, theoretically enhancing the source-claim connection. Moreover, atomic claims can be crafted into conventional web queries, potentially improving retrieval performance.

In general, several papers suggest that PHR systems perform better, supporting the argument made in point (3). Retrieving documents based solely on the question often fails to yield relevant answers. Using the language model as a retriever and adopting a more granular approach from that point forward appears to be more effective [59, 39].

We don't adopt the static corpus approach (Sphere, [92]), as the size of this corpus is prohibitively large at approximately 1TB for use in this thesis. Moreover, the disadvantages discussed previously render a static approach unsuitable for our research.
We opt for retrieval using Google Search for claims identified as useful and factual, as detailed above. This is facilitated by the Google Programmable Search Engine API [93], which provides various configuration options listed in the appendix. Notably, we enable the option to search the entire web and specify a list of preferred domains, including: `["en.wikipedia.org", "arxiv.org/", "stackexchange.com", "stackoverflow.com"]`. Even the exact approaches can't be directly compared, we can orient ourselves on the results of Wang et al. [65] and compare different methods for retrieval.

The individual steps that are implemented and compared are the following:

1. **Retrieval Query - Question**: Using the question / query as a search string for the search engine (Google Search).
2. **Retrieval Query - GPT-3.5**: Using GPT-3.5 to generate a search query for the claims and retrieve content from relevant web pages through Google Search. [65]
3. A **sliding-window** approach to segment retrieved passages [65]
4. **Sentence-BERT [94] embeddings** for the segmented chunks [65].
5. **A re-ranker algorithm** for the segmented chunks / embeddings [65].

Note that the sliding-window approach and the re-ranker are not further specified in the paper. Since each has a significant number of possible parameters, such as window size and overlap for the text splitter, or the exact indexing algorithm for the re-ranker, we cannot compare these implementations with different methods. Instead, we only evaluate the efficacy based on our implementation.
For re-ranking, we use FAISS [95], and for the sliding window approach, we utilize a text splitter provided by the LangChain [96] library, which splits text into chunks of a predefined length based on a specific character (we use spaces, and a targeted length of 256 characters per chunk).

The precise goal of this task is **to evaluate whether the retrieved sources provide any useful content for the claims** segmented from the answer to a query or question. This is accomplished by comparing the retrieved sources with the claims and evaluating the overlap. The exact function takes a claim and the retrieved sources as input and returns a binary value for each claim, indicating whether the source is relevant to the claim. This is evaluated by a *Human Annotator*, a DeBERTa [97] *NLI Model* implementation from Hugging Face [98], and a GPT-3.5 prompt-based evaluation. The specific prompt for the GPT-3.5 model can be found in the appendix. We only evaluate claims that are classified as "factual claims" in the previous subtask. We use all claims segmented by the GPT-3.5-Factscore claim segmentation. From the initial 6,630 claims across all three segmentation systems, 3.016 were produced by the segmentation system proposed in FactScore [8], of which 2.684 claims were unique on a question level. Of those 2.684 claims, the claim classification system classified 2.317 claims as "factual claims", as presented in Table 3.6. These claims are used as a baseline for information retrieval. Figure 3.11 shows the exact workflows of each compared method.
For retrieval based on the question, we store the complete text-based content of the top 3 web search results that could be accessed, in a FAISS-vector database [95] with Sentence-BERT embeddings. After that, we query each claim against the vector store for that question and retrieve the top 5 most similar chunks. Each chunk is provided to a human annotator, the DeBERTa model, and GPT3.5, respectively as context. GPT3.5 and the human annotator additionally have access to the question /query. The same process is repeated for claim-based

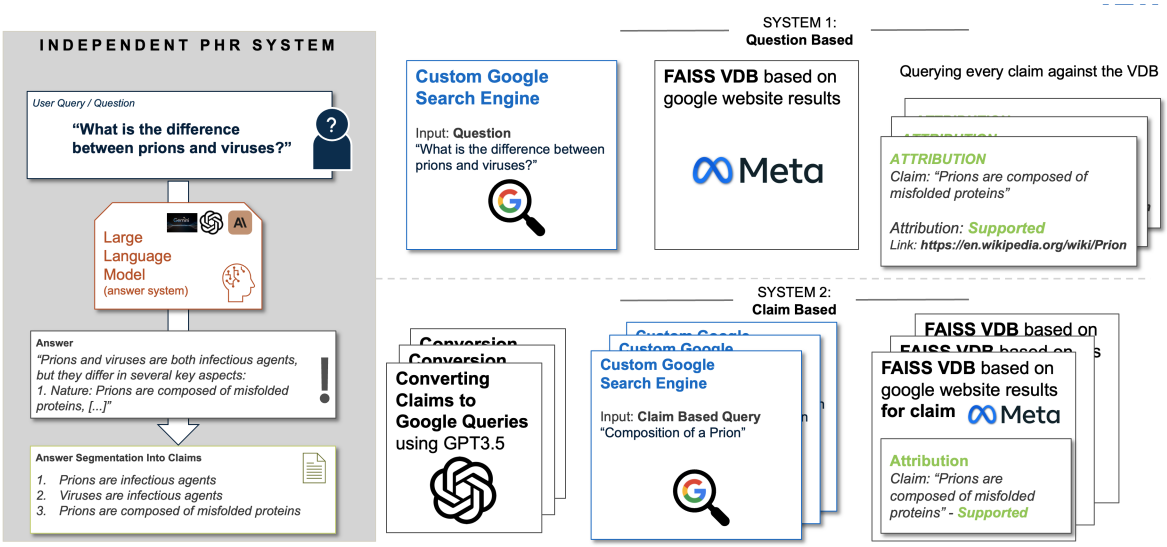


Figure 3.11.: Retrieval systems implementation visualization

retrieval, where the difference lies in the Google search results being based on each claim. Because the retrieval for each claim should be evaluated individually, the vector store is set up for each claim individually. The methodology of retrieving based on the question results in each claim receiving three different excerpts from web pages as sources to be evaluated against. The results are presented in the table below.

The human annotator evaluated 135 randomly selected triplets of question, context, and

| Retrieval System | Entailment | Contradiction | No Relation | Wrong Output |
|---|---|---|---|---|
| Human Evaluation | 6 | 0 | **129** | 0 |
| GPT3.5 | 2871 | 70 | **3153** | 908 |
| Deberta | 772 | 351 | **5879** | 0 |

Table 3.7.: Evaluation of the different retrieval systems for the classification of claims into different categories to evaluate their check-worthiness.

claim. "Wrong output," as a category, was observed only for GPT-3.5 and describes output that does not conform to the predefined classification options of the task. We focus on the class of "No Relation," which describes cases **where the annotation system asserted that there is no relation between the source and claim, indicating poor retrieval**.

The first noticeable observation is the significantly high difference in the classifications of "No Relation" between DeBERTa and GPT-3.5. The DeBerta model classifies around 84% of pairs to have no relation, whereas the few-shot GPT-3.5 model classifies around 45% of triplets to have no relation. This is counterintuitive, as one might assume that an entailment model receiving no additional context, as is the case with the DeBERTa Model in this task, would

classify pairs of (Context, Claim) as being related more frequently. Consider the following example:

> **Question**: "When telling patients that their last round of chemo is falling and would advise them to stop chemo, what is the best way to approach my patient when they hoped this last round would work?"
> **Claim**: "The purpose of offering these resources is to provide further assistance or support."
> **Context**: "process. In addition, an approach that uses effective communication with these patients and integrates their values with current medical evidence is needed. Communication is crucial in establishing trust with patients, gathering information, addressing"
> **DeBERTa**: No Relation
> **GPT-3.5**: Entailment

For the given example, one can argue from two perspectives: First, when considering only the claim and the context, there appears to be a similarity. However, key details are missing from both the context and the claim, preventing the establishment of a real relation. This represents the perspective of the DeBERTa model. On the other hand, the GPT-3.5 model may have a different perspective, given its access to both the question and the claim. It might be able to establish a relation by implicitly adding additional context to the provided context.
The second observation is based on the notably high number of "No Relation" classifications, particularly by the human annotator. A qualitative analysis of these results suggests that most triplets of (Question, Context, Claim) are essentially "nonsensical" and fail to provide any useful information. This outcome is plausible and likely a direct consequence of the error propagation from previously identified issues, such as non-independent claims, insufficient context in the context window, and suboptimal retrieval based on the question. Figures 3.12 and 3.13 display the confusion matrices for the different retrieval systems, using DeBERTa as a reference.
Lastly, based on a qualitative analysis, the default context window of around 256 characters is not sufficient for the selected soures and claims. The confusion matrices indicate that the DeBERTa model and the human annotator classify the majority of triplets similarly. However, the GPT-3.5 model demonstrates significantly different classifications. When using the human annotation as a baseline for comparison with DeBERTa, DeBERTa achieves a precision of 98.1% and a recall of 79.8%. Additionally, the comparison between the DeBERTa model and the GPT-3.5 implementation reveals similarities as well (3.12). With DeBERTa as a baseline, GPT-3.5 achieves a precision of 87.1% and a recall of 46.7% in the "No Relation" category. This performance is notably inferior to that of DeBERTa. The main aspects of this approach can be summarized as follows:

1. Retrieval based on the question is ineffective for the task of sourcing relevant claims.
2. Verifying the relation of non-independent claims to sources is, in some systems, simply not feasible.

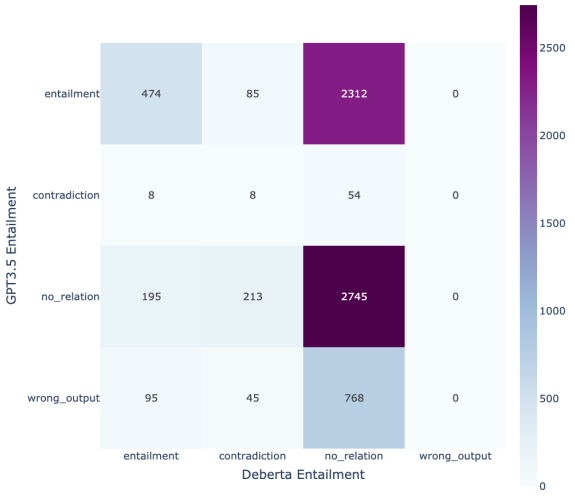


Figure 3.12.: Confusion Matrix for GPT-3.5 based claim source evaluation and DeBERTa based claim source relation for question based Google Search retrieval.

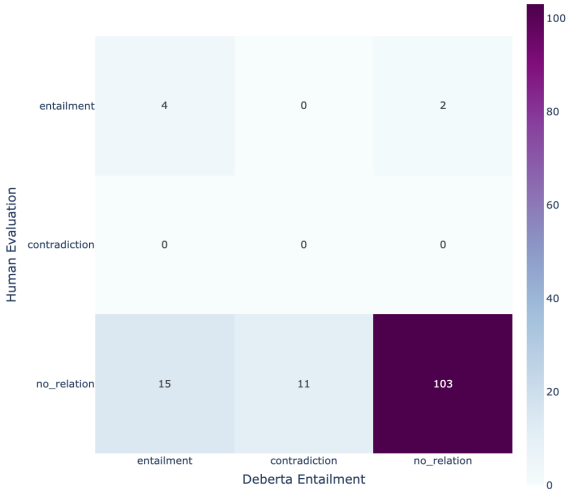


Figure 3.13.: Confusion Matrix for Human and DeBERTa based claim source evaluation for question based Google Search retrieval.

3. Error propagation becomes increasingly significant in a series of different approaches like the one currently under discussion.

