

Friedrich Schiller University Jena
Faculty of Mathematics and Computer Science
Degree Programme Computational and Data Science, M.Sc.

Answering Health-Related Questions using Trusted Sources

Master's Thesis

Lukas Zeit-Altpeter
Born Apr. 7, 1997 in Bonn

Matriculation Number 200283

1. Referee: Prof. Dr. Matthias Hagen
2. Referee: Jan Heinrich Reimer, M.Sc.

Submission date: May 8, 2024

Acknowledgements

I would like to thank Marek Sierts, Adrian Mitomi, Joel Hankiewicz, Julian Klüber, Lander Haag, and Finn for their annotation work. Thanks also go to Dr. Anja Röske and Lukas for their help with the expert study. This work would not have been possible without the countless hours of consultation with Dr. Alexander Bondarenko and Jan Heinrich Reimer! Last but not least, I would like to thank Lukas Leslie for his amazing support throughout the project.

Abstract

We propose a retrieval-augmented generation system for answering non-factual health-related consumer questions. We adapt the existing MashQA dataset for our use case and perform evaluations at each step of our system. Our system consists of a retrieval component that combines a lexical retriever with dense rerankers. We find no benefit in using any query preprocessing steps for the retrieval component. The generation component uses a black-box approach that allows us to use any large language model. Our work focuses on optimizing the prompts for the generation component. We evaluate the usage of two different language models, Mistral7B and GPT4, for answer generation.

The generation results are evaluated both automatically and manually in an expert study. The end-to-end evaluation results for our system show that large language models do not necessarily benefit from retrieval augmentation. We observe a slight improvement in answer quality for a Mistral7B-based generator, while retrieval augmentation has no positive impact on a GPT4-based generator. The evaluation of non-factual question answering is challenging because there are usually multiple correct answers that can be formulated in an infinite number of ways. Our evaluation using both automatic and manual expert judgements allows us to compare how the two evaluation approaches agree. We find that automated metrics mainly check the alignment of the generated answers with the given reference answer, while they are not able to capture the quality of the generated answers themselves in detail. However, we find that some automatic metrics are still able to capture general trends regarding the quality of the generated answers.

Zusammenfassung

Wir schlagen ein Retrieval-Augmented-Generation System zur Beantwortung nicht-faktischer gesundheitsbezogener Verbraucherfragen vor. Wir passen den bestehenden MashQA-Datensatz für unseren Anwendungsfall an und führen Evaluierungen für jeden Schritt unseres Systems durch. Unser System besteht aus einer Retrieval-Komponente, die einen lexikalischen Retriever mit neuronalen Rerankern kombiniert. Wir haben keinen Nutzen in der Verwendung von Query-Preprocessing für die Abfragekomponente gefunden. Für die Generierung der Antworten nutzen wir einen Black-Box-Ansatz, der es uns ermöglicht, beliebige Sprachmodelle zu verwenden. Unsere Arbeit befasst sich mit der Optimierung der Prompts für die Generierungskomponente. Wir evaluieren die Verwendung von zwei verschiedenen Sprachmodellen, Mistral7B und GPT4, für die Generierung von Antworten.

Die Generierungsergebnisse werden sowohl automatisch als auch manuell durch Expert:innen evaluiert. Die End-to-End-Evaluierungsergebnisse für unser System zeigen, dass große Sprachmodelle nicht unbedingt von einer Retrieval-Erweiterung profitieren. Wir beobachten eine leichte Verbesserung der Antwortqualität für einen Mistral7B-basierten Generator, während die Retrieval-Erweiterung keinen positiven Einfluss auf einen GPT4-basierten Generator hat. Die Bewertung der Beantwortung nicht-faktischer Fragen stellt eine besondere Herausforderung dar, da es in der Regel mehrere richtige Antworten gibt, die auf unendlich viele Arten formuliert werden können. Unsere Evaluierung unter Verwendung von automatischen und manuellen Bewertungen ermöglicht es uns zu vergleichen, inwieweit beide Ansätze zur Evaluierung zu ähnlichen Ergebnissen führen. Wir stellen fest, dass automatische Metriken hauptsächlich die Übereinstimmung der generierten Antworten mit gegebenen Referenzantworten überprüfen, während sie nicht in der Lage sind, die Qualität der generierten Antworten selbst detailliert zu erfassen. Wir stellen jedoch fest, dass die automatischen Metriken dennoch in der Lage sind, allgemeine Trends in der Qualität der generierten Antworten zu erfassen.

