Retrieving Supporting Evidence for Generative Question Answering

by

Siqing Huo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes the material which is presented at REML 2023 [25] and SIGIR-AP 2023 [24]. I conducted all experiments and wrote the first drafts of these papers. Negar Arabzadeh and my supervisor Professor Charles L.A. Clarke both contributed ideas, feedback and editing. Professor Charles L.A. Clarke additionally provided supervision.

Abstract

Current large language models (LLMs) can exhibit near-human levels of performance on many natural language-based tasks, including open-domain question answering. Unfortunately, at this time, they also convincingly hallucinate incorrect answers, so that responses to questions must be verified against external sources before they can be accepted at face value. In the thesis, I report two simple experiments to automatically validate generated answers against a corpus. We base our experiments on questions and passages from the MS MARCO (V1) test collection, and a retrieval pipeline consisting of sparse retrieval, dense retrieval and neural rerankers. In the first experiment, we validate the generated answer in its entirety. After presenting a question to an LLM and receiving a generated answer, we query the corpus with the combination of the question + generated answer. We then present the LLM with the combination of the question + generated answer + retrieved answer, prompting it to indicate if the generated answer can be supported by the retrieved answer. In the second experiment, we consider the generated answer at a more granular level, prompting the LLM to extract a list of factual statements from the answer and verifying each statement separately. We query the corpus with each factual statement and then present the LLM with the statement and the corresponding retrieved evidence. The LLM is prompted to indicate if the statement can be supported and make necessary edits using the retrieved material. With an accuracy of over 80%, we find that an LLM is capable of verifying its generated answer when a corpus of supporting material is provided. However, manual assessment of a random sample of questions reveals that incorrect generated answers are missed by this verification process. While this verification process can reduce hallucinations, it can not entirely eliminate them.

Acknowledgements

I would like to express my sincere gratitude to Negar Arabzadeh and my supervisor Professor Charles L.A. Clarke for their invaluable support and meticulous editing throughout the completion of my thesis. I am deeply grateful for their exceptional guidance and editing assistance. Their contributions have been pivotal in the successful completion of this thesis.

This research was funded by Canada's Natural Sciences and Engineering Research Council through its Discovery Grants program.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

There has been rapid progress in the field of Natural Language Processing due to recent advancements in transformer-based large language model (LLM)s [13, 16, 31, 45, 52]. These LLMs have produced substantial improvements in text generation tasks such as question answering, summarization, and machine translation [11, 28, 35, 39, 56, 61, 67]. However, despite the excitement created by these improvements, the LLMs may confidently and convincingly generate hallucinated results [3, 26]. Avoiding hallucinations is particularly important when LLM generated text is presented directly to users, especially in critical circumstances, such as health and medicine. For example, a chatbot which is designed to help people learn more about diseases should not generate responses that are inconsistent with evidence-based medicine [12].

Current LLMs lack the ability to self-detect hallucinations in generated texts as they do not have access to an external source of knowledge [3]. Proposed zero-resource methods of self-detecting LLM hallucinations often require the access to token-level probabilities [37], which are not available for black-box LLMs. On the other hand, information retrieval methods have been long studied and are now capable of rapidly locating the top documents relevant to queries from arbitrarily large text corpora [48]. Researchers have shown interest in enhancing the reliability of LLM's generated texts by incorporating external sources of knowledge using retrieval techniques. Attribution [5, 46, 64] focuses on connecting generated texts to supporting evidence to make them more trustworthy. Retrieval-augmented generation approaches [15, 32] attempt to ensure the reliability of generated texts by conditioning the LLM's generation on retrieved material. However, such approaches still suffer from hallucination and cannot guarantee attribution. The LLM may make claims not found in the retrieved material [14] or contradictory to the retrieved material [36].

As opposed to retrieval-augmented generation that performs retrieval before generation, more recent works such as RARR [18] proposed to perform retrieval after generation. The proposed framework suggests examining the produced text and making edits to align it with the gathered evidence, while maintaining the overall structure of the original text in case of any contradictions. Since our focus is on self-detecting and correcting hallucinations, we also perform retrieval after generation. However, unlike RARR which makes uses of few-shot prompting [7] and external query-document relevance model, our experiments use nothing else besides the LLM itself and a retrieval pipeline. Substantial prompt engineering is not needed.

The LLM's generated texts is often more than just a single and atomic factual claim, it can be helpful to decompose a piece of generated text into a series of factual claims. Many previous works have studied decomposing long piece of text into atomic factual claims [10, 40, 58]. While we took inspiration from these works, we use solely the LLM itself to achieve such decomposition with no training, no human intervention, and minimal prompt engineering.

In the thesis, I investigate the ability for LLMs to self-detect hallucinations by confirming its generated responses against an external corpus. More specifically, we experimentally test the degree to which an LLM hallucinates answers when performing an open-domain, general question-answering task, and whether it can automatically verify its responses when presented with a dataset containing known correct answers, with the help of retrieval methods. Our experiments include manual checks of comparisons made by the LLM. These experiments demonstrate that the LLM can correctly detect its own hallucinations in a majority of cases (an accuracy of over 80%), with the help of retrieval methods. However, while our verification process can reduce hallucinations, it can not entirely eliminate them. Even when verified with by process, we should still be cautious when depending on LLM-generated answers, especially in critical circumstances. The data for all the experiments is released[1].

The main contributions of this thesis can be summarised as follows:

1. Proposed a simple methodology for the LLM to self-detect hallucinations in its generated texts with the help of an information retrieval system to retrieve supporting evidence.

2. Proposed a simple methodology for the LLM to self-detect hallucinations, correct hallucinations, and fully attribute its generated texts, with the help of an information retrieval system to retrieve supporting evidence.

[1]https://github.com/siqingh/llm_retrieve_supporting_evidence

3. Manually labelled 2,200 samples of the LLM's classifications.
4. Provided a detailed analysis of the degree to which an LLM hallucinates answers when performing an open-domain, general question-answering task.
5. Provided a detailed analysis of whether an LLM can automatically verify its responses when presented with a dataset containing known correct answers, with the help of retrieval methods.

This thesis is organized as follows: In Chapter 2, we first introduce the problem of LLM hallucinations, present the concepts in the field of fact checking, and present existing methods of mitigating the issue of LLM hallucinations. We then present the common setup for both of our experiments, including the LLM, dataset and retrieval methods used and their hyperparameters in Chapter 3. In Chapter 4, we present the methodology, results and analysis of our first experiment, where we test the LLM's ability to self-detect hallucinations against retrieval materials. In Chapter 5, we present the methodology, results and analysis of our second experiment, where we test the LLM's ability to self-detect hallucinations, correct hallucinations, and fully attribute its generated texts using retrieved materials. We then discuss the important limitations of our research in Chapter 6. Finally, in Chapter 7, we summarize the findings of our experiments and suggest potential directions for future works.

Chapter 2

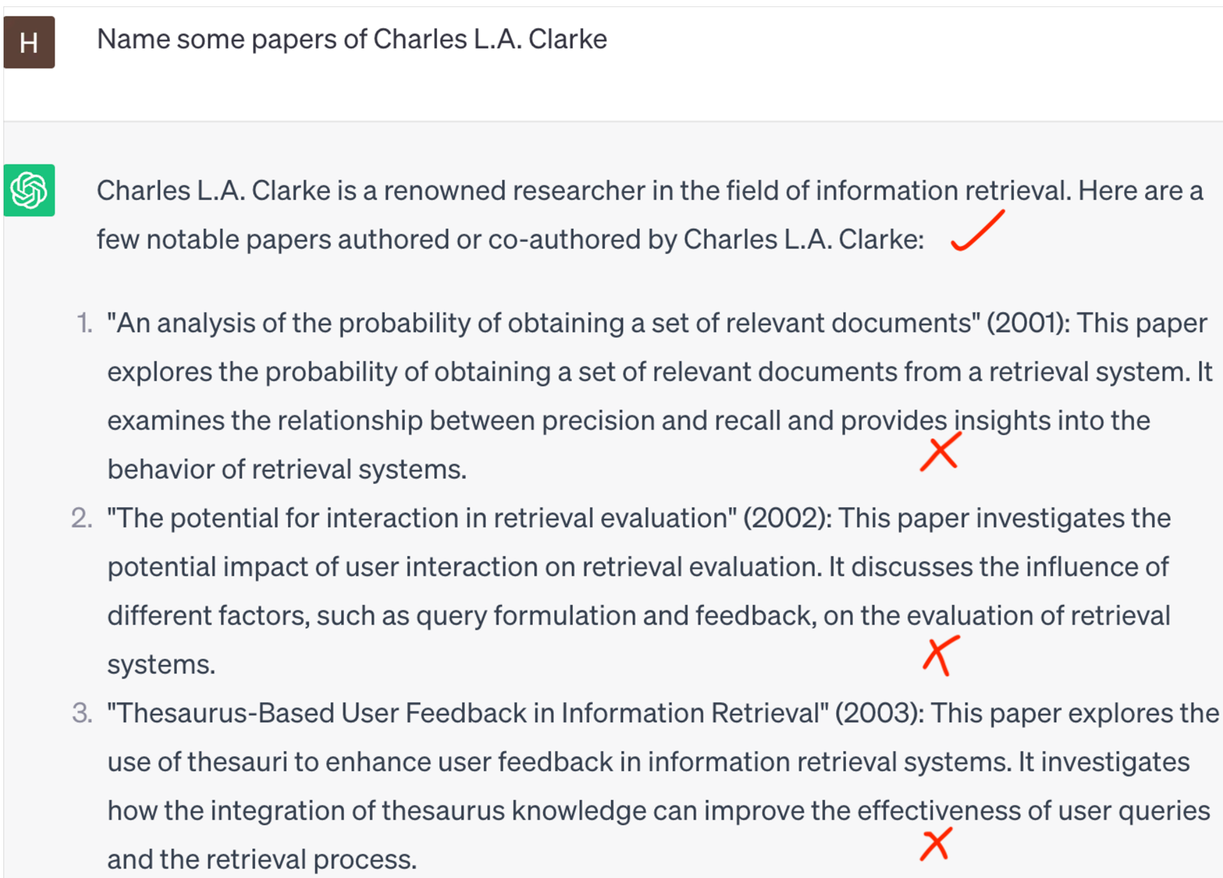
Background

In this chapter, we first provide the reader with the background of the problem of LLM hallucination. We then discuss traditional methods of automated fact-checking and recent approaches of retrieval-augmented generation aiming to mitigate the issue of LLM hallucination.