4. A longer context window appears to generally improve the retrieval of meaningful sources.

### Claim Based Google Search

The second iteration for retrieval is based on Wang et al. [65] and their approach to use GPT3.5 to create Google search queries for individual claims. We can forsee one main issue with this approach, that already became apparent in the previous section: It is not feasible to create search queries based on non-independent claims alone. Therefore, we adapt the proposed approach slightly to include the question for additional context. We use the same sliding window approach, vector database and Google search engine as above. We only evaluate 89 claims since the retrieval process is computationally expensive and there are limits on Google search engine API requests. The exact prompt used in the reference paper can not be used, as it is not provided. We also base the claim evaluation on two different context window lengths, one being 512 characters and one being 256 as in the previous experiment. Two systems, up to 3 retrieved pages per claim and 89 claims result in a total of 396 claim-context pairs to be evaluated. The results are presented in the Table 3.8

| Method & CW | contradiction | entailment | no relation | wrong_out |
|---|---|---|---|---|
| GPT3.5 - 256 | 2 | 111 | 82 (36.0%) | 33 |
| GPT3.5 - 512 | 1 | 126 | 88 (38.6%) | 13 |
| DeBERTa - 256 | 12 | 37 | 179 (78.5%) | - |
| DeBERTa - 512 | 11 | 64 | 153 (67.1%) | - |
| DeBERTa Missing | - | - | - | 13 |
| GPT3.5 Missing | - | - | - | 13 |

Table 3.8.: Claim - source evaluation for claim-adapted Google Search retrieval.

"Missing" indicates that GPT3.5 was unable to generate a valid search query based on the claim and the question alone. It was prompted to return "Missing Information" if essential context was not provided for query creation. This occurred for two (question, claim) pairs, resulting in six (question, claim, source) triplets with missing information. Multiple significant findings are evident in the evaluation.

1. **DeBERTa: Reduction in "No Relation" Classification**: With the same evaluation setup (256-character context window size), the DeBERTa model classifies only 78.5% of claims as having no relation to the retrieved sources. This represents a reduction of around 5.5% points from question-based retrieval, indicating better performance and higher connection for claim-based retrieval.

2. **DeBERTa: Reduction for "No Relation" Classification with Larger Context Window**: Increasing the context window size to 512 characters, the DeBERTa model classifies 67.1% of claims as having no relation to the retrieved sources. This marks a reduction of around 11.4 percentage points from the 256-character context window size, suggesting that a larger context facilitates a more meaningful connection between the claim and the retrieved sources.

3. **GPT3.5: Question-Based "Wrong Output" Comparison**: With the same evaluation setup (256-character context window size), GPT3.5 produces incorrectly formatted output for 14.3% of (question, claim, source) triplets. This is within the same margin as question-based retrieval, where GPT3.5 produced incorrect output for 14.5% of triplets.

4. **GPT3.5: Character Window Based "Wrong Output" Comparison**: Increasing the character window size to 512 reduces the share of incorrectly formatted outputs to below half (5.6%) of its original, indicating that a larger context window helps decrease the number of incorrectly formatted outputs. This is supported by a qualitative analysis of the incorrectly formatted outputs, as exemplified below.

5. **GPT3.5 & DeBERTa: Different course of "No Relation" classifcations**: When increasing the context window of both GPT3.5 and DeBERTa, the number of "No Relation" classifications increases for GPT3.5 and decreases for DeBERTa. The hypothesis for this is that with more context, GPT3.5 will hallucinate less connection while DeBERTa

will also improve towards a better connection. Figure 3.14 and Figure 3.15 offer more detailed insights into the classifications by the different systems.

6. **GPT3.5: Incorrect Output Hypothesis**: We propose the hypothesis that if the (question, claim, context) triplets appear completely disconnected from each other, the likelihood of GPT3.5 producing incorrect output increases, as it attempts to elucidate the connection between the claim and the context. This is corroborated by the below output example, which was post-processed to be classified as "wrong output".

The following qualitative examples gives in indication to GPT3.5 producing incorrect output for the claim-based retrieval.

> **Question**: "Being a project manager in a software company, your team has found a critical bug causing many errors, now the project is behind schedule. What actions would you take to get project back on track and whom would you work with?"
> **Claim**: "The resources will work on fixing the bug."
> **Context**: "really want! Sometimes, simple persistence in attempting to reproduce the bug whilst debugging can be the most effective approach. For this strategy to work, the bug needs to be 'intermittent' rather than completely 'non-reproducible'. If you can repeat a"
> **GPT-3.5 Output**: "Claim: The resources will work on fixing the bug. Document: Sometimes, simple persistence in attempting to reproduce the bug whilst debugging can be the most effective approach. For this strategy to work, the bug needs to be 'intermittent' rather than completely 'non-reproducible'. If you can repeat a bug, you can fix it."

In the example provided above, because the claim is completely disconnected from the question and the context, GPT-3.5 attempts to answer the question by repeating the claim and document, rather than providing the relationship for which it is prompted.

> **Question**: "When was the last time New England won the Super Bowl?"
> **Claim**: "The game took place on February 3, 2019."
> **Context**: "Report. Archived from the original on March 26, 2015. Retrieved March 2, 2014. 'Patriots announce 2010s All-Decade Team'. Patriots.com. April 29, 2020. Retrieved April 30, 2020. [permanent dead link] 'The Patriots Hall of Fame Presented by Raytheon'"
> **GPT-3.5 Output**: "nli(question: str, claim: str, document: str) -> str"

In this instance, GPT-3.5 fails to provide any meaningful output, instead repeating a section of the original prompt. An example like this could potentially be addressed by an improved prompt or a better LLM, such as GPT-4. However, again, the claim and the context do not seem to be connected at all.

Figure 3.14 and Figure 3.15 show the confusion matrices for the different retrieval systems based on the context window size, using DeBERTa as a reference.

The most notable difference is the increase in agreeance between both models when increas-

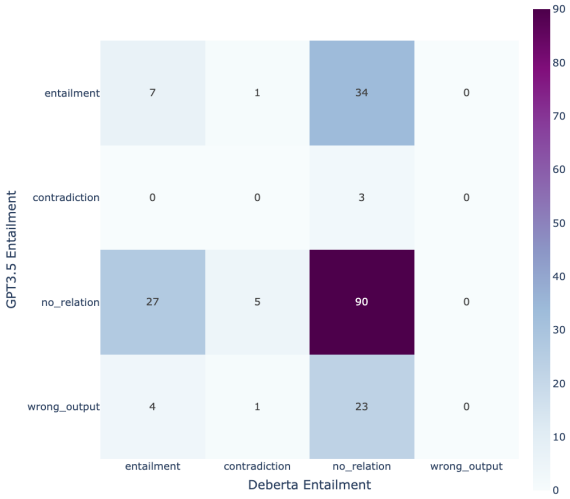


Figure 3.14.: Confusion Matrix for GPT-3.5 and DeBERTa based claim source relation, 256 character context window and claim based Google Search retrieval.

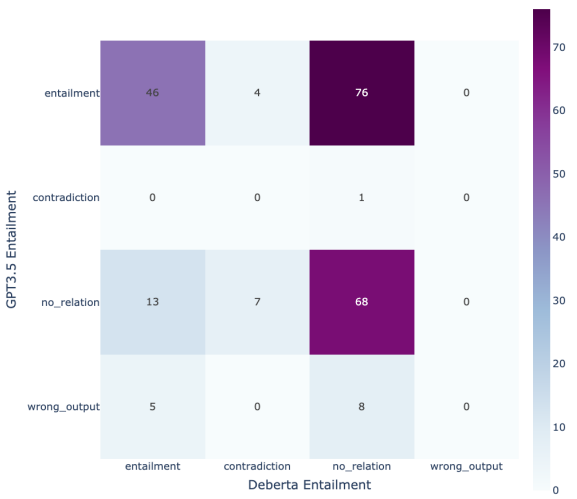


Figure 3.15.: Confusion Matrix for GPT-3.5 and DeBERTa based claim source relation, 512 character context window and claim based Google Search retrieval.

ing the context window. Using DeBERTa as a ground truth model, the F1 score for the "No Relation" classification in the 256 context window lies at 0.52 and at 0.56 for the 512 context window.

### Attribution Evaluation

The final step in the attribution process is to classify the relationship between the claim and the retrieved sources in terms of support. This has already been implicitly addressed in the previous section, where the connection between source and claim was evaluated. There, the focus was on whether there was **any connection** between the source and claim to assess the quality of the retrieval process. The aim of the attribution evaluation is to classify the **type of relationship** between the source and claim. The default categories for this task are *entailment, contradiction, neutral (no relation)*. These distinctions are generally adequate when evaluating precisely one source at a time, assuming a low likelihood of contradiction within the source itself. However, when the number of sources increases and the context window encompasses a concatenation of these sources, the potential for contradictions within the sources themselves arises theoretically. This can be examined by comparing the individual pairs of (claim, source) for each claim to determine whether a specific claim exhibits both a contradiction and an entailment relationship with two different sources.

The evaluation utilizes the same 89 claims as in the previous section, employed in the claim-based Google search retrieval process. Moreover, the comparison of precision, recall, and F1 scores for the question based system are detailed in Table 3.9. DeBERTa serves as the baseline for each verification. The agreement between the models is acceptable. After analyzing a

| Category & CW-Size | Precision | Recall | F1 | Support |
|---|---|---|---|---|
| Con. - 256 | 0 | 0 | 0 | 12 |
| Con. - 512 | 0 | 0 | 0 | 11 |
| Con. - Question Based | 0.11 | 0.02 | 0.04 | 347 |
| Ent. - 256 | 0.23 | 0.68 | 0.34 | 37 |
| Ent. - 512 | 0.37 | 0.72 | 0.48 | 64 |
| Ent. - Question Based | 0.17 | 0.61 | 0.26 | 771 |

Table 3.9.: Attribution type agreeance evaluation for claim-adapted and question based Google Search retrieval.

relatively long list of potential structural errors, we will proceed with an individual analysis based on qualitative examples. Let's first evaluate cases from the claim-based retrieval system where one of the models classified the source-claim relationship as a contradiction. What is notable is that the claim based models do perform worse on the specific attribution categories, compared to the question based retrieval, which outperforms on the "No Relation" classification. We use the claim-based retrieval system in the following qualitative examples.

**Question**: "What will the impact of AI-generated musical products have on the industry at large?"
**Claim**: "AI will not replace human musicians in the future of music."
**Context**: "Creativity. Also, AI-generated music must not replace or displace traditional forms of music creation. AI presents new opportunities for musicians. However, we must be aware of the potential risks it carries. Don't stand alone. As you consider the"
**GPT-3.5 Output**: "Entailment"
**DeBERTa Output**: "No Relation"
**GPT-4 Output (Browser Version)**: "No Relation"

**Question**: "when was the last time new england won the super bowl?"
**Claim**: "The game took place on February 3, 2019."
**Context**: "Hockey League. February 3, 2023. Archived from the original on September 3, 2023. Retrieved December 18, 2023."
**GPT-3.5 Output**: "No Relation"
**DeBERTa Output**: "Contradiction"
**GPT-4 Output (Browser Version)**: "No Relation"

DeBERTa is correct in the first example. The context seems to be very closely related to the claim, as it talks about the impact of AI-generated music on the industry. But it does not

entail the claim, because it only references the replacement of specific music forms, not the replacement of human musicians.
For the second example, DeBERTa seems to not provide a correct classification, as the context does not contradict the claim. The context does not provide any information about the super bowl, as it talks about the hockey league. An interesting learning from this case is that when doing retrieval based on the claimes, Google Search and the VDB embedding system tend to focus on specific numbers and dates, like February 3, 2019 in this case. This is a potential source of error, as the context might not contain the exact date, but still be relevant to the claim.