Contents

1	Introduction	1
1.1	Generative AI in the Medical Domain	2
1.2	Retrieval-Augmented Generation	3
1.3	Outline	4
2	Related Work	5
2.1	Medical Information Retrieval	5
2.2	Question Answering	6
2.2.1	Generative Question Answering	8
2.2.2	Evaluation of Question Answering Systems	8
2.2.3	Question Answering in the Medical Domain	10
2.3	Summary	13
3	Dataset	14
3.1	Existing Datasets	14
3.2	Creating the MashTREC Dataset	16
3.2.1	Creating a Training Dataset	17
3.2.2	Characteristics of MashTREC	18
4	Retrieval	21
4.1	Retrieving from Reputable Sources	21
4.2	Preprocessing	23
4.2.1	Essential Term Extraction	24
4.2.2	Sequence-to-Sequence Preprocessing	26
4.2.3	Comparing Preprocessing Techniques	32
4.2.4	Ablation: Query Preprocessing on TREC 2022 Health Misinformation Track	33
4.3	Creating a Retrieval Pipeline	35
4.4	Evaluation	36
4.4.1	Metrics	37
4.4.2	Grid Search	40

4.4.3	Test Set	44
4.4.4	Evaluation on TREC 2022 Health Misinformation	46
4.5	Summary	48
5	Generating Long Form Answers	49
5.1	Prompt Engineering	49
5.2	Evaluation	51
5.2.1	Automatic Metrics	52
5.2.2	Grid Search	54
5.2.3	Test Set Evaluation	54
5.2.4	Ablation: Generation using MashQA’s context	66
5.3	Summary	67
6	Conclusion	68
6.1	Findings and Contributions	68
6.2	Pitfalls	70
6.3	Future Work	70
	Bibliography	72
	Appendices	i
A	Distribution of Categories for MashTREC Questions	ii
B	Guidelines for Annotation of Query-Question Matches	iii
C	Guidelines for Relevance Judgements for Training Set	v
D	Guidelines for Relevance Judgements for Test Set	viii
E	Guidelines for Preference Judgements on Generated Answers	xi
F	System Prompts Used in the Grid Search	xii
G	Prompt for the Generation of Long Form Answers	xiii
H	Prompt for the Generation of Long Form Answers without Context	xv
I	Prompt for the Completeness Metric	xvi
	Declaration of Academic Integrity	xviii

Chapter 1

Introduction

While the topic of answering health-related questions by consumers has been studied in the past, the use of retrieval-augmented generation for this task is not well explored. In this work, we introduce a system for answering non-factual consumer questions that require long-form answers. We develop and evaluate a retrieval augmented generation pipeline for this specific task.

More than half (55%) of EU citizens use the web to look up medical information [Eurostat, 2022]. For medical laypersons, the reasons can range from getting additional information about a diagnosis, to confirming a doctor’s recommended treatment, to learning what questions to ask doctors before starting medical treatment [Harvey et al., 2017]. Independent research on medical topics can be beneficial to consumers, for example, when traditional medical care is not available due to a lack of specialists. In addition, there is no social pressure when researching online, so consumers are more likely to ask questions about symptoms they may find embarrassing [Chu et al., 2017]. While there is a vast amount of medical information available online, a large number of trustworthy websites are aimed at medical professionals and use terminology that is not easily understood by consumers [Kim et al., 2018]. This language mismatch is also relevant in the formulation of search queries, where users may lack the technical terms to find trustworthy and relevant documents. Online forums that allow non-professionals to ask questions in natural language, where either fellow consumers¹ or medical experts provide answers,² can help overcome language barriers and provide solutions to highly specific and individualized questions. However, when it comes to lengthy consumer-contributed answers based on personal experience, a notable limitation in terms of clinical validity has been observed [Kanthawala et al., 2016].

¹such as Quora: <https://www.quora.com/topic/Health>

²<https://www.webmd.com>

Figure 1.1: Example of ChatGPT answering with inappropriate information. Haver et al. [2023] note that the Centers for Disease Control and Prevention (CDC) has not issued recommendations regarding COVID-19 and mammograms.

Question: Do I need to plan my mammogram around my COVID vaccination?

ChatGPT’s Response: The Centers for Disease Control and Prevention (CDC) and the Society of Breast Imaging recommend that women schedule their screening mammogram either before getting the COVID-19 vaccine or wait 4-6 weeks after the final dose of the COVID-19 vaccine. This is because the vaccine can cause temporary swelling of the lymph nodes in the armpit on the side where the shot was given, which can show up on a mammogram and be mistaken for a breast abnormality, leading to additional testing and potential anxiety.