2.1 LLM Hallucination

Despite exhibiting superb performance on many natural language-based tasks, LLMs are known to be susceptible to hallucination [3]. The hallucinated text generated by an LLM is unfaithful and incorrect, but it is particularly dangerous as it often appears natural and coherent [26].

Past research has shown that the LLM suffers from hallucination in many downstream text generation tasks [3, 26]. In the context of summarization, hallucination refers to the problem that the output text is unfaithful to the input text [38, 49]. In the context of machine translation, hallucination refers to the problem that the output translation is fluent but unrelated to the input text [54, 60]. In the scope of this thesis, we mainly focus on the problem of hallucination in the question answering task. In the context of question answering, hallucinated answers refers to LLM's generated responses that contain unfaithful and incorrect information. As shown in Figure 2.1, the LLM confidently listed several papers which appear to align with what a professor conducting research in information retrieval might write, yet none of these papers are real.


H

Name some papers of Charles L.A. Clarke

Charles L.A. Clarke is a renowned researcher in the field of information retrieval. Here are a few notable papers authored or co-authored by Charles L.A. Clarke:

1. "An analysis of the probability of obtaining a set of relevant documents" (2001): This paper explores the probability of obtaining a set of relevant documents from a retrieval system. It examines the relationship between precision and recall and provides insights into the behavior of retrieval systems.
2. "The potential for interaction in retrieval evaluation" (2002): This paper investigates the potential impact of user interaction on retrieval evaluation. It discusses the influence of different factors, such as query formulation and feedback, on the evaluation of retrieval systems.
3. "Thesaurus-Based User Feedback in Information Retrieval" (2003): This paper explores the use of thesauri to enhance user feedback in information retrieval systems. It investigates how the integration of thesaurus knowledge can improve the effectiveness of user queries and the retrieval process.

Figure 2.1: Example of hallucination in LLM generated texts.

Even though various theories about the underlying reason for hallucinations exist (training data contains hallucinated content [14], training objective does not reward faithfulness [38], and etc.,), the problem remains unresolved and continues to be an area of active research.

2.2 Fact Checking

Fact checking is a concept originated from journalism, where it refers to the process of assessing the truthfulness of claims commonly carried out by journalists when creating news articles [53].

The process of manual fact checking demands a substantial investment of time, often ranging from multiple hours to days for even the most proficient human fact-checkers [22]. Journalist would often be required to find relevant information from numerous potential sources, appraise the reliability of each source, and make a comparative analysis. While transitioning from print media to electronic media, we witness a substantial growth of both the amount of information and the speed of information dissemination. This has heightened the importance of fact checking, yet also implies that manual fact checking is impractical. Therefore, there is an emerging need of automating the process of fact checking, which can potentially be realized with natural language processing and information retrieval techniques.

In general, the process of fact checking can be broken down into three main steps [21]:

1. **Claim Detection**: identify the claims that needs to be checked.
2. **Evidence Retrieval**: find relevant material that may support or contradict a claim.
3. **Claim Verification**: determine whether a claim is supported or contradicted based on the evidence.

In the following subsections, we will delve into more details of each of these steps.

2.2.1 Claim Detection

Claim detection refers to the process of identifying the claims that needs to be checked.

One metric is to check only those claims that are check-worthy, where check-worthy claims refers to non-trivial claims that people are actually interested in knowing the truth value of [23]. An example of a not check-worthy claim would be “water is wet”.

Another possible metric is to classify claims based on subjectivity. We check only those claims about the world, but leave subjective claims such as “cubist art is beautiful” alone [29].

Furthermore, in reality when validating complex claims we often discover they are only partially supported. One potential solution to such issue is to add another step to claim detection. Namely, after initial discovery of claims, we decompose a complex claim into a list of atomic claims so we can retrieve evidence and verify each atomic claim separately [10, 40, 58].

In our first experiment, we treat the LLM’s full response to each given question as a claim, and we define those responses that do not provide a valid answer as claims that are not check-worthy. In our second experiment, we prompt the LLM to automatically extract a list of atomic facts that are check-worthy.

2.2.2 Evidence Retrieval

Evidence retrieval refers to the process of finding relevant material that may support or contradict a claim.

Although evidence can take various forms (such as images, audios, videos, texts, tables, etc.), our experiments exclusively focus on the retrieval of textual information from a corpus using information retrieval techniques.

Furthermore, in reality, not all source of information is trustworthy. However, in the context of our experiments, we assume all the documents in the corpus which we retrieve supporting material from are trustworthy. Assessing the evidence themselves are out of the scope of this thesis.

2.2.3 Claim Verification

Claim verification refers to the process of determining whether a claim is supported or contradicted based on the evidence.

Claim verification can be broken down into two components that can be addressed either independently or in combination: verdict prediction, where we determine the truthfulness of a claim, and justification production, where we explain how we determined the truthfulness of a claim.

Verdict Prediction

Verdict prediction refers to the process of determining the truthfulness of a claim given an identified claim and its retrieved supporting material. The simplest approach is treating verdict prediction as a binary classification task, where one class indicate the claim is true and another class indicate the claim is false [43]. An extension to this approach is to include another class to indicate a lack of information to predict the truthfulness of the claim [51]. A more refined classification scheme would perform a multi-class classification and label the claim according to its degree of truthfulness [55].

In our first experiment, we follow a stepped classification approach. We first attempt to rule out all the claim-evidence pairs where the evidence does not provide enough information to determine the truthfulness of the claim. Then, we simply classify the remaining claim-evidence pairs as either the claim is true or false according to the evidence.

In our second experiment, we label a claim-evidence pair as one of the following: "Supported" indicates that the claim is supported by the evidence, "Contradictory" indicates that the claim is refuted by the evidence, "Neither" indicates the evidence does not provide enough information to determine the truthfulness of the claim.

Justification Production

Justification production refers to give a clear and coherent explanation of why a claim-evidence pair is classified as it is.

Justification can be particularly important for automated fact checking because black-box models may have unrevealed biases inherent from the training data, and such biases may lead to inadvertent and detrimental outcomes [42]. Providing justification of the model's predictions may give human an opportunity to discover the underlying issues of black-box models [34].

One of the simplest approach to justification is to present all the evidence used to reach the classification along with the claim. It is simple because it does not explain how the evidence is used to arrive at the final predicted label. These evidence presented alongside of the claim is referred to as the attribution of the claim [5, 46, 64]. More sophisticated justification should include not only the evidence used, but also how the evidence was employed and the logical steps of how the decision was reached. In the case of automated fact checking, the justification should also be automatically generated [30].

In all of our experiments, although we do ask the LLM to generate a brief explanation of its classification, we are not concerned with the quality of the justifications. Instead, we merely use these justifications to gain insights into the potential cause of errors in the LLM's verdict predictions.

2.3 Methods of Mitigating Hallucination in LLM's Generated Texts with Retrieved Supporting Evidence

As previously discussed in Section 2.1, hallucination can have slightly different definition for different downstream text generation task. In the scope of this thesis, we are mainly concerned with LLM hallucination that refers to unfaithful or incorrect information generated. In this section, we will present some existing methods attempting to resolve the issue of LLM hallucination.

In general, methods using information retrieval to mitigate LLM hallucination follows two main paradigms:

- **the retrieve-then-generate approach**: often referred to as "retrieval-augmented generation". First retrieve relevant documents for the given question and then let the LLM to generate an answer conditioning on the retrieved documents.
- **the generate-then-retrieve approach**: perform retrieval after generation. First let the LLM generate a response to the question, then retrieve supporting material. The LLM's generated texts are then attributed to the retrieved evidence or post-edited based on the retrieved evidence to produce a final generated response that is supposedly free of hallucination.

In the following subsections, we will delve into more details of each type of existing methods.

2.3.1 Retrieve-then-generate

Retrieval-augmented generation approaches [15, 32] first retrieve supporting material and then generate texts based on the retrieved information. Retrieval-augmented generation approaches give an LLM direct to access a vast external knowledge base at inference time. This type of methods serves as an effective alternative to scaling up the amount of training data and the number of parameters in an LLM to improve the LLM's performance.

Providing the retrieved material as the context and prompt the LLM to generate based on the given context is shown to be effective in mitigating the issue of hallucination. Simple tricks like giving each piece of evidence a unique tag and prompt an LLM to cite all of its generated texts can also be helpful [15]. However, retrieval-augmented generation approaches can still generate hallucinated texts which contradicts the retrieved evidence [36] or not supported by the retrieved information [14].

2.3.2 Generate-then-retrieve

In the generate-then-retrieve paradigm, generation happens first and we perform retrieval based on the LLM's generated contents [18, 20, 27, 63]. We then decide if the generated texts are supported by the retrieved materials. If the generated texts are supported by the retrieved evidence, they can be attributed to the retrieved evidence. If the retrieved material contradicts the generated texts, we can post-edit the generated texts using the retrieved material. The goal is such that the final generated texts can be fully attributed and free of hallucination.

After supporting material is retrieved for the generated text, the first decision to make is whether the generated text is supported by the retrieved evidence. The LLM itself can be prompted to perform the task, but it is shown to be subjective to errors such as unable to properly handle fine-grained information or overlooking contextual cues [64]. We can also separately train and finetune a model for this purpose [66]. We can also break down a piece of generated text into a more fine-grained list of atomic facts. This way we can determine the percentage of the text that is supported instead of making a simple binary classification [40, 65].

The other main task is post-editing, which can also be achieved by prompting the LLM itself (zero-shot or few-shot prompting) or by training a separate model exclusively for this purpose [9].

The generate-then-retrieve paradigm is shown to be able to achieve decent correctness yet much worse attribution quality than the retrieve-then-generate paradigm in general [19].

In our experiments, we follow the generate-then-retrieve paradigm and perform minimal prompt engineer. In our first experiment, we simply make a binary decision of whether the generated text can be supported by the retrieved evidence. In our second experiment, we also attempt to fully attribute the generated texts and post-edit any claim contradicted by the retrieved evidence.

Chapter 3

Experimental Setup

In our experiments, we choose `gpt-3.5-turbo` as the LLM representative with the temperature set to 0, consistent with OpenAI recommendations for classification tasks. We used the MS MARCO (V1) passage collection[1] [41] for questions and answer validation. MS MARCO is a large-scale dataset with over 8 million passages for the development and evaluation of machine reading comprehension models. MS MARCO is accompanied by sparsely labeled queries as its training set, development set and test set. In this paper, we run experiments on the 6980 questions in the MS MARCO (V1) small development set.