### 3.2.4. RTR-System

"Retrieve Then Read" (RTR) systems follow a different workflow but face the same challenges as PHR systems, albeit being arguably less challenging in the context of attribution. A standard RTR system retrieves information solely based on the question, similar to the first retrieval approach compared, and bases its answer explicitly on the retrieved documents. This eliminates one of the key challenges in PHR systems, which is to find relevant documents for claims since all sources are already available when formulating the answer. "Attribution" reduces to back-referencing and entailing the segmented claims of the answer to the already retrieved documents. Everything that cannot be entailed can be viewed as hallucination in strict systems.

However, the advantage due to the elimination of the challenge of retrieving sources to attribute claims appears only at first glance: The challenge of the retrieval process is simply moved to the first step. We have already established that retrieving documents based on the question is ineffective for the task of sourcing relevant claims. Therefore, RTR systems also need to find a workaround for this challenge. One common approach is to use Language Models as Retrievers, where first, a language model is queried to answer the question, and after that, documents are retrieved based on the segmented claims. This approach, at its core, is just a PHR System that provides the user with the answer at a later step of the pipeline instead of directly. Therefore, the same challenges apply as discussed above.

### 3.2.5. Summary & Results

The evaluation of the different systems and methods for the attribution process has shown that the current state of the art is not yet able to provide a reliable and consistent attribution of claims to sources. The main challenges and findings are:

1. **Claim Segmentation**: The current state of claim segmentation does not provide reliable and usable claims for end-to-end pipelines.

2. **Claim Independence**: Claim independence is a crucial factor for the usefulness of a claim, especially in complex domains and end-to-end applications. If claims are not

independent, most current systems fail to provide reliable results, and there exists significant potential for error propagation.

3. **Claim Usefulness**: The determination of the usefulness, relevance, and check-worthiness of a claim present a challenging task for current systems, as they fail to provide the necessary context for evaluating the claims.

4. **Longer Context Window for Retrieval is Better**: A longer context window appears to generally improve the retrieval of meaningful sources.

5. **GPT-3.5 Hallucinates Connections**: GPT-3.5 tends to hallucinate connections between claims and sources, especially when the context window is too short.

6. **DeBERTa vs. GPT-3.5**: DeBERTa and GPT-3.5 differ in their attribution evaluation, where GPT-3.5 seems to be biased towards "Entailment," and DeBERTa towards "No Relation."

7. **RTR Systems**: RTR Systems face the same challenges as PHR Systems, but at a different stage of the pipeline.

8. **Error Propagation**: Error propagation becomes increasingly significant in a series of different approaches, like the one currently under discussion.

## 3.3. Developing Solutions

We are now going to propose systemic solutions for selected key issues identified in the previous chapter. This aims to address Research Question 3, which is formulated as follows:

**RQ3: Improving Attribution and Attribution Evaluation**
3.1 Can we develop an abstract framework to evaluate attribution that is independent of the domain and complexity of the question?
3.2 Can existing attribution evaluation methodologies be adapted to complex answers?
3.3 How might existing metrics be enhanced to accommodate complex QA scenarios?
**Deliverables:** A structural analysis of existing and novel approaches and components for attributing claims.

### 3.3.1. Methodology

The previous section outlined the core challenges in attributing complex answers. Building upon these results, the existing evaluation framework is utilized to modularly exchange individual approaches and assess their effects on the overall system, as previously observed.

### 3.3.2. Claim Segmentation

One of the primary reasons for the weak performance of previous systems was the lack of independence among claims. Even when aiming to create atomic fact based claims, most existing systems fail to provide sufficient context, making it difficult for the claims to stand alone. This leads to significant error propagation and misleading outcomes in the information retrieval process and in evaluating the attribution of a claim. The right sections for the specific claims are not retrieved and the relation between claim and source can not be evaluated. To address this issue, we build upon current approaches and propose an adaptation aimed at resolving this challenge.

There are three different types of claims produced by current systems that require additional context for accurate evaluation:

1. **Anaphoric References (Simple Contextualization)**: Claims that include one or more anaphors referring to previously mentioned entities or concepts. Thus, they cannot be evaluated independently, leading to suboptimal retrieval results. This issue is a direct consequence of relying on sentences that contain anaphors as the sole input.
   **Example**: "The purpose of *these strategies* is to reduce energy consumption.", "*They* ensure the well-being of everyone involved."

2. **Conditioning (Detailed Contextualization)**: Claims that lack entire sentences or conditions necessary for proper contextualization. While not always obvious from the claim itself, this information is crucial for accurately evaluating the claims.
   **Example:** "Chemotherapy is no longer the recommended course of action."

3. **Answer Extracts (Hypothetical Setup)**: Claims that arise specifically from questions describing a hypothetical setup (as described in Section 3.1.4). Current answer systems often replicate aspects of the question in the answer, leading to claims that are specific to the hypothetical scenario posed in the question but are not present in the answer. These claims cannot be evaluated independently.
   **Example:** "A young girl is running in front of cars."

As a solution, we propose a claim segmentation approach designed to provide the necessary context for each claim. There are two distinct strategies to achieve this objective: Firstly, we edit extracted claims to incorporate necessary context from both the answer and the question. A system employing this strategy would implement the function $f_{\text{enrich}}(q, a, c_{\text{non-independent}})$, where $c_{\text{non-independent}}$ is the non-independent claim, $a$ is the answer, and $q$ is the question from which the claim was segmented. Secondly, we suggest a system that directly segments the answer into multiple independent claims, each supplemented with the required context. This system would use the function $f_{\text{segment}}(a, q)$, differing from the initial (as in Section 3.2.3) by incorporating the entire answer and question rather than basing the segmentation on individual sentences. The effectiveness of these systems are assessed subsequently. Our approach builds on previous methods that employ LLMs and prompting techniques (both

| | GPT3.5 | | | GPT4 | | |
|---|---|---|---|---|---|---|
| System | Prec. | Rec. | F1 | Prec. | Rec. | F1 |
| **Overall** | 0.94 | 0.27 | 0.42 | **0.96** | **0.74** | **0.84** |
| gpt35_factscore | 0.93 | 0.29 | 0.44 | 0.95 | 0.75 | 0.84 |
| segment5_propsegment | 0.90 | 0.19 | 0.32 | 0.96 | 0.66 | 0.78 |
| spaCy_sentences | 1.0 | 0.5 | 0.67 | 1.0 | 1.0 | 1.0 |

Table 3.10.: Non-Independence detection performance compared to human evaluation as described in Table 3.5.

zero-shot and few-shot) to accomplish this goal. we compare the performance of GPT-3.5 and GPT-4 in executing this task.

For the first function, it is necessary to determine whether a claim requires additional context. This is achieved using a single-shot prompt that assesses the independence claims, and the results are then compared with human evaluations presented in Table 3.5. The specific prompt used for this assessment is available in the appendix.
Table 3.10 displays the performance scores of GPT-3.5 and GPT-4 in comparison with human evaluation for detecting independence. It is evident that both GPT-3.5 and GPT-4 exhibit significantly high precision in identifying non-independent claims, with GPT-4 outperforming in terms of recall and F1 score, without compromising on precision. We conclude that the independence of claims can definitely be detected by LLMs. Moving forward, we utilize the claims classified as "non-independent" by GPT-4 to assess the performance of the function $f_{\text{enrich}}(q, a, c_{\text{non-independent}})$.
For this purpose, we consider all 2,317 factual and unique claims, as segmented by the GPT-3.5 FactScore [8]-based system, as described in Table 3.6, and evaluate a random sample of 500 for their independence using the GPT-4 based prompt. **290 out of 500** were deemed to be "not independent" by GPT-4. We then apply a single-shot based prompt with both GPT-3.5 and GPT-4 to implement the function $f_{\text{enrich}}(q, a, c_{\text{non-independent}})$ and compare the results to the original claims. The comparison is conducted using the non-independence detection system described above. The quality of this step measured in the reduction of non-independent claims. The results are is presented in Figure 3.16. As observed, the implemented function further reduces the number of non-independent claims by 36.9% for GPT-3.5 as a model and by 41.7% for GPT-4. Although the enrichment significantly aids the process, it still leaves over half of the claims without context. It is notable that the contextualization/enrichment has significantly increased the average number of characters of the claims. Initially, the average number of characters for the independently segmented claims was 66.0, and 59.4 for non-independent claims. The revision by GPT-4 increased this to 155.6 characters, and the enrichment by GPT-3.5 increased it to 145.9 characters. Moving forward, we evaluate the impact of claim enrichment on the retrieval process. The results are discussed in Section 3.3.3.

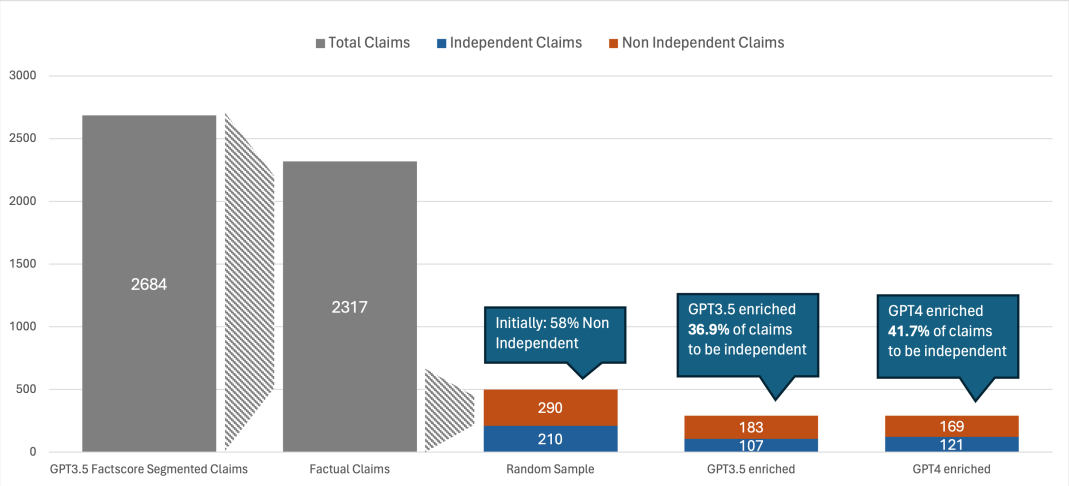


Figure 3.16.: Statistics of contextualization of the 290 created claims by GPT3.5 and GPT4, evaluated by GPT4

### Claim Segmentation - Direct Segmentation

An alternative to enriching non-independent claims is to directly segment the answer into multiple independent claims. This approach implements the function $f_{\text{segment}}(a, q)$ by using a single-shot prompt and GPT3.5 and GPT4 as LLMs. To evaluate the result on a quantified basis, we compare the average number of claims and the length of claims with those from alternative approaches to claim segmentation. This step is executed on all 100 (question, answer) tuples. The prompt requests the model to print out a structured list of claims. The exact prompt can be found in the appendix. The initial quantifications of claim length and number of claims are presented in Table 3.11 and compared to the results presented in Table 3.4. It can be observed that both, GPT3.5 and GPT4 create significantly less claims then both

| Segmentation System | Number of $c$ | Unique #$c$ | avg. len($c$) | $c$ / Sentence |
|---|---|---|---|---|
| **GPT3.5 direct** | 661 | 661 | 94.8 | 0.77 |
| **GPT4 direct** | 841 | 841 | 85.6 | 0.98 |
| spaCy_sentences | 938 | 855 | 103.2 | 1.00 |
| gpt35_factscore | 3016 | 2684 | 61.4 | 3.22 |

Table 3.11.: Descriptive direct claim segmentation comparison.

sentence based segmentation systems, and even spaCy sentence segmentation itself. Since this seems unlikely, because a lot of sentences contain more than one atomic claim, we do a qualitative analysis of the claims and sentences segmented by GPT4 and compare them to the spaCy sentence segmentation.