1.1 Generative AI in the Medical Domain

One possibility to offer consumers individualized answers is the use of generative AI. In November 2022, OpenAI unveiled its transformer-driven chatbot named *ChatGPT*³, which reached 100 million users within two months. Although trained as a general-purpose model with no specific focus on medical topics, the tool has become relevant in the medical field, with both medical professionals and consumers using it to generate answers to their health-related questions. This has led to the coining of the term *Dr. ChatGPT*, alluding to *Dr. Google*. The latter is used to describe the phenomenon of patients looking up medical information using the search engine instead of contacting a medical professional [Van Bulck and Moons, 2023]. Both the GTP3.5 model and its successor GPT4, which form the basis of ChatGPT, have been shown to approximate expert-level medical knowledge encoded in their weights, achieving passing grades on the United States Medical Licensing Examinations [Kung et al., 2023, Thirunavukarasu et al., 2023b]. When comparing answers provided by experts in an online question-and-answer forum with those generated by ChatGPT, doctors rated the generated answers as higher quality and more empathetic to the patient [Ayers et al., 2023]. Large language models (LLMs), such as Google’s Med-PaLM [Singhal et al., 2023b], that were specifically trained for the medical domain have been released.

While the potential for medical use of LLMs has been demonstrated, it

³<https://chat.openai.com>

has also been noted that there are still risks associated with using LLMs as a provider of medical information: The overall performance of ChatGPT varies greatly depending on the specific topic being asked about [Thirunavukarasu et al., 2023a]. Rao et al. [2023] note that given the high-risk use case, even small errors can lead to adverse patient outcomes. Figure 1.1 shows an example of ChatGPT responding with outdated and potentially dangerous information for breast cancer patients. Walker et al. [2023] note that factual errors made by ChatGPT are likely due to the fact that its training data is not limited to trustworthy (medical) documents, and recommend that LLMs use only peer-reviewed sources for generation and provide a bibliography so that users can confirm the information provided. In addition, it was noted that LLMs generally operate as a black box and do not provide transparency on how the answer was generated, making it difficult for the consumer to judge the trustworthiness of the answer [Thirunavukarasu et al., 2023b].

1.2 Retrieval-Augmented Generation

One way to address the issue of trustworthiness is to use retrieval-augmented generation (RAG) [Lewis et al., 2020]. Instead of generating text purely based on the input, RAG systems first retrieve relevant documents and then generate answers based on the retrieved documents. While RAG can be implemented using either specific model architectures [Izacard and Grave, 2021] or fine-tuned LLMs [Lewis et al., 2020], the usage of off-the-shelf models [Lazaridou et al., 2022] has also been shown to be an effective way of reducing the risk of generating misinformation. In the latter case, retrieved documents are concatenated to the original prompt and fed into the model. The usage of augmentation using retrieved domain-specific documents has been shown to improve the quality of the generated answers to health-related questions [Xiong et al., 2024]. One example of RAG being used in the medical domain is Clinfo.ai⁴ by Lozano et al. [2023]. The system retrieves and summarizes medical papers to answer questions from professionals or students.

Generative approaches are specifically useful for answering non-factual questions, which are questions that do not have a single correct answer and usually require a longer explanation [Welivita and Pu, 2023]. Instead of providing a single answer in a short snippet, RAG-based approaches allow for the generation of longer, more detailed answers that mix retrieved information with knowledge embedded into the weights of the model. However, generative systems come with specific challenges regarding the evaluation of the generated answers. This is due to the fact that traditional evaluation metrics rely largely on

⁴<https://www.clinfo.ai>

a lexical overlap between the generated answer and a given reference answer [Xu et al., 2023a]. Due to the nature of generative models, the generated answer can be correct and informative, even if it does not share many words with the reference answer.

1.3 Outline

This thesis aims to tackle the task of creating trustworthy long-form answers for non-factual questions. We focus on answering questions from consumers in a language that is understandable to medical laypersons. We introduce a retrieval-augmented generation pipeline for this task.