We run our set of experiments with two different retrieval methods.

As the first retriever, we employ the Okapi BM25 [47] ranking function, which is a well-known and widely-used baseline retrieval method. For the BM25 function parameters, k_1 is set to 0.82 and b is set to 0.68, which are standard values tuned for the MS MARCO passage retrieval task by grid search. Since BM25 requires exact matching between query terms and document terms, we speculate that it may perform well for answer verification by providing support for the terms used in the generated answer.

The second retrieval method we adopt for our experiments is a more modern neural retrieval method that emphasizes the quality of the retrieved passages over retrieval efficiency. The pipeline comprises an initial retrieval stage followed by a reranking stage. For the retrieval stage, we employ a combined pool of sparse and dense retrieval. We use SPLADE [17] as the sparse retrieval method, and ANCE [59] as the dense retrieval method. Both retrieval methods are shown to be highly effective [50, 62]. We pool the top 100 documents retrieved by both retrieval methods. For the reranking stage,

[1] https://github.com/microsoft/MSMARCO-Passage-Ranking

we use a combination of MonoT5 and DuoT5 neural rerankers [44]. We used MonoT5 to rerank the pooled documents from the retrieval stage, and we use DuoT5 to rerank the top 10 documents selected by MonoT5. We select this multi-stage neural retrieval stack (SPLADE+ANCE+MonoT5+DuoT5) as similar approaches have shown excellent performance on the MS MARCO passage ranking task. Executing our implementation of this neural retrieval pipeline on the MS MARCO (V1) small development set achieves a MRR@10 of 0.40. All the aforementioned methods are implemented using the Pyserini toolkit with default parameters[2] [33].

[2] https://github.com/castorini/pyserini/

Chapter 4

Experiment 1: Validate the Generated Response as a Whole

In this chapter, we propose a straightforward experiment to investigate the LLM's ability to self-verify its generated texts against retrieved passages. We explain the proposed methodology and present the results of this experiment.

4.1 Methodology

The first proposed experiment is straightforward, essentially the simplest method that one could envision for employing LLMs to self-verify against retrieved passages. We first prompt the LLM to answer the question. We then combine its generated answer with the original question and use the result to query a corpus of passages expected to contain supporting evidence. In order for the LLM to self-detect hallucinations, we then present the question, the generated answer, and the potential evidence to the LLM, prompting it to determine if the evidence supports the answer.

Figure 4.1 shows an overview of our pipeline. Starting with a question, we prompt the LLM to answer it (Figure 4.2). We direct the LLM to act as an expert in order to set a more rigorous and less casual tone for the response [57]. Then, inspired by query expansion methods which have shown to be effective and help avoid topic drift problems [1, 2, 4, 8, 68], we employ the generated answer to curate a combined query for the fact-checking step. In other words, we combine the original question with the answer generated by the LLM for a second fact-checking or confirmation phase. We execute the combined query over a

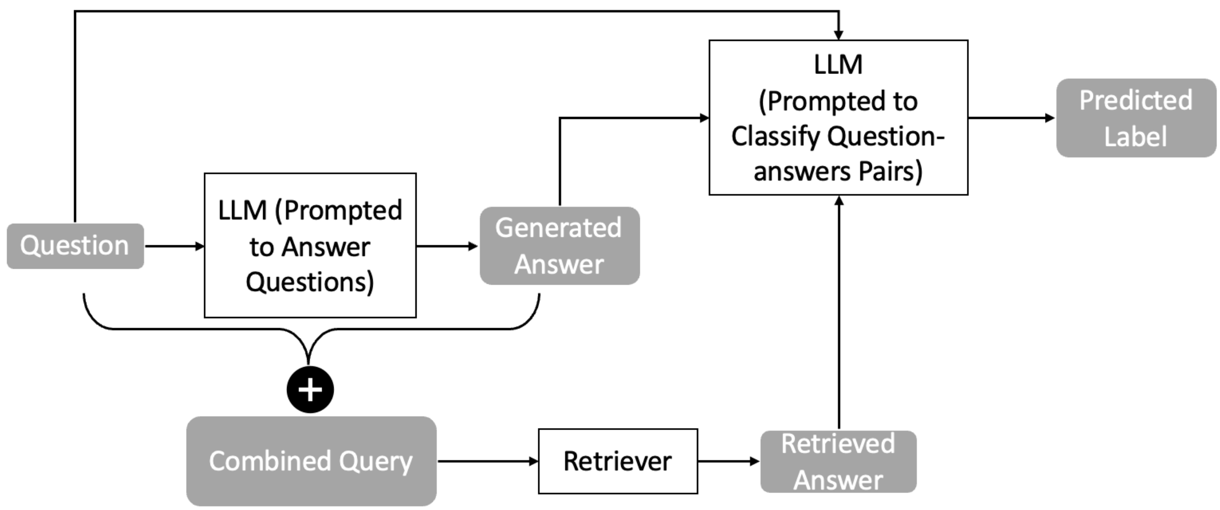


Figure 4.1: Self-detecting hallucination in LLMs.

You are an expert in this field. Please answer the question as simply and concisely as possible.

Question: {query}
Answer:

Figure 4.2: Prompt for answering question.

collection of passages to retrieve passages that are both relevant to the original question and that may support the LLM's generated answer. We then combine the original question, the generated answer, and the retrieved answer, and prompt the LLM to determine if the retrieved answer supports the generated answer (Figure 4.3, Figure 4.4, and Figure 4.5). We summarize our proposed strategy as follows:

1. Prompt the LLM to answer the question.

2. Combine the LLM's answer with the original question.

3. Execute the combined query on an external corpus (expected to contain correct answers), retrieving the most relevant passage(s).

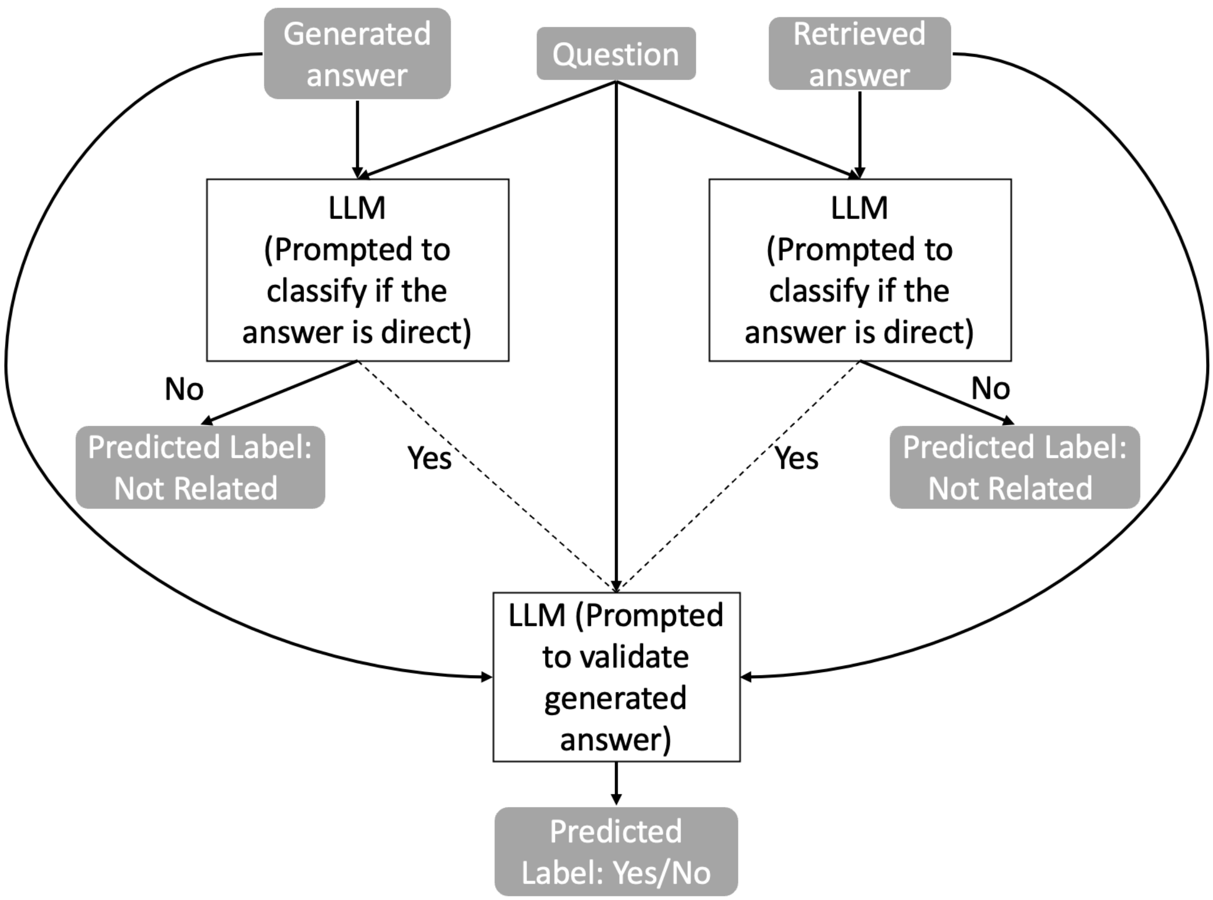


Figure 4.3: Stepped classification of a question-answers pair.

I want you to act as an assessor of the answer. You will be given a question and an answer, and you need to determine whether the answer directly answers the question. Examples of non-direct answers would be claiming it does not know or does not have enough information, and provide some alternative ways to find answers. Also note that if an answer claims that the question itself is wrong, it also is a form of direct answer. Your response should be 'Yes' if the answer actually answers the question, ad 'No' if the answer does not actually answer the question. Please also include a short and concise explanation of your classification.

Question: {query}
Answer: {answer}

Figure 4.4: Prompt for assessing whether the answer directly addresses the question.

4. Prompt the LLM to compare its generated answer against the retrieved results from the combined query, with the goal of self-detecting hallucinations.

In the following subsections, we elaborate on individual components of this pipeline.

4.1.1 Retrieved Answer

We experiment with three different types of retrieved answer:

- BM25 retrieved answer: the most relevant passage retrieved using the Okapi BM25 ranking function.
- Neural retrieved answer: the most relevant passage retrieved using the multi-stage neural retrieval stack (SPLADE+ANCE+MonoT5+DuoT5).
- Reader extracted answer: obtained by prompting the LLM to act as a reader and extract a more concise answer from the top three passages retrieved by the multi-stage neural retrieval stack.