**Question**: "What is the difference between prions and viruses?"
**Answer**: "Prions and viruses are both infectious agents, but they differ in several key

aspects: 1. Nature: Prions are composed of misfolded proteins,[...].
**SpaCy Sentence Segmentation::**

1. "Prions and viruses are both infectious agents, but they differ in several key aspects:."
2. "1. Nature: "
3. "Prions are composed of misfolded proteins [...]."
4. "2. Replication: "
5. ...

**GPT-3.5 SpaCy Based Segmentation**:

1. "Prions and viruses are both infectious agents."
2. "They differ in several key aspects:"
3. "Nature exists"
4. "Nature encompasses the physical world and all living organisims."
5. ...

**GPT-4 Direct Segmentation**:

1. "Prions consist of misfolded proteins."
2. "Viruses consist of genetic material surrounded by a protein coat."
3. "Prions replicate by inducing misfolding in normal proteins."
4. "Viruses replicate by using the host cell's machinery to produce more viruses."
5. ...
6. "Viruses can cause a variety of diseases, including the common cold, flu, HIV/AIDS, and COVID-19."

We can extract several insights from the example above. First, spaCy segments the answer into 11 unique and non-empty sentences. Three of these sentences consist of singular words, such as "1. Nature" in the given example. Here, the title of a paragraph within the answer is segmented into its individual sentence. GPT-3.5, which processes this input, produces 49 non-empty segments and becomes visibly "confused" by this input, generating completely out-of-context and clearly hallucinated claims, such as "Nature encompasses the physical world and all living organisms." In contrast, direct segmentation by GPT-4 produces 12 claims for this example, which do not contain syntactical artifacts like the segmentations from spaCy. However, as the last example demonstrates, they are not perfectly atomic. Example (6) from the direct segmentation should be further divided into individual atomic claims, for instance, "Viruses can cause the common cold." Additionally, the first sentence of the answer, "Prions and viruses are both infectious agents," is not segmented into an atomic claim by the direct

segmentation.

Based on these findings, we adopt two strategies: Firstly, we modify the prompt for direct segmentation in hopes of achieving better segmentation. Secondly, we utilize a GPT-3.5 FactScore-based evaluation [8] to further segment the claims into atomic claims. The results are presented in the table below (Table 3.12).
The adopted prompt appeared to significantly increase the number of claims generated by

| **Segmentation System** | **Number of *c*** | **Unique #*c*** | **avg. len(*c*)** | ***c* / Sentence** |
|---|---|---|---|---|
| **GPT3.5 direct v2** | 644 | 644 | 102.8 | 0.75 |
| **GPT4 direct v2** | 948 | 948 | 84.1 | 1.11 |
| **Factscore & Direct (4, v2)** | 2494 | 2494 | 61.4 | 2.92 |
| GPT4 direct | 841 | 841 | 85.6 | 0.98 |
| gpt35_factscore | 3016 | 2684 | 61.4 | 3.22 |

Table 3.12.: Descriptive comparison of adopted claim segmentation approaches

GPT-4, resulting in approximately 1.11 claims per sentence and a total of 948 claims, compared to the previously generated 841 claims. The average length of the claims remained consistent. GPT-3.5 performed less effectively with the adopted prompt, generating approximately 20 fewer claims than its predecessor.
Upon applying the FactScore approach to the claims generated by GPT-4, an increase in the number of claims was observed, aligning with the levels obtained through original FactScore segmentation. This implementation aims to diminish non-independence, given that the original FactScore segmentation relied on SpaCy sentences, which exhibited non-independence in 80% of instances. As generating independent claims from non-independent inputs is not possible, employing GPT-4 as a baseline may mitigate this issue.

**Factuality & Independence**

The next step in the evaluation involves analyzing the factuality of the individual claims, employing the same methodology as described in Section 3.2.3, with previous results presented in Table 3.6. The outcomes of the direct claim segmentation are depicted in the figure below. This figure clearly demonstrates an improvement in the factuality rate of the claims generated by both GPT-4 and GPT-3.5 compared to SpaCy sentence segmentation, with the factuality rate increasing from 64.3% to 99.4% for GPT-4 and to 91.9% for GPT-3.5. A comparable enhancement in the GPT-4 Direct + FactScore system relative to the SpaCy + FactScore system is evident, with the factuality rate rising from 87.1% to 93.5%. These results suggest that this approach is a significantly better alternative to SpaCy sentence segmentation.

Finally, we evaluate the impact of the direct claim segmentation on the **independence of the claims**. For that, we compare the number of non-independent claims generated by the different systems, using the same system for evaluating non-independence as described in

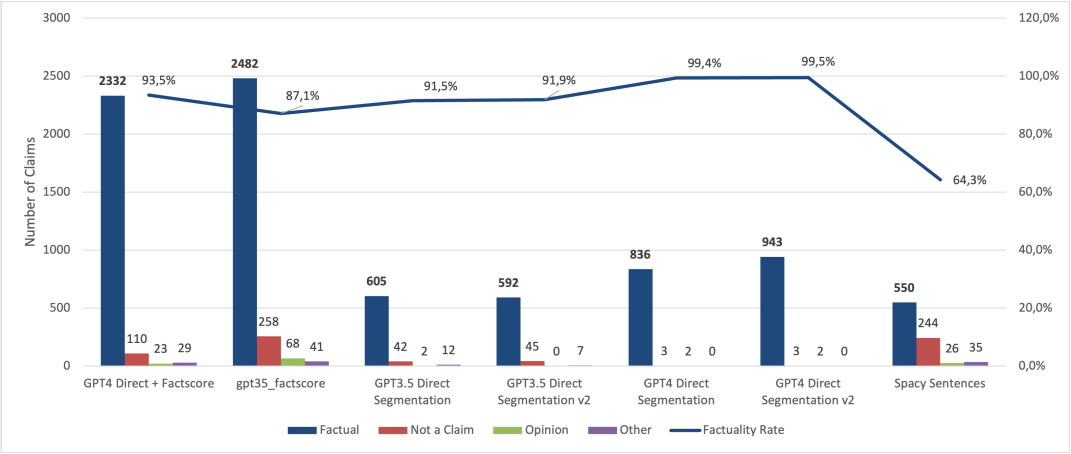


Figure 3.17.: Visualization of the factuality evaluation statistics for the five different systems compared to those in Table 3.6.

Section 3.3.2 and Table 3.10. The results are presented in Figure 3.18. Figure 3.18 demonstrates

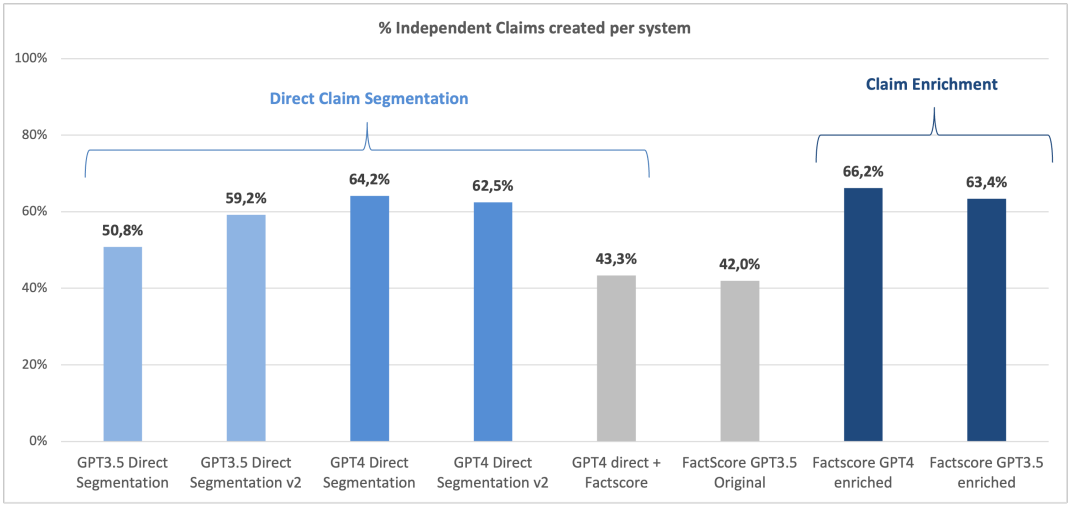


Figure 3.18.: Visualization of the share of created independent claims by all different systems, incldluing the previous evaluation visible in Figure 3.16.

that direct segmentation by GPT-4 has substantially increased the proportion of independent claims, closely matching the performance of the GPT-4 based enrichment. The claims generated by the FactScore & Direct (4, v2) *(label in this figure: GPT4 direct + FactScore)* appear to achieve a similar rate of independence as the original FactScore segmentation, which is based on spaCy sentences. Nevertheless, the enrichment of non-independent claims with GPT-4 emerges as the most effective method, achieving a 66.2% share of independent claims, compared to the 64.2% share achieved by direct segmentation.

### 3.3.3. Impact on the Retrieval Process

The evaluation of the impact of claim enrichment on the retrieval process is conducted using the same 2,317 (question, answer, claim) triplets, which were classified by the GPT-3.5 FactScore system as factual, as in the previous setup. The retrieval process is conducted using the same GPT-3.5 adapted claim-based retrieval system as described in Section 3.2.3.
For assessing the impact of claim enrichment on retrieval (function $f_{\text{enrich}}(q, a, c_{\text{non-independent}})$), we compare a sampled yet stratified set of claims across four categories: **originally independent, originally non-independent, enriched (by GPT4) non-independent, and enriched (by GPT4) independent claims**, as compared in Figure 3.16. The enriched claims are based on the same originally non independent claims, to ensure full comparability. We utilize DeBERTa for evaluating the claim-source relationship. The findings are presented in Table 3.13.

The table reveals several interesting findings: Firstly, it is evident that originally independent claims significantly outperform originally non-independent claims in the retrieval pipeline. Originally independent claims have a combined "Missing + No Relation" share of only 52.2%, indicating that for 47.8% of the claims, the retrieval pipeline was able to identify a connected source. In contrast, originally non-independent claims have a combined "Missing + No Relation" share of 72.3%, meaning that only for 27.7% of the claims, the retrieval pipeline could find a connected source. Upon enriching the originally non-independent claims with GPT-4, as described in the previous section (function $f_{\text{enrich}}(q, a, c_{\text{non-independent}})$), the claims that were successfully enriched show a notable improvement in performance within the retrieval pipeline. This indicates that enriching (contextualizing) claims, as proposed in Section 3.3.2, significantly enhances retrieval performance.
The successfully enriched claims approach the performance of the originally independent claims, with a combined "Missing + No Relation" share of 58.6%. However, claims that were not successfully enriched exhibit worse performance than the originally non-independent claims, with a combined "Missing + No Relation" share of 78.2%. Another observation is that enriching the claim increases the share of queries that produce "Missing Information", meaning that the Google Serach Query Builder prompt can't handle the information and puts out an error message. These error messages can be due to server issues as well and are therefore, at times, disregardable.
**The net-effect of claim enrichment is a 7.8 percentage point reduction of claim-source pairs with no relation.**

| Model | Contradiction | Entailment | Missing | No Relation |
|---|---|---|---|---|
| Enriched - **Independent** | 6.1% | 35.4% | 4.9% | **53.7%** |
| Enriched - **Not-Ind.** | 1.3% | 20.5% | 7.7% | **70.5%** |
| Originally **Independent** | 5.6% | 42.2% | 0.0% | **52.2%** |
| Originally **Non-Ind.** | 3.6% | 24.1% | 2.4% | **69.9%** |

Table 3.13.: Comparison of claim enrichment on the retrieval performance.

**Impact of Direct Segmentation**

Additionally, we evaluate the impact of the direct claim segmentation on the retrieval process. For that, we use the a random sample of 40 (question, answer, claim) triplets per direct segmentation system, as described in Section 3.3.2. We focus on the three systems of GPT3.5 direct v2, GPT4 direct v2, and FactScore & Direct (4, v2). The results are presented in Table 3.14. As above, we analyze the share of (claim, source) pairs that are classified as "Missing" or "No Relation" by DeBERTa, where a lower share is a sign of a better retrieval process. The pipeline setup is left as described above.

The table demonstrates another notable enhancement in comparison to the enriched claims

| Model | Contradiction | Entailment | Missing | No Relation |
|---|---|---|---|---|
| GPT3.5 Direct v2 - **Independent** | 4.2% | 47.2% | 0% | **48.6%** |
| GPT3.5 Direct v2 - **Not-Ind.** | 0% | 27.0% | 2.7% | **70.3%** |
| GPT4 Direct v2 - **Independent** | 0% | 51.5% | 0% | **48.5%** |
| GPT4 Direct v2 - **Not-Ind.** | 2.0% | 14.3% | 2.0% | **81.6%** |
| FactScore & Direct - **Independent** | 1.3% | 40.0% | 0% | **58.7%** |
| FactScore & Direct - **Not-Ind.** | 0% | 24.4% | 0% | **75.6%** |
| Original **Independent** | 5.6% | 42.2% | 0.0% | **52.2%** |
| Original **Non-Ind.** | 3.6% | 24.1% | 2.4% | **69.9%** |

Table 3.14.: Comparison of direct claim segmentation on the retrieval performance.

for the direct segmentation systems. Direct segmentation by GPT-4 records a combined "Missing + No Relation" share of 48.5% for independent claims and 81.6% for non-independent claims. This represents a significant improvement for independent claims compared to both enriched and original claims. Conversely, the combination of FactScore and direct segmentation by GPT-4 exhibits a reduced effectiveness in retrieval, showing a macro (weighted average between independent and non-independent claims with no-relation) share of 65.0% for (claim, source) pairs with no relation.

To summarize the findings, it can be concluded that direct segmentation by GPT-4 significantly surpasses both the original and enriched claims and outperforms comparative methods in aspects of retrieval, time efficiency, and independent claim generation. It nearly matches the performance of GPT-4 in enriching non-independent claims regarding the creation of independent claims and surpasses it in the retrieval process at the macro level.

### 3.3.4. Retrieval Process

As a final step, we evaluate the retrieval process itself, analyzing different embedding models, context window sizes, and their comparisons. We utilize claims generated by GPT-4 Direct v2, as this system has been evaluated as the best so far. We modify two dimensions of the retrieval process: the embedding model and the context window splitter. Instead of SentenceBert, we

employ OpenAI Ada 2.0, which provides embeddings from GPT-3.5, and AnglE-Embeddings [99] from a Hugging Face pretrained model optimized for retrieval [**li2023angle**]. Rather than using a simple sliding window approach, we implement a recursive text splitter with overlap to capture more relevant information.
The search engine (Google Search Custom Search Engine) and query builder (based on GPT-3.5) remain unchanged. Consequently, each method compares on the same corpus of information and simulates a standard retrieval method. The results are presented in Table 3.15.
The results in the table (Table 3.15) demonstrate that the ADA2.0 Embeddings with the fixed

| Model | Contradiction | Entailment | Missing | No Relation |
|---|---|---|---|---|
| Ada 2.0 | 2.9% | 41.0% | 0% | **56.0%** |
| AnglE | 2.9% | 39.5% | 0% | **57.5%** |
| SBert + Rec. CW | 0.0% | 22.1% | 0% | **76.1%** |
| SBert Baseline (Macro) 3.14 | 0.9% | 35.7% | 0% | **62.5%** |

Table 3.15.: Comparison of different embedding models and context window splitters on the retrieval performance.

512c-size context window splitter outperform the overall Sentence Bert baseline, which was used in the above evaluations and depicted the best performance. The AnglE-Embeddings, which are optimized for retrieval, also outperform the Sentence Bert baseline but fall behind the GPT based Ada2.0 embeddings from OpenAi.
Interestingly the recursive context window splitter with Sentence Bert embeddings does perform significantly worse than the fixed context window splitter. This is especially interesting as this approach was expected to capture more meaningful sections of the source documents.

Using this optimized pipeline and the above results, we now evaluate the impact of the created taxonomy, as discussed in Section 3.1.4, on the retrieval process. The results are presented in Figure 3.19. The figure demonstrates that the retrieval pipeline efficiently retrieves information for claims formulated within taxonomy category 1, "Directed Questions," for about 56% of the (claim, source) tuples, showcasing the best performance for these instances. This outcome is expected, as these questions are more factual and tend to be answerable by individual claims, thereby simplifying the retrieval of connected information. For all other user needs, the proportion of "No Relation" categorizations significantly increases, suggesting that it becomes more challenging for the retrieval pipeline to identify related sources. Particularly, user needs for *Prediction / Consequence Analysis* and *Comparison* tend to produce challenging and non-attributable claims, with a "No Relation" rate of 71.4% and 62.5%, respectively.

### 3.3.5. Summary & Results

We can summarize the results and finding of the developed solutions as follows:

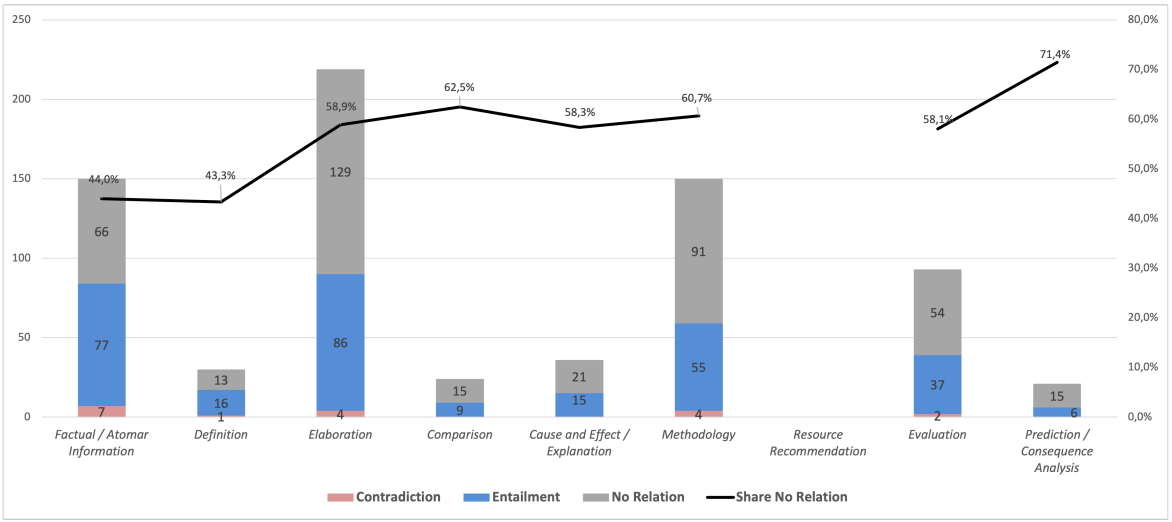


Figure 3.19.: Visualization of user need-based claim evaluation for (claim, source) tuples generated by the optimized retrieval pipeline, using GPT-4 Direct v2 segmentation and Ada 2.0 embeddings.

1. **Claim Independency**: Non-Independent claims perform significantly worse on the existing retrieval pipeline than independent claims. Contextualization is expected to increase retrieval performance.

2. **Non-Independent Claim Enrichment**: Classifying and enriching non-independent claims with GPT-4 significantly improves the performance of the retrieval process. The net-effect of claim enrichment is a 7.8 percentage point reduction of claim-source pairs with no relation. GPT3.5 also improves the net-macro performance of the retrieval process, but to a lesser extent.

3. **Direct Claim Segmentation**: Direct Claim Segmentation using GPT4 outperforms the original and enriched claims and outperforms comparative methods in aspects of retrieval, time efficiency, and independent claim generation. It nearly matches the performance of GPT-4 in enriching non-independent claims regarding the creation of independent claims and surpasses it in the retrieval process at the macro level. The only concern is the exhaustiveness of the segmented claims in relation to the answer.

4. **Retrieval Process**: Using generally better performing embedding models and context window splitters, the retrieval process can be significantly improved. The Ada2.0 embeddings with a fixed context window splitter outperform the overall Sentence Bert baseline, which was used in the previous evaluation and reference papers [65].

5. **User Need Impact on Claims**: The user need has significant impact on the retrieval process, making it easier for the pipeline to identify related sources for factual and directed questions, and more challenging for the retrieval pipeline to identify related sources for prediction and comparison questions.

## 3.4. Evaluation of Domain Dependency

This sections aims to analyse the impact of specific domains and think about adaptation to enterprise-level domains which are not present in the initial dataset.

**RQ4: Cross-Domain Performance**
4.1 Which domains have similarities to the questions cataloged initially?
4.2 Which domains vary significantly from those defined in RQ1?
4.3 How do our attribution and attribution evaluation methods fare across these various domains?
**Deliverables:** A structural analysis of the developed attribution aspects in the context of specific domains.

### 3.4.1. Domain Analysis

The initial dataset of 100 questions was constructed based on ExpertQA [59] and NaturalQuestions [41]. ExpertQA focuses on expert-level questions from specific domains, leading to an implicit analysis of questions within these areas. The domains included in the initial dataset are diverse, facilitating a comprehensive analysis of attribution methods. However, these domains are not exhaustive and do not encompass all potential areas, particularly those not structurally equivalent, such as code, enterprise-specific knowledge, or specific documents. First, we analyze the user need and question structure by domain. Then, we evaluate the performance of the attribution methods across these domains. ExpertQA's general domain distribution encompasses a total of 31 specific knowledge/expert domains. The domain distribution within NaturalQuestions is not specified.
Based on these domains, the Figure 3.20 illustrates the distribution of user needs by domain. The domain with the highest number of questions is "Healthcare and Medicine", consistent with ExpertQA's initial distribution. Every field for Expert based question in the dataset has a relatively comparable share of user needs that request elaboration, making this user the a baseline category. Differences occur in other user need categories, like "Cause and Effect / Explanation" and "Methodology". Those user needs are primarily requested in the domains of "Healthcare and Medicine", as well as "Engineering and Technology".
The distribution claim categorization follows a similar pattern. Figure 3.21 displays this. From this figure, we derive insights into the attributability of each domain. By using claims created through direct segmentation with GPT-4 (v2) and the retrieval process with Ada embeddings, we observe that the Healthcare domain is well-supported by sources, with only 41% of claims remaining unattributed. In contrast, the "Engineering and Technology" domain exhibits approximately 61% of non-attributable claims, indicating greater challenges and less well-structured online sources for these domains. This is understandable, as the healthcare domain is more established and has a broader online presence, whereas the Engineering and Technology domain is more fragmented and lacks centralized sources.