  There can be multiple ways to address the same question, and the answers provided by the model may span various perspectives. Consequently, we retain the top three retrieved passages. However, simply concatenating these passages without any refinement may lead to abrupt shifts, repetitions, or excessive length. Hence, we utilize

I want you to act as an assessor of the answer. You will be provided with a question, an answer, and relevant evidence. Your task is to assess whether the evidence provided supports the given answer. If the evidence supports the answer, reply with a 'Yes'. Otherwise, reply with a 'No'. Please also include a short and concise explanation of your classification.

Question: {query}
Answer: {LLM answer}
Evidence: {Retrieved answer}

Figure 4.5: Prompt for validating generated answer.

a reader to perform a question-based summary of the top three retrieved passages, resulting in a summary that aligns better with the concise and direct nature of the LLM's generated answer. To do so, we prompt the LLM with the prompt shown in Figure 4.6, asking the LLM to act as an expert to extract the relevant answer in a concise format, given a question and the top three retrieved passages concatenated.

4.1.2 Validating Generated Answer

Inspired by query expansion methods [1, 2, 4, 68], we combine the original question with the generated answer for retrieval to satisfy the goal of retrieving passages that are not only relevant to the original question but also directly support the LLM's answer. Some questions will have multiple acceptable answers and interpretations, and we want to retrieve the passage that best supports the generated answer. We assume that the content of the retrieved passage is truly relevant to the question, and since the retrieved passage is also close to the LLM's answer, it can serve as the benchmark to validate the LLM's answer. In order to make this comparison, we perform a stepped classification of each pair of question, generated answer, and retrieved answer as shown in Figure 4.3. The first step is prompting the LLM with the prompt shown in Figure 4.4 for it to decide if the generated answer and the retrieved answer actually address the question or not. In the second step, we classify the question, generated answer and retrieved answer pair. If the LLM classifies either one of the generated answer and the retrieved answer as not answering the question, we classify the question-answers pair as 'Not Related'. Only if both answers are classified as direct answers to the question, we prompt the LLM to classify if the generated answer is supported by the retrieved answer with the prompt shown in Figure 4.5.

I want you to act as a question-based summarizer for a set of passages. Given a question and a passage containing answer to the question, your task is to provide a clear and concise summary of the passage that directly answer the question and contain minimal extra information. Your summary should be easy to understand and accurately represent the passage. Keep in mind that your summary should be objective and avoid including personal opinions or biases. If the passage does not answer, simply reply with 'No Answer', otherwise reply with just the summary itself and nothing else.

Question: {query}
Passage 1: {passage1}
Passage 2: {passage2}
Passage 3: {passage3}

Figure 4.6: Prompt for reader task.

We categorize the outcome of the LLM's decision into three different classes:

- We interpret the "Yes" class as indicating that there is no hallucination since the retrieved passages provide supporting evidence to the LLM's answer;
- We interpret the "No" class as indicating there is likely hallucination since the retrieved passages fail to support the LLM's answer;
- We interpret the "Not Related" class to indicate the LLM responded with a clarification request or claims it does not know the answer (For example, "*I would need more context to provide a specific answer. Please provide additional details about the situation or event you are referring to.*", "*I'm sorry, but I don't have access to ...*" or "*I do not know. It's best to check ... for the most up-to-date information on ...*") or the retriever failed to retrieve relevant passage. In either case, the question-answers pair will be irrelevant for the task of hallucination detection.

In addition, we also experiment with prompting the LLM to classify if the generated answer is supported by the qrel passages with the prompt shown in Figure 4.5. Qrel passages are passages deemed highly relevant to the question by human annotators, so we assume these passages actually address the question without prompting the LLM with the

Table 4.1: LLM's classifications of the answers.

	Does the LLM's generated answer directly answer the question?	**Does the reader extracted answer directly answer the question?**	**Does the neural retrieved answer directly answer the question?**	**Does the BM25 retrieved answer directly answer the question?**
Yes	6,512 (93.30%)	5,292 (75.82%)	4,202 (60.20%)	2,698 (38.65%)
No	468 (6.70%)	1,688 (24.18%)	2,778 (39.80%)	4,282 (61.35%)

prompt shown in Figure 4.4. Since these passages cannot be obtained free of human intervention, this is irrelevant to the main task of exploring the LLM's ability to automatically validate generated answers against a corpus. It is included to gain more insights into the effectiveness of our approach of combining the original question with the generated answer to curate a combined query for the fact-checking step.

Furthermore, note that in Figure 4.4 and Figure 4.5 we have asked the LLM to briefly explain its classifications. This is unnecessary and unrelated to the primary task of classification. Its inclusion serves the sole purpose of potentially gaining further insights into the causes of LLM's mistakes during manual inspections.

4.2 Results

4.2.1 Step 1: Classifying Answers

Table 4.1 shows how LLM classifies different types of answers. According to the LLM's classifications, the generated answer provided by LLM adequately addresses the question in the majority of cases (approximately 93% of the time). The order of relevance for the retrieved answers, from most to least relevant to the question, is as follows: the reader-extracted answer, the neural retrieved answer, and the BM25 retrieved answer, which aligns with the expectation.

To further investigate the reliability of these classifications, we manually examine and carefully inspect 100 randomly selected samples from each cell of Table 4.1. In other words, for each of the four types of answers (generated answer, reader extracted answer, neural retrieved answer, and BM25 retrieved answer), we randomly select 100 samples where the

Table 4.2: Results of manually verifying LLM's classifications of the answers.

LLM's Classification	**Labeller's Opinion**	**Does the LLM's generated answer directly answer the question?**	**Does the reader extracted answer directly answer the question?**	**Does the neural retrieved answer directly answer the question?**	**Does the BM25 retrieved answer directly answer the question?**
Yes (100 samples)	**Correct**	98	99	99	92
	Incorrect	2	1	1	8
No (100 samples)	**Correct**	82	58	38	62
	Incorrect	18	42	62	38

LLM classified as direct answers and 100 samples where the LLM classified as non-direct answers. The manual inspection results are shown in Table 4.2.

From Table 4.2 we see that in general there are far more false negatives than false positives. Furthermore, for the samples where the LLM classified as non-direct answers, its only achieves high accuracy for the generated answer but is only slightly better than random guesses (50%) for all types of retrieved answers. This suggests that perhaps different prompt is needed for classifying the retrieved answers.

In the following subsections, we will present specific instances and interesting patterns observed where the labeller and the LLM disagree.

Classify Generated Answer

An illustrative example of false positive errors when the LLM classifies generated answer can be observed in Figure 4.7, despite the prompt clearly stating that "*Examples of non-direct answers would be ... and provide some alternative ways to find answers*", the LLM still makes an incorrect classification and explains its choice by "*the answer directly answers the question by acknowledging its limitations and offering an alternative solution*". This example demonstrates that such errors cannot be rectified by simply stating the desired outcome in the prompt. Instead, better prompt engineering or fine-tuning of the LLM may be needed. Furthermore, note that in other similar cases, the LLM is able to make the correct choice (Figure 4.8).

Question: what is the weather in powell wy
Generated Answer: I'm sorry, as an AI language model, I don't have access to real-time weather information. However, you can check current weather conditions in Powell, WY by searching online or checking a weather app.

LLM's Classification: Yes, the answer directly answers the question by acknowledging its limitations and offering an alternative solution to finding the information requested.

Figure 4.7: False positive example of generated answer classification.

Question: what time was earthquake in alaska?
Generated Answer: I'm sorry, I do not have the ability to provide real-time information on natural disasters. Please check with a reliable news source or the United States Geological Survey for up-to-date information on earthquakes in Alaska.

LLM's Classification: No. The answer does not directly answer the question. Instead, it states that it does not have the ability to provide real-time information. It then suggests alternative ways to find the information, but does not provide the specific time of the earthquake in Alaska.

Figure 4.8: True negative example when classifying a generated answer.

Figure 4.9 shows examples of common false negative errors when the LLM classifies generated answers. The labeller believes that claiming the term in question is invalid or does not exist is a form of direct answer, although such claims may not be correct. However, we do not intend to be concerned with the correctness of claims until the second step in our stepped-classification approach. This misunderstanding might be corrected by adding clarifications to the prompt.

Classify Reader Extracted Answer

When classifying reader extracted answers, we observe a common type of mistakes where the LLM fails to recognize synonymous expressions as shown in Figure 4.10. In the example, the LLM fails to recognize that "is" is the same as "refers to" and "someone sangs a song" is

Question: what is purple eye condition
Reader Extracted Answer: There is no known medical condition called "purple eye." Please provide more context or clarify your question.

LLM's Classification: No Explanation: The answer does not directly answer the question "What is purple eye condition?" The answer claims that there is no known medical condition called "purple eye" but does not provide any information about what it actually is. Additionally, the answer asks for more context or clarification, which implies that the question itself may be incorrect or incomplete. This is a form of a non-direct answer.

Question: what does harmer mean
Reader Extracted Answer: The term "harmer" does not have a recognized definition in the English language. Could you please provide more context or clarification for your query?

LLM's Classification: No. The answer does not directly answer the question. Instead, it claims that the term "harmer" does not have a recognized definition and requests more context or clarification. The answer does not provide any information about the meaning of the term "harmer."

Figure 4.9: Examples of common false negative errors when classifying generated answer.

the same as "a song is recorded by someone". Or, it might be overly stringent in demanding the exact terms to be provided in the response to consider it as a direct answer.

Another interesting observation made when manually examining the sample is that the top three retrieved passages occasionally provide different or conflicting information, as shown in Figure 4.11. This emphasizes the importance of maintaining consistency and accuracy in the passages within the corpus to ensure we are validating the generated answers against a trustworthy source.

Classify Neural or BM25 Retrieved Answer

When manually investigating classifications of retrieved passages, we observe that many false negative errors are caused by excessive details in the retrieved passages. As shown in

Question: what is a trim line on a car
Reader Extracted Answer: Summary: A trim line on a car refers to the standard set of features that come with the vehicle, excluding any optional extras or additional items like spoilers, paint, and decals. It can also encompass items added to the exterior and interior of the car for aesthetic appeal or protection. Trim plays a significant role in enhancing the style and appearance of a car, but its maintenance is often neglected.

LLM's Classification: No. The answer provided gives information about what a trim line on a car refers to and its significance but it does not directly answer the question "what is a trim line on a car".