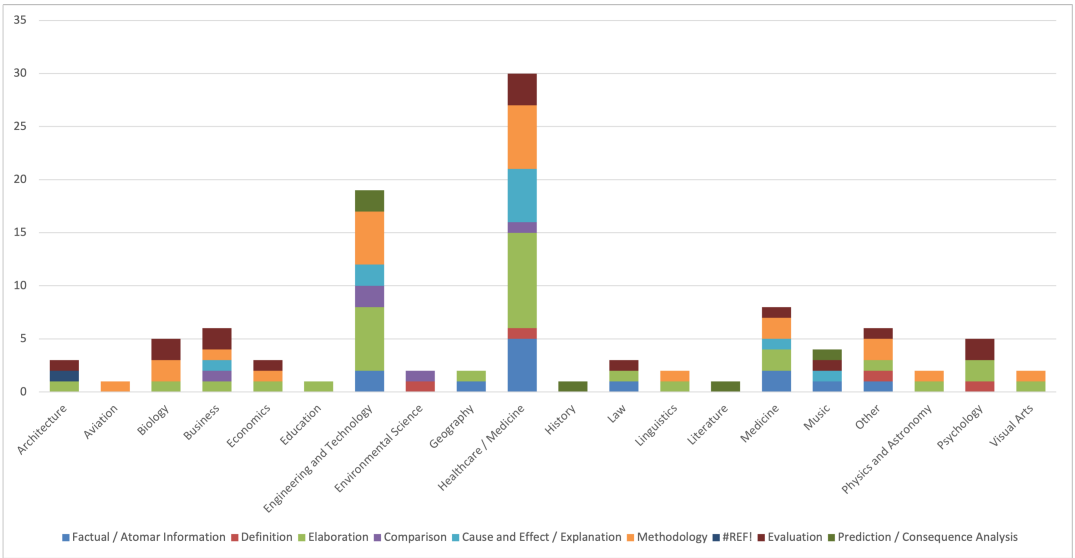


Figure 3.20.: Distribution of user needs per domain based on ExpertQA [59] for the 100 questions in the dataset.

The insights provided above clearly illustrate the critical importance of having well-structured and accessible data for attributing the claims of a LLM. As suggested by the findings from other research questions, retrieving related information poses the greatest challenge for attribution. It is evident that claims within well-structured and accessible sources are more easily attributed.

### 3.4.2. Implications on Enterprise Domains

We define enterprise domains as domains that present specific challenges not found in most well-defined research-based datasets. While research datasets are well-structured and biased towards practical use cases, business domains often entail more complexity and necessitate a more adaptable approach. The primary challenges for attribution in enterprise domains include:

- **Enterprise Knowledge**: Enterprise knowledge is not publicly accessible and is poorly structured, complicating the training of models on this data. As a result, a purely pre-trained heuristic-based (PHR) approach may not be viable, since no pre-trained model can offer relevant information for a specific enterprise. Consequently, a model may either be fine-tuned on enterprise data or rely on a rule-based retrieval (RTR) approach, which typically underperforms compared to PHR approaches [59].
- **Process Automation**: LLMs and queries against them are frequently used for complex automation tasks that require an in-depth, natural language-based understanding. In this context, attributing the responses of LLMs is crucial for enterprises to validate certain automation processes. For example, a company may use a LLM to automate

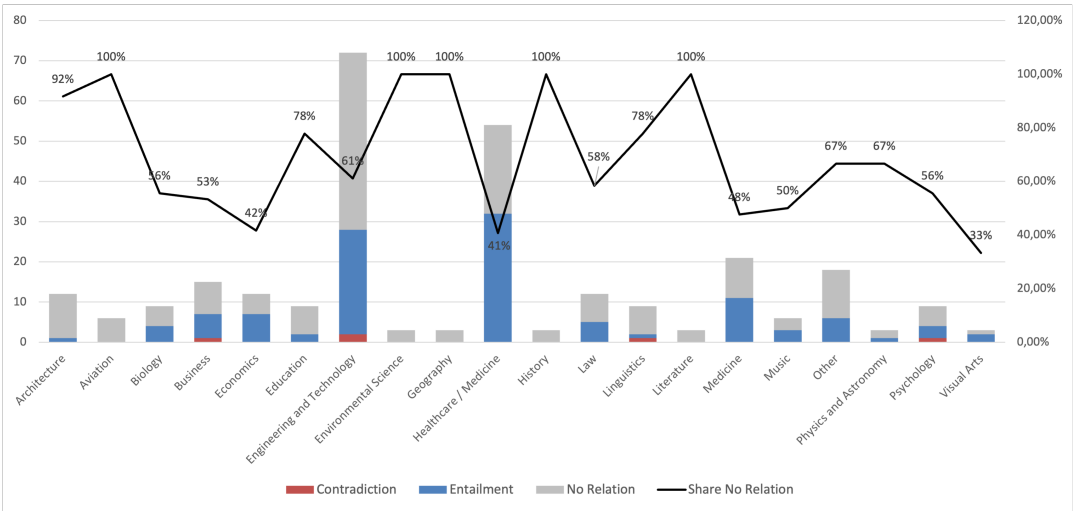


Figure 3.21.: Distribution of claim classification based on the respective field from ExpertQA.

the migration of contracts from one system to another. In this case, the information extracted from the LLM must be attributed to the original contract, which may be in PDF or another non-structured format.

- **Input Data Structure**: Although this thesis concentrates on attribution for language models, recent developments in foundational models indicate a significant shift towards multimodal models. Attributing responses from multimodal models introduces numerous new challenges to attribution, as referencing specific aspects of the input data becomes more complex compared to language models. Code is another example of complex and different input data that requires different approaches. Core components of regular attribution pipelines, such as the final attribution evaluation or the retrieval itself, may need to be adapted to accommodate code as in- and output.

Some of the issues addressed were previously discussed in the context of the taxonomy and the dataset. For instance, enterprise knowledge could be utilized to initially fine-tune a Large Language Model for the domain. Since it is not feasible to regularly fine-tune LLMs, the model could subsequently be employed in a LLM-as-Retriever based Retrieval-Augmented Generation (RAG) approach. This would enhance the retrieval process by enabling higher quality retrieval based on claims, which can then be more easily matched with an up-to-date knowledge base.
In general, although the context or specific application of enterprise-level attribution can sometimes differ, the findings and results can be considered as universal and can be adapted to the use case accordingly.

# 4. Discussion

This chapter aims to discuss and interpret the results from previous chapters within the context of their implications, limitations, and possible applications. To do so, we examine our approach and findings, particularly the attribution pipeline, as proposed in Chapter 3.

## 4.1. General

Two general points are worth to put emphasis on in relation to the results.
Firstly, our analysis predominantly utilized GPT-3.5 and GPT-4, which, while not always the top-performing models for certain tasks, are among the most widely used. Secondly, the framework was architected to facilitate easy substitution of the endpoint model. The real value of the framework is in its ability to adopt within the prompts and the pipeline setup. This is particularly important, given the rapid pace of development in the field of LLMs.
Additionally, in most tasks where a decision between specific LLMs was required, GPT-4 significantly outperformed GPT-3.5 in the majority of cases. Given that GPT-4 is the later and more advanced model, this outcome is expected. In general, it is visible across a vast amount of tasks and current research, that using the "best" or most advanced foundational LLM produces the best results across the board. However, not all tasks could be completely addressed by the more advanced model, suggesting the need for specialized solutions.

The overall costs for all API calls are outlined in the appendix.

## 4.2. RQ1: Research, Taxonomy, and Dataset

We demonstrated, by comparing multiple Q&A datasets with one specifically designed for LLMs, that the nature of information needs and the manner in which questions are posed to automated answering systems have significantly evolved due to LLMs.
The taxonomy provided a valuable distinction between different types of information needs, which had a measurable impact on various steps of the pipeline. Utilizing multi-label classification, we ensured that there were no questions whose user needs were not covered by the taxonomy, making it comprehensive from the current perspective. It streamlines its intent towards being a user-need-focused taxonomy with a secondary dimension for question structure, eliminating inconsistencies found in previous taxonomies. Additionally, redundant categories for user needs, such as "summarization", have been removed as well. A notable limitation of the taxonomy is its restriction to singular interactions, i.e., single (question, answer) pairs. The reality of current LLM-based Q&A systems appears to be shifting towards more

complex, conversational interactions, which our framework does not address. Additionally, the final Cohen-Kappa scores after revision are still below 0.5 for some categories, as seen in Figure 3.6. This can be attributed to the inherent subjectivity of the task and the general ambiguity of language and user needs. Ultimately, only the user knows exactly what they are seeking, and sometimes not even they are certain. Overall, the taxonomy provides a valuable framework for differentiating and evaluating attribution models but could be extended to accommodate approaches focusing on conversational interactions or user needs that are more structural and process-oriented, like information extraction from a document.

The question structure categories are also well-defined and account for over half of all complex questions directed at LLMs. This is particularly noteworthy, as most question structure-based taxonomies deploy a significantly more detailed set of categories, which we funnel into the question structure of "other" questions. This means that the question structure for queries in the context of LLMs has indeed changed significantly, and the taxonomy reflects that with the two categories of "Hypothetical Setup" question structures and "Follow-Up" question structures.

The dataset structure was well used during the evaluation of different approaches, providing a standardized framework for comparison and versioning. It reflects the possible variations for sub tasks during the attribution process and can be flexibly extended to accommodate new approaches. The data itself, sourcing from 70 questions from ExperQA [59] and 30 questions from Natural Questions [41], represent a variety of domains due to the nature of ExpertQA and user needs. Yet, due to its relatively small size of only 100 questions, it can be seen as heavily biased in content and structure. This limitation can be adressed by adding additional questions from other sources, in the optimal case queries or questions which are directed at LLMs.

## 4.3. RQ2: Evaluation of Attribution Approaches

The results for the evaluation of different attribution methods were presented in Section 3.2.5. A key interpretation from these findings is that, given the complexity of the query or user need, the main challenge lies not in linking the answer from a LLM to a given source but in the retrieval of the source itself. Current systems are advanced enough that, assuming a source exists for every factual and independent claim on the internet, the primary task is to retrieve and evaluate this specific source, rather than assessing the relationship between the source and the claim. This is evident in the comparison between question-based retrieval approaches and claim-based retrieval. Both implicitly refer to the same database — the internet — but yield remarkably different results for the final evaluation, with claim-based retrieval significantly outperforming question-based retrieval.
Furthermore, even within the relatively narrow scope of retrieving the most relevant segments from the top 3 retrieved websites for a given claim, we observe notable differences based on the embedding space and context window used to map the claim to the source. Resulting

challenges therefore are the contextualization of a claim such that it performs best against a given database of source documents, and the large-scale retrieval of the source documents given the same claim. This task is, in its core, hierarchical retrieval based on the same query using two different retrieval systems.

Prior to the retrieval and evaluation of sources, the segmentation of answers into individual claims and the assessment of those claims were conducted. The findings in this section highlighted significant shortcomings in terms of claim independence and the quality of the segmented claims, issues that were addressed within Research Question 3. An aspect not explored in this thesis is the segmentation of claims into larger and more semantically connected sections, as proposed by Min et al. [8], instead of segmenting into atomic facts. For some domains, where a majority of information might be expected to reside within a single document, this approach could prove more beneficial. Additionally, most Rule-Based Retrieval (RTR)-based systems require the attribution of longer, non-atomic claims since the answers are expected to either replicate or at most, rephrase the retrieved input.

Even though this finding may seem surprising, given that most literature has already proposed solutions for automatically segmenting claims, it is inherently sensible to pay focused attention to this area. Comparative studies did not address these shortcomings, as their datasets were novel and the final claim creation was refined by human annotators. **Claims ultimately determine what actually needs to be attributed and therefore act as the query for the retrieval process. They determine the output and quality of the entire attribution pipeline.**

## 4.4. RQ3: Developing Solutions

Within the section for RQ3, we took the pain points of the evaluation that were determined to be the most crucial to fix and developed and compared novel approaches. The first adressed issues was the non-independence of segmented claims, which was assumed to lead to a significantly worse performance. This assumption was proven to be right, as the developed solutions show. The best model used for evaluation and creation was based on GPT4. Using GPT4 for creation and evaluation could potentially induce a bias into the claim-independence detection system (Section 3.3.2). For future work, it would be advantageous to fine-tune a model that can be deployed locally for the detection of non-independent (non-contextualized) claims or to utilize two distinct systems. A bias from training data appears unlikely since the training data from ExpertQA was released after the versions of GPT-3.5 and GPT-4 that were used.
Moreover, as previously mentioned, segmenting answers into **independent (contextualized)** claims was most effectively done using GPT-4, yet it did not achieve an 80% success rate. This indicates that a generalized language model might not be the best choice for this task and could be outperformed by a more specialized and smaller model designed specifically for this purpose.

Regarding the retrieval process, we have highlighted the importance of a well-adjusted pipeline, including embeddings and context windows for source document segmentation. Future research should explore additional embedding models and, in particular, Vector Database (VDB) systems, as our comparison was limited to three embedding systems and one VDB indexer (FAISS, [95]). Since this component is relatively independent from the rest of the pipeline and has already been extensively researched, replacing the retrieval model should be comparatively straightforward.

## 4.5. RQ4: Domain Comparison

The domain comparison was conducted using the created dataset and the developed pipeline. This research question points to the least explored applications of the developed attribution pipeline. The findings indicate that attribution's effectiveness varies across expert-based domains, significantly depending on well-established and easily accessible internet corpora for data retrieval.
It is important to note the potential bias in the dataset, primarily derived from ExpertQA, and thus based on its preselected domains.

Overall, the framework and pipeline created for attribution facilitate easy and modular adaptations to other structurally distinct domains, outperforming existing comparative solutions regardless of the used LLM.

# 5. Conclusion & Outlook

To conclude this thesis, we provide a high-level summary and an outlook on potential future work. A more detailed summary can be found at the end of every section within the main body.

## 5.1. Conclusion

In the "Related Work & Background" (chapter 2), we evaluated the current state of research and related work significant to this thesis. Reviewing the related work also enabled us to establish boundaries and initial definitions for the comparative work and approaches developed in Sections 3.1 and 3.3 of the Main Part (chapter 3). This effort includes the novel definition of aspects of attribution and precise formulations of subtasks involved in attribution. Based on this background work, we created a new taxonomy for user information needs, evaluated existing approaches for attribution, identified weaknesses, developed solutions for those weaknesses, and analyzed the results considering different domains and the initially created taxonomy.

We conclude that the newly developed two-dimensional taxonomy for user need and question structure offers a more well-defined and relevant framework for classifying interactions with large language models than existing taxonomies. It has significant impact on the answers and attribution process of LLMs and therefore provides a valuable distinction as a basis for analysis and tackling specific challenges. However, ambiguity persists due to the complexity of language itself, which further research could potentially reduce.
Furthermore, we conclude that the main challenge for automatic attribution remains the **retrieval of relevant information**. Especially in complex domains, such as expert-level Q&A settings, retrieving verified information from external and dynamic corpora poses a significant challenge. This is revealed in Section 3.2 of the Main Part (chapter 3), where the performance of existing approaches and subtasks is evaluated. Bad retrieval performance makes attribution impossible. The biggest identified leverage for improving retrieval was recognized to be the **quality of the segmented claims**. This was improved in the following section, comparing multiple approaches to creating independent and high quality claims.

Overall, the research conducted and the solutions developed represent a significant contribution to improving and understanding attribution across multiple domains. As LLMs and foundational models continue to grow in usage, their impact on society and the economy expands accordingly. The enhancement in the performance of natural language understanding

and reasoning of these models allows us to concentrate on retrieving relevant information to fact-check the output. Moreover, it enables the utilization of their advanced natural language understanding capabilities to self-assess the quality of LLM outputs. Hallucination is inevitable, and therefore attribution.

## 5.2. Outlook

Multiple areas of future work have been identified based on the results and the limitations of this thesis. The most significant areas include:

1. **Increasing the Dataset Size**: The current dataset is limited to 100 questions from two existing datasets. This number should be increased by incorporating more diverse queries, ideally focused on usage in the context of LLMs.

2. **Increasing the Dataset Domain Variety**: The current dataset is limited to specific domains. A natural extension would be to include various expert knowledge-based domains from Malaviya et al. [59].

3. **User Need - Taxonomy Evaluation**: The taxonomy was found to be mutually exclusive and collectively exhaustive (MECE) for the created dataset, but this does not guarantee a well-defined structure on a larger scale. The taxonomy should be evaluated more extensively.

4. **Question Structure - Taxonomy Extension**: The taxonomy created for question structure has shown significant value for this thesis but could potentially be extended and evaluated on a larger scale.

5. **Fine-Tuned Model for Independent Claim Segmentation**: The proposed method for answer segmentation into contextualized claims has proven to be a significant improvement over existing methods. However, this insight could be used to fine-tune existing models with human-annotated data for this specific task to further increase performance in claim segmentation.

6. **Detailed Claim Relevance Evaluation**: We utilized an existing approach for claim relevance evaluation proposed in FactcheckGPT [65]. This step could potentially be improved to reduce the number of unnecessary researched claims.

7. **Retrieval Performance Improvement**: The retrieval performance relies on two relatively standard yet challenging approaches that have been extensively researched and developed over the years. The performance of attribution directly benefits from enhanced retrieval performance, especially in the context of Vector Space Models.

8. **Enterprise Domains and Use Cases**: As described in Section 3.4.2, exploring additional enterprise domains presents an interesting field of research with its own challenges, representing a potential area of future work.

# A. General Addenda

The entire code used for producing the results within this thesis is available at: `https://github.com/LucaM1/NLP_MT`.

The reference commit is `5c778a9`. Note that later commits might contain clean-up and additional comments.

The code is written in python. All reference libraries used can be found in the `pyproject.toml` file. The structure of the repository allows for a simple set up using `poetry`.

The sum of the Microsoft Azure API and Google Cloud Services API costs for all results created in this theses approximates 300,00€. This includes Ada 2.0 Embeddings, GPT4 input and completion tokens (90% of the cost), GPT3.5 input and completion tokens and Google Cloud Programmable Search Engine Paid API costs.

## A.1. Prompts for Subtasks

Direct Claim Segmentation v2:

```
Objective: Transform the answer to a question into it's discrete, fundamental claims. Eac

Conciseness: Formulate each claim as a brief, standalone sentence.
Atomicity: Ensure that each claim represents a single fact or statement, requiring no fur
Independence: Craft each claim to be verifiable on its own, devoid of reliance on additio
Consistency in Terminology: Utilize language and terms that reflect the original question
Non-reliance: Design each claim to be independent from other claims, eliminating sequenti
Exhaustiveness: Ensure that the claims cover all the relevant information in the answer,

Strictly stick to the below output format, which numbers any claims and separates them by
Don't add any explanation or commentary to the output.
```

All additional prompts and few-shot prompt examples used for the subtasks can be found in the `data/prompts` folder of the repository. The file names are self-explanatory.

# List of Figures

# List of Tables

# Bibliography


[1] T. B. Brown, B. Mann, +. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165.

[2] OpenAI, : J. Achiam, et al. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[3] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, S. T. Truong, S. Arora, M. Mazeika, D. Hendrycks, Z. Lin, Y. Cheng, S. Koyejo, D. Song, and B. Li. *DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models*. 2024. arXiv: 2306.11698 [cs.CL].

[4] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz. "Capabilities of gpt-4 on medical challenge problems". In: *arXiv preprint arXiv:2303.13375* (2023).

[5] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann. "BloombergGPT: A Large Language Model for Finance". In: *ArXiv* abs/2303.17564 (2023). URL: https://api.semanticscholar.org/CorpusID:257833842.

[6] P. P. Ray. "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope". In: *Internet of Things and Cyber-Physical Systems* (2023). URL: https://api.semanticscholar.org/CorpusID:258157875.

[7] N. F. Liu, T. Zhang, and P. Liang. *Evaluating Verifiability in Generative Search Engines*. 2023. arXiv: 2304.09848 [cs.CL].

[8] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi. *FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation*. 2023. arXiv: 2305.14251 [cs.CL].

[9] M. Lewis. "Generative Artificial Intelligence and Copyright Current Issues". In: *Morgan Lewis LawFlash* (Mar. 2023). URL: https://www.morganlewis.com/pubs/2023/03/generative-artificial-intelligence-and-copyright-current-issues.

[10] M. M. Grynbaum and R. Mac. "The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work". In: *The New York Times* (Dec. 2023). URL: https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

[11] H. Xu, B. Liu, L. Shu, and P. S. Yu. "BERT post-training for review reading comprehension and aspect-based sentiment analysis". In: *arXiv preprint arXiv:1904.02232* (2019).

[12] H. Yan, B. Deng, X. Li, and X. Qiu. *TENER: Adapting Transformer Encoder for Named Entity Recognition*. 2019. arXiv: 1911.04474 [cs.CL].

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[15] P. He, X. Liu, J. Gao, and W. Chen. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. arXiv: 2006.03654 [cs.CL].

[16] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2020. arXiv: 1906.08237 [cs.CL].

[17] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. 2020. arXiv: 2003.10555 [cs.CL].

[18] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. "MTEB: Massive Text Embedding Benchmark". In: *arXiv preprint arXiv:2210.07316* (2022). DOI: 10.48550/ARXIV.2210.07316. URL: https://arxiv.org/abs/2210.07316.

[19] H. Liu, R. Ning, Z. Teng, J. Liu, Q. Zhou, and Y. Zhang. *Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4*. 2023. arXiv: 2304.03439 [cs.CL].

[20] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, et al. "A comparative study on transformer vs rnn in speech applications". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2019, pp. 449–456.

[21] S. M. Lakew, M. Cettolo, and M. Federico. "A comparison of transformer and recurrent neural networks on multilingual neural machine translation". In: *arXiv preprint arXiv:1806.06957* (2018).

[22] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney. "A comparison of transformer and lstm encoder decoder models for asr". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2019, pp. 8–15.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[24] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. "Improving language understanding by generative pre-training". In: (2018).

[25] R. Lowe and J. Leike. *Aligning Language Models To Follow Instructions*. Accessed: 2024-01-14. Jan. 2022. URL: https://openai.com/research/instruction-following.

[26] D. Amodei, P. Christiano, and A. Ray. *Learning from Human Preferences*. https://openai.com/research/learning-from-human-preferences. June 2017.

[27] W. Knight. "OpenAI's CEO Says the Age of Giant AI Models Is Already Over". In: *Wired* (Apr. 2023). URL: https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/.

[28] M. A. Team. "Mixtral of experts". In: *Mistral AI | Open-weight models* (Dec. 2023). URL: https://mistral.ai/news/mixtral-of-experts/.

[29] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer". In: *arXiv preprint arXiv:1701.06538* (2017).