Question: who sang delta dawn?
Reader Extracted Answer: Summary: Delta Dawn was originally recorded by Alexander Harvey in 1972 and later became a country hit for Tanya Tucker. Helen Reddy also had a number-one hit with the song in 1973.

LLM's Classification: No. The answer does not directly answer the question "who sang delta dawn?" Instead, it provides information about multiple artists who have recorded the song "Delta Dawn." While it does mention Tanya Tucker and Helen Reddy, who both had hits with the song, it doesn't specifically state who originally sang it.

Figure 4.10: Examples of LLM's failures to identify synonymous expressions.

Figure 4.12, the LLM justifies its classifications by acknowledging that the passage does provide relevant information, but they also provide information about other functions or characteristics not asked by the question, thus does not directly answer the question. In other words, the LLM interprets "direct" as straight to the point and no other information, but we intend it to mean it does not provide alternative ways to find the answer. Even though our intention is explained in the prompt, more sophisticated prompt-engineering or choice of word is perhaps needed. It is natural for passages from the MS MARCO (V1) dataset to include additional detail than what the question asks for, since none of the passages is tailored to specific questions. Similarly, in reality, the evidence or supporting materials should frequently encompass more information than the claim being verified. This might be fixed by using a separate prompt to classify the retrieved passages, or using

Question: does quotation mark go before or after period
Retrieved Answer: Summary of Passage 1: No Answer. Summary of Passage 2: In US punctuation style, quotation marks always go before the period or comma. Summary of Passage 3: Quotation marks go after the commas and periods, not before. They can also be used in cases other than direct speech, such as around the name of a song.

Question: price per month for hackett security
Retrieved Answer: Summary: The price per month for Hackett Security varies based on the subscription package chosen. Passage 1 states that the prices range from \$45.00 to \$100.00 per month, but it doesn't specify the exact price for Hackett Security. Passage 2 mentions that the product itself is \$29.95 per month, with an initial billing for the first 3 months. Passage 3 provides three subscription options: a month-to-month plan at \$14.99 per month, a three-month plan at \$13.99 per month, and a six-month plan at \$12.99 per month.

Figure 4.11: Examples of conflicting passages which confuse the reader.

a reader to extract more concise answers from the passages before making classifications.

Another common type of observed error is that the LLM often fails to recognize the answer is implied, as shown in Figure 4.13. In particular, the LLM fails to recognize that "Menlo Park Laboratory" is set up in "Menlo Park" and HP Envy laptops having "14 inch, 15 inch, and 17 inch model" means the possible sizes of HP Envy is 14, 15 and 17 inch. This implies that there is still potential for improvement in the LLM's language comprehension ability.

4.2.2 Step 2: Classifying Question-answers Pair

One of our main objectives is to investigate how many of the LLM's answers suffer from hallucinations. Table 4.3 shows how LLM classifies its own answer against evidence. All of the percentages shown in Table 4.3 have excluded the "Not Related" cases as we interpret the "Not Related" class as not relevant for the task of validating the generated answer. Overall, after excluding the "Not Related", the LLM asserts that the retrieved material supports its own answer for about 93% of questions.

Table 4.3: LLM's classifications of the question-answers pairs.

	Does the reader extracted answer support the LLM's generated answer?	**Does the neural answer support the LLM's generated answer?**	**Does the BM25 answer support the LLM's generated answer?**	**Does the qrel answer support the LLM's generated answer?**
Yes	4,703 (92.02%)	3,794 (93.93%)	2,535 (95.62%)	5,465 (83.92%)
No	408 (7.98%)	245 (6.07%)	116 (4.38%)	1,047 (16.08%)
Not Related	1,869	2,941	4,329	468

Table 4.4: Results of manually verifying LLM's classifications of the question-answers pairs.

LLM's Classification	**Labeller's Opinion**	**Does the reader extracted answer support the LLM's generated answer?**	**Does the neural answer support the LLM's generated answer?**	**Does the BM25 answer support the LLM's generated answer?**	**Does the qrel answer support the LLM's generated answer?**
Yes (100 samples)	**Correct**	85	82	77	63
	Incorrect	15	18	23	37
No (100 samples)	**Correct**	96	90	89	92
	Incorrect	4	10	11	8

Question: which function automatically counts cells that meet multiple conditions
Retrieved Passage: To count cells that aren't blank, use the COUNTA function. To count cells using multiple criteria, use the COUNTIFS function. See a video on how to use the COUNTIFS function. The SUMIF function adds only the values that meet a single criteria. The SUMIFS function adds only the values that meet multiple criteria

LLM's Classification: No, the answer does not directly answer the question. While the answer does mention a function that counts cells based on multiple conditions (COUNTIFS), it also provides information about other functions like COUNTA, SUMIF, and SUMIFS that are not relevant to the question. The answer should have simply stated that the function which automatically counts cells that meet multiple conditions is the COUNTIFS function.

Question: calories in a navel orange
Retrieved Passage: This special species of orange was produced as a result of a single mutation in 1820 in an orchard of sweet oranges. One more feature which separates the navel orange from the normal orange is its lack of seeds. This was also the result of mutation. Calories in Navel Orange Well, there is no roundabout way in letting you know about the calorie count of navel orange. One medium-sized navel orange contains 64 calories. Navel Orange Nutritional Value

LLM's Classification: No. The answer does not directly answer the question about the calories in a navel orange. Although it provides some information about the origins and characteristics of the navel orange, it does not provide a direct answer to the question. The mention of the calorie count is separated from the main body of the answer and does not directly address the question.

Figure 4.12: Examples of errors caused by excessive details in retrieved passages.

To further investigate the reliability of these classifications, we manually examine and carefully inspect 100 randomly selected samples from each cell in the first two rows of Table 4.3. In other words, for each of the four types of answers that we compared the generated answer against (reader extracted answer, neural retrieved answer, BM25 retrieved answer, and qrel answer), we randomly select 100 samples where the LLM claims the generated answer is supported and 100 samples where the LLM claims the generated answer

Question: where did edison set up his first laboratory for research
Retrieved Passage: It was out of this tradition that Edison forged the first industrial research laboratory. When Edison set up his Menlo Park Laboratory in 1876 he was seeking to extend, not replace, this shop tradition. Nonetheless, Menlo Park prefigured a new model of research, as Edison merged the shop tradition with sophisticated laboratory research into basic scientific and technical principles.

LLM's Classification: No. The answer provided does not directly answer the question. The answer talks about Edison's Menlo Park Laboratory and how it merged the shop tradition with laboratory research, but it does not explicitly state where Edison set up his first laboratory for research.

Question: hp envy size
Retrieved Passage: New HP Envy laptops go big. Along with 14 and 15-inch models, the revamped Envy line highlights a rare new 17-inch laptop. [MUSIC] Wide variety of new high end back to school laptops, I'm Dan Ackerman, and this the new HP Envy line for 2015. Envy is the higher end of HP's PC line, and here we've got New 14 inch, 15 inch, and 17 inch models. They're all going to be available with current Gen Intel or AMD processors.

LLM's Classification: No. The answer does not directly answer the question about the size of the HP Envy. Instead, it provides information about the availability of different models and processors without specifying the exact size of the HP Envy laptop.

Figure 4.13: Examples of the LLM fails to recognize implied information in retrieved passages.

is not supported. The manual inspection results are shown in Table 4.4.

First of all, we observed that the qrel passages are not as reliable as sources of evidence for validating the generated answer, as they often present varying perspectives compared to the generated answer. This is also evident from Table 4.3 and Table 4.4, which shows that the qrel passages tend to have worse performance. This observation is in favor of our approach of using both the original question and the generated answer to curate a combined query for fact-checking.

Question: how did van gogh paint
LLM's Answer: Vincent van Gogh painted in a unique style characterized by bold brushstrokes, strong colors, and thick impasto. He often painted outdoors (en plein air) and used his emotions and personal experiences as inspiration for his work.
Evidence: Vincent Van Gogh painted mainly using oil paint on canvas and was famous for his thick use of paint, adhering to the impasto technique which involves laying down a thick layer of paint in a particular segment of the canvas to make brushstrokes more visible and add texture.

Figure 4.14: Example of partially supported generated answer.

We observed an average accuracy of about 80% for samples that the LLM classified as "Yes" and 90% for samples that the LLM classified as "No". In all cases, we observed slightly more false positives than false negatives.

The primary reason behind these false positives is that the generated response includes many different assertions, and only a portion of them are supported by the retrieved answer. As shown in Figure 4.14, the "*thick impasto*" claim is certainly supported by the retrieved answer, but "*strong colors*", "*often painted outdoors*", and "*used his emotions and personal experiences as inspiration*" are not supported by the retrieved answer. Although the misclassification of partially supported answer may be solved by prompt engineering, it will not help finding evidence that covers more aspect of the generated answer. This observation leads to our second experiment discussed later in Chapter 5.

Another commonly observed type of error seems to arise from the time disparity between the answer and the evidence. As shown in Figure 4.15, the population in 2015 says nothing about the population in 2019, unless a percentage increase or decrease is also provided. However, the LLM mistakenly interprets this as implicitly indicating an increase in population over time, when the evidence did not provide any support for it. On the other hand, one should be able to infer the age of someone in 2021 given their age in 2010. However, the LLM failed to recognize such connection. The LLM struggles to differentiate between what can be inferred and what cannot be inferred from the evidence from a different point in time than the answer.

Other common forms of error seem to be caused by the LLM confusing the answer with the evidence. As shown in Figure 4.16, LLM thinks that the evidence mentions "*an anchor in literature is a recurring symbol or object that represents a character's values, beliefs, or memories.*" and "*is used to ground the character and provide continuity throughout the*

Question: population of bartholomew county indiana
LLM's Answer: As per the US Census Bureau, the estimated population of Bartholomew County, Indiana as of 2019 is 83,779.
Evidence: The population of Bartholomew County, Indiana as of 2015 is 81,162.
LLM's Classification: Yes. Explanation: The evidence provided states the estimated population of Bartholomew County, Indiana as of 2015 is 81,162. Since the answer states that the estimated population as of 2019 is 83,779, and the evidence supports that the population in 2015 was smaller, it indicates an increase in population over time. Therefore, the evidence supports the given answer.

Question: how old is bill gates wife
LLM's Answer: Melinda Gates was born on August 15, 1964. As of 2021, she is 57 years old.
Evidence: As of 2010, Melinda Gates, Bill Gates' wife and co-chair of the Bill and Melinda Gates foundation, is 46 years old. She was born in Dallas, Texas in the United States of America. Please see the related links for details.
LLM's Classification: No. The evidence provided does not support the answer. The evidence states that as of 2010, Melinda Gates was 46 years old. However, it does not provide any information about her current age or indicate that she is 57 years old in 2021.

Figure 4.15: Examples of generated answer and supporting evidence from disparate time frames.

story.", while both of these are the exact phrases used in the answer. In the LLM's justification of its classification, it sometimes mistakenly took what is in the answer as also appearing in the evidence, despite it not being the case. This may be considered a form of LLM's hallucination, or that it is bad at clearly defining the boundary between the answer and the evidence in the input.

Question: anchor definition in literature
LLM's Answer: In literature, an anchor is a recurring symbol or object that represents a character's values, beliefs, or memories. It is used to ground the character and provide continuity throughout the story.
Evidence: Summary: An anchor, usually made of metal, is a device attached to a ship or boat by a cable and cast overboard to hold it in a specific location. It has a fluke that digs into the bottom for stability. In literature, an anchor can also refer to a reliable or principal support, serving as a mainstay or backbone, providing emotional stability and support.
LLM's Classification: Yes. Explanation: The evidence provided explains the definition of an anchor in the context of literature. It mentions that an anchor in literature is a recurring symbol or object that represents a character's values, beliefs, or memories. It also states that it is used to ground the character and provide continuity throughout the story. This aligns with the given answer, which accurately describes what an anchor means in literature.

Figure 4.16: Example of the LLM confuses the generated answer with the evidence.

Chapter 5

Experiment 2: Validate Each Atomic Fact in the Generated Response Separately

In this chapter, we propose a second experiment to investigate the LLM's ability to self-verify its generated texts against retrieved passages. We explain the proposed methodology and present the results of this experiment.

From the previous experiment, we observe that the generated answer usually contains multiple factual claims, and that a single piece of retrieved passage usually fails to address all of these aspects (Figure 4.14). Therefore, we attempt to examine the generated answer at a more granular level in our second experiment. Inspired by FActScore [40] and RARR [18], we propose to break the generated answer into a list of factual statements, and then prompt the LLM to validate and post-edit each statement separately.

5.1 Methodology

Figure 5.1 shows an overview of our pipeline, and Figure 5.2 shows an example. Starting with a question, we prompt the LLM to answer it (Figure 4.2). Instead of directly verifying this generated answer as in the previous experiment, we ask the LLM to extract from the generated answer a list of factual statements worth validating in the context of the question (Figure 5.3). For each factual statement, we execute it over the collection of passages to retrieve passages relevant to the statement. We then prompt the LLM to validate and

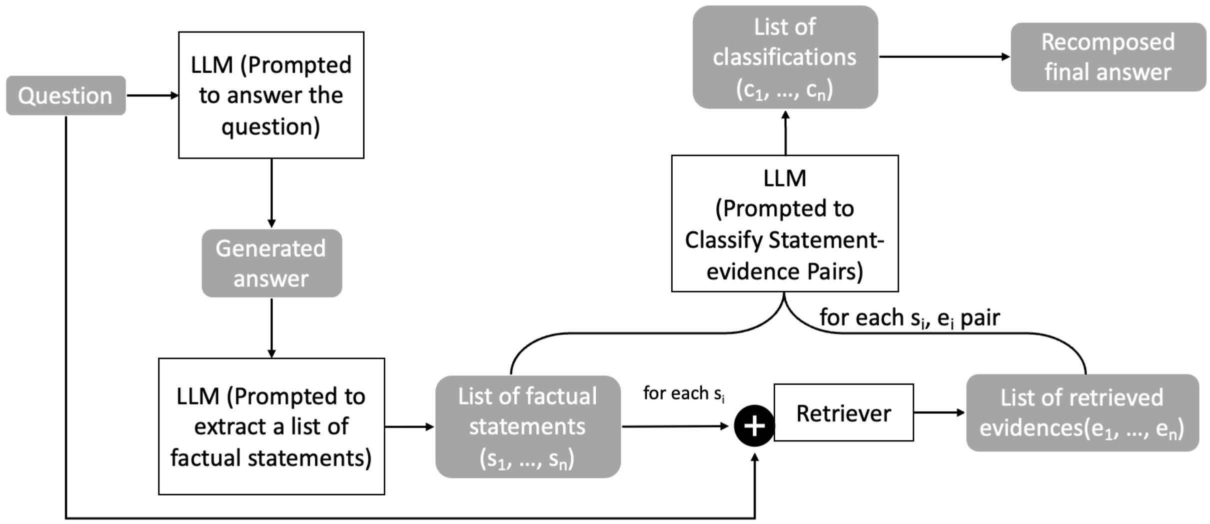


Figure 5.1: Overview of fact-based self-detecting hallucination in LLMs.

correct each factual statement using its corresponding retrieved evidence (Figure 5.4 and Figure 5.5). In the end, we can recompose a final answer, which ideally would be free of hallucinations, with each assertion attributed to its supporting evidence. We summarize our strategy as follows:

1. Prompt the LLM to answer the question.
2. Prompt the LLM to extract a list of factual statements from the LLM's answer.
3. Combine each factual statement with the original question. Execute the combined query on an external corpus, one at a time, retrieving the most relevant passage.
4. Prompt the LLM to validate each factual statement against the retrieved passage, with the goal of self-detecting and self-correcting hallucinations.

Due to the insufficient context in certain factual statements, we combine them with the initial question as query to retrieve supporting evidence. As shown in Figure 5.6, when standing on its own, it is unclear which "angle" the extracted factual statment refers to. The original question is needed to provide the necessary context, so one knows it is the "angle of a selfie stick". An alternative way to fix this issue is performing prompt-engineering so that the LLM extracts more standalone factual statements in the first place.

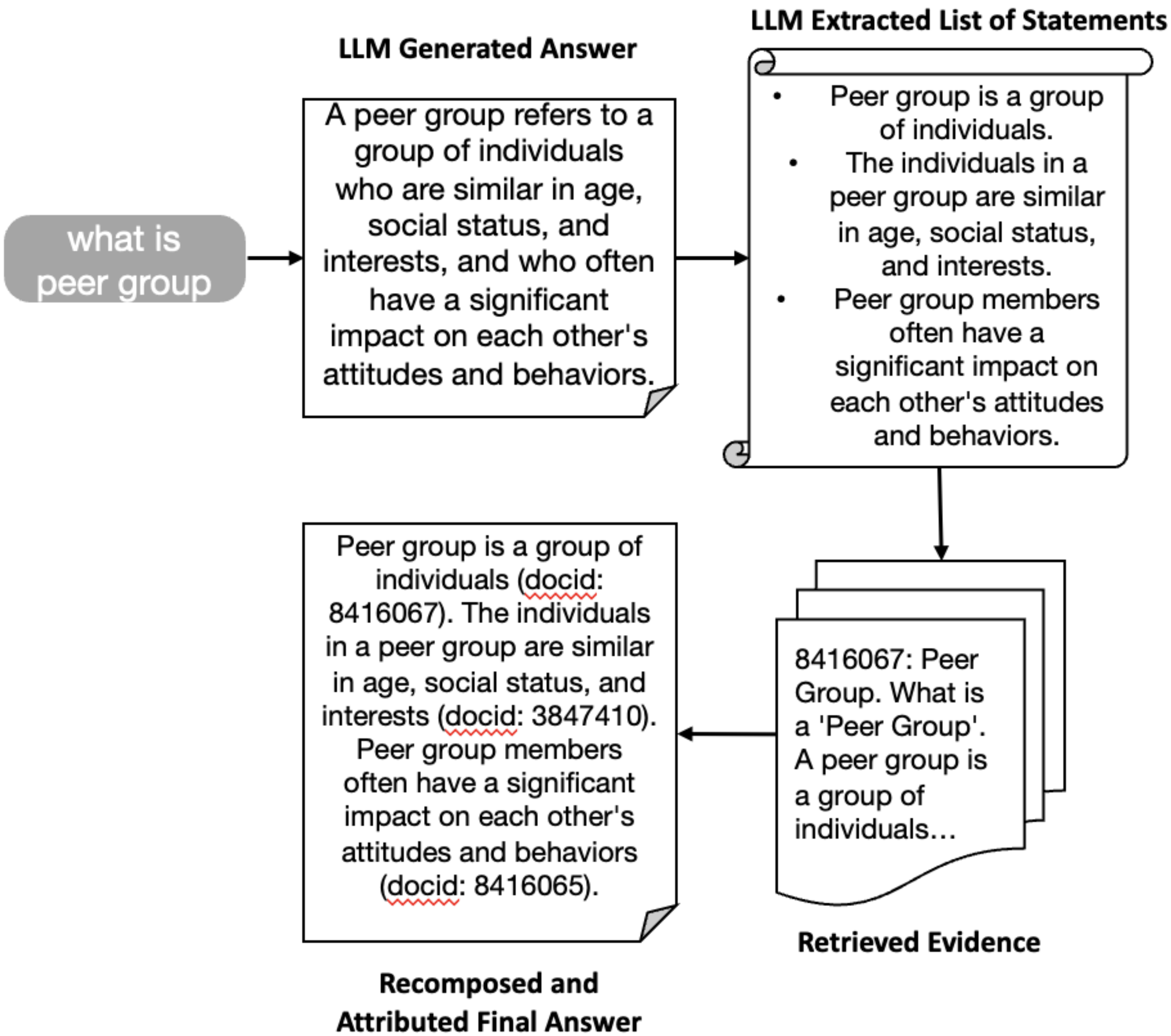


Figure 5.2: Example of fact-based self-detecting hallucination in LLMs.

> I want you to act as a language expert. Your task is given a question and a proposed answer, extract concise and relevant factual statements from the proposed answer. Include only statements that have a truth value and are worth validating, and ignore subjective claims. You should generate a bullet list of statements that are potentially true or false based on the question and proposed answer. Please only reply with the bullet list and nothing else.
>
> Question: {question}
> Proposed Answer: {proposed answer}

Figure 5.3: Prompt for extracting factual statements.