[30] M. Artetxe, S. Bhosale, N. Goyal, T. Mihaylov, M. Ott, S. Shleifer, X. V. Lin, J. Du, S. Iyer, R. Pasunuru, et al. "Efficient large scale language modeling with mixtures of experts". In: *arXiv preprint arXiv:2112.10684* (2021).

[31] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. *A large annotated corpus for learning natural language inference*. 2015. arXiv: 1508.05326 [cs.CL].

[32] A. Williams, N. Nangia, and S. R. Bowman. *A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference*. 2018. arXiv: 1704.05426 [cs.CL].

[33] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela. *Adversarial NLI: A New Benchmark for Natural Language Understanding*. 2020. arXiv: 1910.14599 [cs.CL].

[34] M. Laurer, W. Van Atteveldt, A. Casas, and K. Welbers. "Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and BERT-NLI". In: *Political Analysis* 32.1 (2024), pp. 84–100.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG].

[36] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma. *Entailment as Few-Shot Learner*. 2021. arXiv: 2104.14690 [cs.CL].

[37] A. Bouziane, D. Bouchiha, N. Doumi, and M. Malki. "Question answering systems: survey and trends". In: *Procedia Computer Science* 73 (2015), pp. 366–375.

[38] D. Chen and W.-t. Yih. "Open-domain question answering". In: *Proceedings of the 58th annual meeting of the association for computational linguistics: tutorial abstracts*. 2020, pp. 34–37.

[39] T. Shen, G. Long, X. Geng, C. Tao, T. Zhou, and D. Jiang. "Large Language Models are Strong Zero-Shot Retriever". In: *arXiv preprint arXiv:2304.14233* (2023).

[40] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. 2016. arXiv: 1606.05250 [cs.CL].

[41] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. "Natural questions: a benchmark for question answering research". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 453–466.

[42] S. Li, R. Li, and V. Peng. *Ensemble ALBERT on SQuAD 2.0*. 2021. arXiv: 2110.09665 [cs.CL].

[43] Z. Xu, S. Jain, and M. Kankanhalli. *Hallucination is Inevitable: An Innate Limitation of Large Language Models*. 2024. arXiv: 2401.11817 [cs.CL].

[44] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 2023. arXiv: 2309.01219 [cs.CL].

[45] R. Friel and A. Sanyal. "Chainpoll: A high efficacy method for llm hallucination detection". In: *arXiv preprint arXiv:2310.18344* (2023).

[46] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (Mar. 2023), pp. 1–38. ISSN: 1557-7341. DOI: 10.1145/3571730. URL: http://dx.doi.org/10.1145/3571730.

[47] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. 2023. arXiv: 2311.05232 [cs.CL].

[48] A. Pagnoni, V. Balachandran, and Y. Tsvetkov. *Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics*. 2021. arXiv: 2104.13346 [cs.CL].

[49] Y. Xiao and W. Y. Wang. "On Hallucination and Predictive Uncertainty in Conditional Language Generation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by P. Merlo, J. Tiedemann, and R. Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 2734–2744. DOI: 10.18653/v1/2021.eacl-main.236. URL: https://aclanthology.org/2021.eacl-main.236.

[50] N. M. Guerreiro, D. M. Alves, J. Waldendorf, B. Haddow, A. Birch, P. Colombo, and A. F. Martins. "Hallucinations in large multilingual translation models". In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 1500–1517.

[51] L. van der Poel, R. Cotterell, and C. Meister. "Mutual Information Alleviates Hallucinations in Abstractive Summarization". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Goldberg, Z. Kozareva, and Y. Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5956–5965. DOI: 10.18653/v1/2022.emnlp-main.399. URL: https://aclanthology.org/2022.emnlp-main.399.

[52] N. Miao, Y. W. Teh, and T. Rainforth. *SelfCheck: Using LLMs to Zero-Shot Check Their Own Step-by-Step Reasoning*. 2023. arXiv: 2308.00436 [cs.AI].

[53] S. Chen, Y. Zhao, J. Zhang, I.-C. Chern, S. Gao, P. Liu, and J. He. *FELM: Benchmarking Factuality Evaluation of Large Language Models*. 2023. arXiv: 2310.00741 [cs.CL].

[54] S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das. *A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models*. 2024. arXiv: 2401.01313 [cs.CL].

[55] A. Köksal, R. Aksitov, and C.-C. Chang. *Hallucination Augmented Recitations for Language Models*. 2023. arXiv: 2311.07424 [cs.CL].

[56] Z. Sun, X. Wang, Y. Tay, Y. Yang, and D. Zhou. "Recitation-augmented language models". In: *arXiv preprint arXiv:2210.01296* (2022).

[57] R. Weng, H. Yu, X. Wei, and W. Luo. "Towards enhancing faithfulness for neural machine translation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 2675–2684.

[58] H. Li, J. Zhu, J. Zhang, and C. Zong. "Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization". In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 1430–1441.

[59] C. Malaviya, S. Lee, S. Chen, E. Sieber, M. Yatskar, and D. Roth. *ExpertQA: Expert-Curated Questions and Attributed Answers*. 2023. arXiv: 2309.07852 [cs.CL].

[60] A. Mittal. "Tackling Hallucination in Large Language Models: A Survey of Cutting-Edge Techniques". In: *Unite.AI* (Jan. 2024). URL: https://www.unite.ai/tackling-hallucination-in-large-language-models-a-survey-of-cutting-edge-techniques/.

[61] S. Chen, S. Buthpitiya, A. Fabrikant, D. Roth, and T. Schuster. *PropSegmEnt: A Large-Scale Corpus for Proposition-Level Segmentation and Entailment Recognition*. 2023. arXiv: 2212.10750 [cs.CL].

[62] X. Zhang and W. Gao. *Towards LLM-based Fact Verification on News Claims with a Hierarchical Step-by-Step Prompting Method*. 2023. arXiv: 2310.00305 [cs.CL].

[63] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[64] I.-C. Chern, S. Chern, S. Chen, W. Yuan, K. Feng, C. Zhou, J. He, G. Neubig, and P. Liu. *FacTool: Factuality Detection in Generative AI – A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios*. 2023. arXiv: 2307.13528 [cs.CL].

[65] Y. Wang, R. G. Reddy, Z. M. Mujahid, A. Arora, A. Rubashevskii, J. Geng, O. M. Afzal, L. Pan, N. Borenstein, A. Pillai, I. Augenstein, I. Gurevych, and P. Nakov. *Factcheck-GPT: End-to-End Fine-Grained Document-Level Fact-Checking and Correction of LLM Output*. 2023. arXiv: 2311.09000 [cs.CL].

[66] S. Balloccu, P. Schmidtová, M. Lango, and O. Dušek. *Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs*. 2024. arXiv: 2402.03927 [cs.CL].

[67] G. Salton, A. Wong, and C.-S. Yang. "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11 (1975), pp. 613–620.

[68] A. Das and A. Jain. "Indexing the world wide web: The journey so far". In: *Next Generation Search Engines: Advanced Models for Information Retrieval*. IGI Global, 2012, pp. 1–28.

[69] M. Kobayashi and K. Takeda. "Information retrieval on the web". In: *ACM computing surveys (CSUR)* 32.2 (2000), pp. 144–173.

[70] S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, and S. Quarteroni. *Web information retrieval*. Springer Science & Business Media, 2013.

[71] D. E. Rose and D. Levinson. "Understanding user goals in web search". In: *Proceedings of the 13th international conference on World Wide Web*. 2004, pp. 13–19.

[72] A. Singhal et al. "Modern information retrieval: A brief overview". In: *IEEE Data Eng. Bull.* 24.4 (2001), pp. 35–43.

[73] C. D. Manning. *An introduction to information retrieval*. Cambridge university press, 2009.

[74] J. J. Pan, J. Wang, and G. Li. "Survey of vector database management systems". In: *arXiv preprint arXiv:2310.14021* (2023).

[75] *Apache Hadoop*. Accessed: 2024-02-11. The Apache Software Foundation. URL: https://hadoop.apache.org/.

[76] J. Johnson, M. Douze, and H. Jégou. "Billion-scale similarity search with gpus". In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[77] *Chroma*. https://www.trychroma.com/. Accessed: 2024-02-11.

[78] J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu, et al. "Milvus: A purpose-built vector data management system". In: *Proceedings of the 2021 International Conference on Management of Data*. 2021, pp. 2614–2627.

[79] *Vector Database Benchmark Tool*. https://zilliz.com/vector-database-benchmark-tool. Accessed: 2024-02-11. Zilliz.

[80] *ANN-Benchmarks*. https://ann-benchmarks.com/. Accessed: 2024-02-11.

[81] Y. A. Malkov and D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". In: *IEEE transactions on pattern analysis and machine intelligence* 42.4 (2018), pp. 824–836.

[82] H. Jegou, M. Douze, and C. Schmid. "Product quantization for nearest neighbor search". In: *IEEE transactions on pattern analysis and machine intelligence* 33.1 (2010), pp. 117–128.

[83] *Approximate Nearest Neighbors in C++/Python optimized for memory usage and loading/saving to disk*. https://github.com/spotify/annoy. Accessed: 2024-02-11. Spotify.

[84] University of Freiburg. *Large Question Answering Datasets*. https://github.com/ad-freiburg/large-qa-datasets. 2020.

[85] J. Berant, A. Chou, R. Frostig, and P. Liang. "Semantic parsing on freebase from question-answer pairs". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, pp. 1533–1544.

[86] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. "MS MARCO: A human generated machine reading comprehension dataset". In: *choice* 2640 (2016), p. 660.

[87] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension". In: *arXiv preprint arXiv:1705.03551* (2017).

[88] *PCA Documentation in scikit-learn 1.4.1*. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. Accessed: 2024-02-18.

[89] S. Chen, H. Zhang, T. Chen, B. Zhou, W. Yu, D. Yu, B. Peng, H. Wang, D. Roth, and D. Yu. "Sub-Sentence Encoder: Contrastive Learning of Propositional Semantic Representations". In: *arXiv preprint arXiv:2311.04335* (2023). URL: https://arxiv.org/pdf/2311.04335.pdf.

[90] Y. Xia. *Factcheck-GPT*. https://github.com/yuxiaw/Factcheck-GPT. Accessed: 2024-03-04. 2023.

[91] S. Robertson, H. Zaragoza, et al. "The probabilistic relevance framework: BM25 and beyond". In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389.

[92] A. Piktus, F. Petroni, V. Karpukhin, D. Okhonko, S. Broscheit, G. Izacard, P. Lewis, B. Oğuz, E. Grave, W.-t. Yih, et al. "The web is your oyster-knowledge-intensive NLP against a very large web corpus". In: *arXiv preprint arXiv:2112.09924* (2021).

[93] Google. *Programmable Search Engine*. Accessed: 2024-03-05. 2024. URL: https://programmablesearchengine.google.com/about/.

[94] N. Reimers and I. Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[95] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. "The Faiss library". In: (2024). arXiv: 2401.08281 [cs.LG].

[96] LangChain. *LangChain: Building, Observing, and Deploying LLM-powered Applications*. https://www.langchain.com/. Accessed: 2024-03-05. 2023.

[97] P. He, J. Gao, and W. Chen. *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*. 2023. arXiv: 2111.09543 [cs.CL].

[98] M. Laurer, W. van Atteveldt, A. Salleras Casas, and K. Welbers. *Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI*. https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli. Accessed: 2024-03-05. 2022.

[99] X. Li and J. Li. *AnglE-optimized Text Embeddings*. 2023. arXiv: 2309.12871 [cs.CL].