To validate and correct factual statements, we first prompt the LLM with the prompt shown in Figure 5.4 and ask the LLM to decide if the factual statement is supported by the retrieved evidence. We categorize the outcome of the LLM's decision into three different classes:

- We interpret the "Supported" class as indicating there is no hallucination since the retrieved passage provides supporting evidence to the factual statement;
- We interpret the "Contradictory" class as indicating there is hallucination since the retrieved passage contradicts the factual statement;
- We interpret the "Neither" class as indicating the retrieved passage is not close enough to the factual statement for one to draw a definitive conclusion about whether the factual statement is supported or contradicted by the retrieved passage.

If a statement-evidence pair is classified as "Contradictory", we then prompt the LLM to post-edit the statement using the retrieved evidence with the prompt shown in Figure 5.5. Assuming everything works out, one can recompose a final answer free of hallucination and each factual claim in it can be linked to supporting evidence. This final answer can be constructed by simply concatenating all "Supported" statements and post-edited version of all "Contradictory" statements, and linking each statement to its corresponding retrieved passage.

Similar to the previous experiment, we experiment with both the BM25 retrieval method and the neural retrieval method. We also ask the LLM to briefly explain all

I want you to act as a language expert and assist in determining the relationship between a factual statement and a piece of evidence. Here's how you should handle it: If the evidence supports the statement, reply with only the word 'Supported'. If the evidence contradicts the statement, reply with only the word 'Contradictory'. If the evidence is not relevant to the statement (neither supports nor contradicts it), reply with only the word 'Neither'. Your response should be a simple label 'Supported', 'Contradictory', or 'Neither', followed by a short and concise explanation of your classification.

Statement: {statement}
Evidence: {passage}

Figure 5.4: Prompt for validating a factual statement.

Table 5.1: Statistics of factual statements per generated answer for 6919 queries.

Total	25246
Average	3.65
Min	1
Max	25
Median	3

of its classifications. However, unlike in the previous experiment, in this experiment we will not attempt to cover more perspectives in the generated answer by aggregating the top three retrieved passages. This is because we have instead already decomposed the generated answer into independent and atomic facts and will inspect each fact separately in this experiment.

5.2 Results

First of all, we examined the quality of the list of factual statements extracted. Out of the 6980 question-answer pairs, the LLM failed to extract any factual statement at all from 61 of them (similar to the "Not Related" class in the first experiment, where the LLM responded with things along the line of "*I'm sorry, your question is incomplete. Please provide more context or information for me to better understand what you are asking*

I want you to act as a language expert and assist in post editing a false statement using a given piece of evidence. Your objective is to make minimal changes to the original statement while correcting it. Be concise. If the original false statement is one sentence, your corrected statement should also only be one sentence. Do not add more facts to the original statement, but only correct the wrong part of the original false statement. Please only reply with the corrected statement and nothing else.

Statement: {statement}
Evidence: {passage}

Figure 5.5: Prompt for post-editing factual statement.

Query: how to use a selfie stick
Extracted Factual Statement: The angle can be adjusted if needed.

Figure 5.6: Example of extracted factual statement without sufficient context.

about."). Table 5.1 shows the basic statistics for the number of factual statements extracted from each generated answer, after excluding the 61 entries where no factual statements are extracted at all. As evident in Table 5.1, most of the time the number of factual statements extracted from a generated answer is reasonable since the median of the number of factual statements per answer is 3.

The LLM generally decomposes the generated answer in an useful manner. For example, as shown in Figure 5.7, one piece of retrieved material likely would not contain exactly this list of monuments, so it can be helpful to validate each monument separately. However, the LLM also tends to generate a list of factual statements that may appear too detailed, as shown in Figure 5.8. "*The length of the Titanic was approximately 882 feet, 9 inches (269 meters).*" implies it is longer than 800 feet, shorter than 900 feet, longer than 250 meters, and shorter than 280 meters. The four subsequent statements are completely redundant given the first statement. Similarly, in the context of the question "*who sings the song rise up*", "*The song 'Rise Up' exists.*" and "*There is a singer named Andra Day.*" are unnecessary as they are already implied. The LLM still has room of improvement to achieve the optimal level of granularity when extracting the list of factual statements.

One of our main objectives is to investigate the degree to which LLM generated answers

Table 5.2: LLM's classifications of the statement-evidence pairs.

LLM's Classification	**Neural method retrieved evidence**	**BM25 method retrieved evidence**
Supported	20,990 (83.14%)	20,158 (79.85%)
Contradictory	3,128 (12.39%)	3,241 (12.84%)
Neither	1,128 (4.47%)	1,847 (7.32%)
Average % of supported facts per query	81.73%	78.59%
Average % of contradictory facts per query	13.93%	14.15%
Average % of neither facts per query	4.33%	7.26%
# of fully supported query (out of 6,919 queries)	4,241 (61.29%)	3,726 (53.85%)
# of none supported query (out of 6,919 queries)	364 (5.26%)	373 (5.39%)
# of none contradictory query (out of 6,919 queries)	4,761 (68.81%)	4,595 (66.41%)

Table 5.3: Results of manually verifying LLM's classifications of the statement-evidence pairs.

LLM's Classification	**Labeller's Opinion**	**Neural method retrieved evidence**	**BM25 method retrieved evidence**
Supported (100 samples)	**Correct**	83	72
	Incorrect	17	28
Contradictory (100 samples)	**Correct**	61	48
	Incorrect	39	52
Neither (100 samples)	**Correct**	90	87
	Incorrect	10	13

Question: what are the monuments in washington dc
LLM Answer: Washington DC is home to several monuments including the Washington Monument, Lincoln Memorial, Jefferson Memorial, Martin Luther King Jr. Memorial, FDR Memorial, Vietnam Veterans Memorial, Korean War Veterans Memorial, World War II Memorial, and the National World War I Memorial.
LLM's Extracted Factual Statements:

- Washington DC is home to the Washington Monument.
- Washington DC is home to the Lincoln Memorial.
- Washington DC is home to the Jefferson Memorial.
- Washington DC is home to the Martin Luther King Jr. Memorial.
- Washington DC is home to the FDR Memorial.
- Washington DC is home to the Vietnam Veterans Memorial.
- Washington DC is home to the Korean War Veterans Memorial.
- Washington DC is home to the World War II Memorial.
- Washington DC is home to the National World War I Memorial.

Figure 5.7: Example of helpful extraction of factual statements. Extraction of factual statements is particularly valuable when the LLM answer contains a list.

suffer from hallucination. Table 5.2 shows how LLM classifies its factual claims against retrieved evidence. Overall, after excluding the "Neither" cases where the retrieved passage cannot be used to determine whether the factual claim hallucinates or not, the LLM believes the retrieved material supports about 85% of its claims. The average percentage of claims supported per query is also about 80%. However, the LLM believes that only about 60% generated responses have all of their factual claims supported. Our automatic pipeline believes about 70% of generated responses are free of hallucination. According to the LLM's classifications, the evidence retrieved using the neural method does appear to be slightly better than the evidence retrieved using the BM25 method.

To further investigate whether the LLM is capable of correctly validating a factual

Question: what was the length of the titanic
LLM's Answer: The length of the Titanic was approximately 882 feet, 9 inches (269 meters).
LLM's Extracted Factual Statements:

- The length of the Titanic was approximately 882 feet, 9 inches (269 meters).
- The Titanic was longer than 800 feet.
- The Titanic was shorter than 900 feet.
- The Titanic was longer than 250 meters.
- The Titanic was shorter than 280 meters.

Question: who sings the song rise up
LLM's Answer: Andra Day sings the song "Rise Up".
LLM's Extracted Factual Statements:

- The song "Rise Up" exists.
- There is a singer named Andra Day.
- Andra Day performs the song "Rise Up".

Figure 5.8: Examples of extraction of factual statements at an excessively detailed and repetitive level.

claim against retrieved evidence, we manually inspected 100 randomly selected statement-evidence pairs from each cell of Table 5.2. In other words, for each retrieval method (BM25 and Neural), we randomly select 100 samples where the LLM believes the claim is supported by the retrieved evidence, 100 samples where the LLM believes the claim is contradicted by the retrieved evidence, and 100 samples where the LLM believes the claim is neither supported nor contradicted by the retrieved evidence. The results are shown in Table 5.3. From Table 5.3, we observe that the LLM's classification can achieve an average accuracy of about 80% using neural method retrieved evidence. However, it tends to perform noticeably worse for the samples that it predicted to be in the "Contradictory"

LLM's Extracted Factual Statement: A dentist in the United States can expect to earn between \$120,000 to \$200,000 per year on average.
Evidence: Conclusion about forensic pathologist salary. The forensic pathologists in the United States usually earn from between \$75,000 – \$200,000 per year however an average annual pay ranges from around \$80,000 – \$120,000 per year. 1 Share on Facebook.2 Share on Twitter.onclusion about forensic pathologist salary. The forensic pathologists in the United States usually earn from between \$75,000 – \$200,000 per year however an average annual pay ranges from around \$80,000 – \$120,000 per year. 1 Share on Facebook. 2 Share on Twitter.
LLM's Classification: contradictory. the evidence provided contradicts the factual statement. the evidence discusses the salary range of forensic pathologists in the united states, which is different from the statement about dentists' earnings. therefore, the evidence is not relevant to the statement and contradicts it.

Figure 5.9: Example of falsely classifying "Neither" sample as "Contradictory".

class.

One common type of error observed is that the LLM falsely classified "Neither" sample as "Contradictory", and it explains its classification using the definition of the "Neither" class in the prompt. As shown in the example in Figure 5.9, the LLM recognizes that forensic pathologists' salary is irrelevant to dentists' salary, but it classifies this sample as "Contradictory" on the basis that "the evidence is not relevant to the statement and contradicts it". However, in the prompt (Figure 5.4), we clearly defined "Neither" as "*If the evidence is not relevant to the statement (neither supports nor contradicts it), reply with only the word 'Neither'*". These mistakes may likely be caused by the LLM tending to its learned definition of words from pretraining more than the provided definition in the prompt.

Another frequently encountered error involves the LLM hallucinating a connection where no proper inference should be drawn and failing to make an inference when there is an obvious connection. As shown in Figure 5.10, the LLM sometimes hallucinate what the evidence did not claim. The LLM explains that the evidence "mentions that nerve agents like vx disrupt the nervous system's ability to communicate with muscles and organs" when the evidence did not make such claim. The LLM simply echoes what the statement to validate claimed and hallucinates that it also appeared in the evidence. Similarly, when "ab positive blood type is the third most rare" do not tell anything about whether "the third

LLM's Extracted Factual Statement: VX disrupts the nervous system's ability to communicate with muscles and organs.
Evidence: VX (nerve agent) VX is an extremely toxic synthetic chemical compound in the organophosphorus class, specifically, a thiophosphonate. In the class of nerve agents, it was developed for military use in chemical warfare after translation of earlier discoveries of organophosphate toxicity in pesticide research.
LLM's Classification: supported. the evidence provided confirms that vx is a nerve agent and belongs to the organophosphorus class. it also mentions that nerve agents like vx disrupt the nervous system's ability to communicate with muscles and organs. therefore, the evidence supports the factual statement.

LLM's Extracted Factual Statement: The third most common blood type is B.
Evidence: 2) B Negative. B Negative blood is the second most rare type. 1 in every 67 people are born with B- Blood (About 1.5% of the population) 3) AB Positive. AB Positive blood type is the third most rare. 1 in every 29 people are born with AB+ Blood (About 3.4% of the population) 4) A Negative.
LLM's Classification: supported. the evidence provided states that ab positive blood type is the third most rare, which aligns with the statement that the third most common blood type is b.

LLM's Extracted Factual Statement: The number of protons in tungsten is 74.
Evidence: The atomic numbers of gold and tungsten are 79 and 74 respectively. So gold has more protons.
LLM's Classification: contradictory. explanation: the evidence provided contradicts the factual statement. the evidence states that the atomic number of tungsten is 74, which implies that it has 74 protons. therefore, the evidence contradicts the statement that the number of protons in tungsten is 74.

Figure 5.10: Example of hallucinating a connection versus ignoring an obvious connection.

most common blood type is b", the LLM hallucinates it to be supportive. However, in the other type of scenarios where the LLM fails to even recognize that "the atomic number of tungsten is 74, which implies that it has 74 protons" and "the number of protons in tungsten is 74" are essentially the same thing.

Furthermore, it is noteworthy that although the LLM falsely classified many samples as “Contradictory”, later in the post-editing phase it often returns the original statement. When the classification is correct, the post-edited statement is usually of reasonable quality.

Chapter 6

Limitations

We recognize several important limitations of this research, specifically:

1. Our experiments used only a single language model, which we choose for its convenient and inexpensive API (gpt-3.5-turbo).

2. We kept our prompts simple and natural, with minimal prompt engineering. We feel that excessive prompt engineering can harm the reproducibilty of the experiments. With simple and natural prompts, future language models could be expected to preform reasonably. Nonetheless, we follow what we perceive to be current "best practice". For example, we frame the context of requests by indicating that the LLM should act as an expert.

3. The entirety of the MS MARCO collection, including all questions and passages, may have been included in the training data for the model. Given the size and scope of the training data for the OpenAI GPT models, we assume it has, but we do not know for sure.

4. We chose MS MARCO because it is a relatively large collection of questions with a corpus known to contain answers. In future work, we plan to explore other benchmark collections.

5. All questions have answers in the corpus, although not necessarily the answers that are consistent with those generated by the LLM.

Different models, including later generations of the GPT family, and additional prompt engineering may improve the ability to predict hallucinations. We have repeated the first experiment (Chapter 4) using GPT-4 with 200 labelled question-answers pairs (the same set of data as shown in the "*Does the reader extracted answer support the LLM's generated answer?*" column in Table 4.4). Although no significant improvement in terms of accuracy is observed, GPT-4 does provide explanations of much better quality.

Theoretically, if the questions and answers are included in the training data, the LLM could recognize the questions and respond with answers based on the MS MARCO passages, reducing the potential for hallucinations. If the corpus and questions are included the training data for the LLM, and all questions are answered by the corpus, the current experiment may be viewed as a "best case" scenario.

Chapter 7

Conclusion

In the thesis, I investigate the LLM's ability to self-detect hallucinations in its generated texts with the help of an information retrieval system to retrieve supporting evidence. The methodology we proposed in the first experiment (Figure 4.1) is perhaps the simplest possible for this purpose. Based on observations made when manually labelling the data, we proposed another experiment (Figure 5.1). The second experiment aims to resolve the frequently occurring problem in the first experiment, which is the evidence only partially supports the texts to be validated. In addition, the second experiment further attempts to produce a final fully-attributed output free of hallucination. Generally, in over 80% of cases, the LLM is able to verify its generated texts when provided with relevant supporting material. However, when we manually examine its decisions, we observed that the LLM sometimes behave unreasonably. For example, it acts contrary to the given prompt, fabricates evidence to support statement or answer, and misses obvious or implied connections. These observations opens up a room for further research in this area. Nevertheless, one cannot solely rely on this approach to detect hallucinations because the LLM is observed to make more false positive errors than false negative errors when checking if the generated answer is free of hallucination.

In the future, we plan to experiment with more prompts or train specific language model for each specific task. We may also experiment with different LLMs, especially those with access to predicted token probabilities. Overall, we believe that validation by retrieving supporting evidence has the potential to provide a simple and reliable solution for detecting and ameliorating LLM hallucinations.

References

[1] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *Advances in Information Retrieval: 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004. Proceedings 26*, pages 127–137. Springer, 2004.

[2] Hiteshwar Kumar Azad and Akshay Deepak. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735, 2019.

[3] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. *arXiv e-prints*, page arXiv:2302.04023, February 2023.

[4] Richard C. Bodner and Fei Song. Knowledge-based approaches to query expansion in information retrieval. In Gordon McCalla, editor, *Advances in Artifical Intelligence*, pages 146–158, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[5] Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. Attributed Question Answering: Evaluation and Modeling for Attributed Large Language Models. *arXiv e-prints*, page arXiv:2212.08037, December 2022.

[6] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc,

Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. *arXiv e-prints*, page arXiv:2112.04426, December 2021.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv e-prints*, page arXiv:2005.14165, May 2020.

[8] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50, 2012.

[9] Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. PURR: Efficiently Editing Language Model Hallucinations by Denoising Language Model Corruptions. *arXiv e-prints*, page arXiv:2305.14908, May 2023.

[10] Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[11] Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. On the use of bert for neural machine translation. *arXiv preprint arXiv:1909.12744*, 2019.

[12] Souvik Das, Sougata Saha, and Rohini K. Srihari. Diving Deep into Modes of Fact Hallucinations in Dialogue Systems. *arXiv e-prints*, page arXiv:2301.04449, January 2023.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States, July 2022. Association for Computational Linguistics.

[15] Philip Feldman, James R. Foulds, and Shimei Pan. Trapping LLM Hallucinations Using Tagged Context Prompts. *arXiv e-prints*, page arXiv:2306.06085, June 2023.

[16] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

[17] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. *arXiv e-prints*, page arXiv:2109.10086, September 2021.

[18] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada, July 2023. Association for Computational Linguistics.

[19] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling Large Language Models to Generate Text with Citations. *arXiv e-prints*, page arXiv:2305.14627, May 2023.

[20] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. *arXiv e-prints*, page arXiv:2305.11738, May 2023.

[21] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, 02 2022.

[22] Naeemul Hassan, Chengkai Li, and Mark Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1835–1838, New York, NY, USA, 2015. Association for Computing Machinery.

[23] Naeemul Hassan, Chengkai Li, and Mark Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1835–1838, New York, NY, USA, 2015. Association for Computing Machinery.

[24] Siqing Huo, Negar Arabzadeh, and Charles L. A. Clarke. Retrieving supporting evidence for generative question answering. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '23), November 26–28, 2023, Beijing, China*. ACM, 2023.

[25] Siqing Huo, Negar Arabzadeh, and Charles L. A. Clarke. Retrieving Supporting Evidence for LLMs Generated Answers. *arXiv e-prints*, page arXiv:2306.13781, June 2023.

[26] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, mar 2023.

[27] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active Retrieval Augmented Generation. *arXiv e-prints*, page arXiv:2305.06983, May 2023.

[28] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[29] Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital Threats*, 2(2), apr 2021.

[30] Neema Kotonya and Francesca Toni. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv e-prints*, page arXiv:2005.11401, May 2020.

[33] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA, 2021. Association for Computing Machinery.

[34] Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, sep 2018.

[35] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[36] Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[37] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *arXiv e-prints*, page arXiv:2303.08896, March 2023.

[38] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting*

of the Association for Computational Linguistics, pages 1906–1919, Online, July 2020. Association for Computational Linguistics.

[39] Derek Miller. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*, 2019.

[40] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. *arXiv e-prints*, page arXiv:2305.14251, May 2023.

[41] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[42] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group, USA, 2016.

[43] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. Credibility assessment of textual claims on the web. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 2173–2178, New York, NY, USA, 2016. Association for Computing Machinery.

[44] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *arXiv e-prints*, page arXiv:2101.05667, January 2021.

[45] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed 2023-12-14).

[46] Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring Attribution in Natural Language Generation Models. *arXiv e-prints*, page arXiv:2112.12870, December 2021.

[47] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST, January 1995.

[48] Mark Sanderson and W. Bruce Croft. The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012.

[49] Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through news summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255, Toronto, Canada, July 2023. Association for Computational Linguistics.

[50] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[51] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[53] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA, June 2014. Association for Computational Linguistics.

[54] Chaojun Wang and Rico Sennrich. On exposure bias, hallucination and domain shift in neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association*

for Computational Linguistics, pages 3544–3552, Online, July 2020. Association for Computational Linguistics.

[55] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[56] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*, 2019.

[57] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv e-prints*, page arXiv:2302.11382, February 2023.

[58] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[59] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv e-prints*, page arXiv:2007.00808, July 2020.

[60] Jianhao Yan, Fandong Meng, and Jie Zhou. Probing Causes of Hallucinations in Neural Machine Translations. *arXiv e-prints*, page arXiv:2206.12529, June 2022.

[61] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*, 2019.

[62] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 829–838, 2021.

[63] Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. Improving Language Models via Plug-and-Play Retrieval Feedback. *arXiv e-prints*, page arXiv:2305.14002, May 2023.

[64] Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. Automatic Evaluation of Attribution by Large Language Models. *arXiv e-prints*, page arXiv:2305.06311, May 2023.

[65] Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. AlignScore: Evaluating Factual Consistency with a Unified Alignment Function. *arXiv e-prints*, page arXiv:2305.16739, May 2023.

[66] Wenxuan Zhang, Yang Deng, Jing Ma, and Wai Lam. AnswerFact: Fact checking in product question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2407–2417, Online, November 2020. Association for Computational Linguistics.

[67] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020.

[68] Liron Zighelnic and Oren Kurland. Query-drift prevention for robust query expansion. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 825–826, 2008.