We begin by offering a literature review of work done on medical information retrieval and question answering systems. This gives an overview over the state of the art and highlights the challenges that yet need to be addressed, helping to motivate our work. After this, we first review a selection of existing datasets for non-factual health-related question answering before augmenting the MashQA dataset for our specific use case. Then, we address the task of retrieving relevant documents for a given question. We describe our grid search that we use to optimize the retrieval step and perform an evaluation using our dataset. We provide ablation studies for both the query preprocessing and the retrieval pipeline. Finally, we combine the document retrieval with a generative step for creating long form answers. We evaluate a multitude of prompt configurations and two base models for this task using the dataset we created before. The evaluation is done both automatically by comparing the generated answers to reference answers but also using a manual expert evaluation. We provide a detailed analysis of the results and discuss the implications of our findings. We close by summarizing our findings and discussing potential future work.

Our contributions with this thesis are manifold: We augment the MashQA dataset for non-factual medical question answering with TREC-style narratives to enable a better evaluation of retrieval-augmented systems. Furthermore, we produce manual relevance and correctness judgements for retrieved documents. We introduce and evaluate two methods for turning questions posed in natural language into keyword queries. Furthermore we introduce and evaluate a pipeline for the retrieval of relevant passages to answer health-related questions. Lastly, we engineer prompts for the generative step of the RAG pipeline and evaluate the quality of the generated answers. The evaluation process additionally yields some insights on manual and automatic evaluation methods for the generative question answering task.

Chapter 2

Related Work

As we have shown in the introduction, both medical professionals and consumers are using search engines to find medical information. This chapter provides an overview of research in the fields of medical information retrieval and question answering. We use this to highlight where our research fits in and aims to address the challenges identified in prior works. We start by discussing research regarding medical information retrieval, before moving on to question answering. We then describe how both fields of research are combined. Finally, we summarize our findings and show what our research aims to contribute.

2.1 Medical Information Retrieval

Search in the medical domain is multi faceted, differing with respect to the user’s background and the context of the query: this includes the general public researching prior to a doctor’s visit, researchers looking for scientific papers, and doctors looking for treatment guidelines with respect to a specific patient. This highlights the problem of language mismatches due to differing expertise between authors of the retrieved documents and searchers [Upadhyay et al., 2022]. The lexical mismatch is especially damning when using keyword-based retrieval systems. It is countered by an overall shift towards semantic retrieval systems [Tamine and Goeuriot, 2022]. Especially the developments regarding the transformer architecture [Vaswani et al., 2017] and more specifically the BERT foundation model [Devlin et al., 2019] has impacted the field of medical information retrieval, with models such as BioBert [Lee et al., 2020], that are fine-tuned on medical corpora. Overview for medical IR shared tasks show an overwhelming trend towards combination of lexical retrieval combined with transformer based rerankers [Clarke et al., 2021, Suominen et al., 2021].

MEDLINE*plus*¹ is an online search engine for consumers developed by the US National Library of Medicine (NLM). It features access to search in the MEDLINE² database, a collection of biomedical abstracts and citations, as well as retrieval of dedicated health resources tailored towards laypersons [Miller et al., 2000]. The Trip Database³ is a commercial search engine aimed at medical professionals that retrieves biomedical scientific literature from Pubmed and other sources. Click logs of Trip were then used to produce the TripClick dataset, that serves as a benchmark in the medical IR settings [Rekabsaz et al., 2021]. Hofstätter et al. [2022] produce a baseline for TripClick that combines a BM25 retrieval with a BERT-based reranker.

Medical IR has been the topic of multiple dedicated tracks across different information retrieval (IR) conferences such as SIGIR, TREC or CLEF [Tamine and Goeuriot, 2022]. This produced a further selection of benchmarks for medical IR, ranging from consumer health search at CLEF eHealth [Suominen et al., 2021], bio-medical question answering at the BioASQ task [Nentidis et al., 2023], or retrieval in clinical settings focussing more on expert users at TREC BioMedical track [Roberts et al., 2022]. The TREC Health Misinformation Track [Clarke et al., 2022] highlights the dangers of misinformation in health information retrieval, evaluating retrieval systems not only on topical relevance but also on their ability to avoid retrieving misinformation. Bondarenko et al. [2022] tackle the task of retrieving trustworthy medical information from the web by first retrieving documents using a combination of BM25 with the neural Mono-DuoT5-Reranker [Pradeep et al., 2021]. The produced scores are then augmented by boosting the scores of documents containing statements that are also made in retrieved scientific sources.

2.2 Question Answering

The task of question answering (QA) consists of automatically giving answers to questions posed in natural language.⁴ The answers are typically created using a knowledge base that can consist of structured data (e.g., relational databases or knowledge graphs) or unstructured text data [Zhu et al., 2021]. For this thesis we focus on textual QA using unstructured text data.

QA systems are usually divided into two separate components: a retriever that retrieves relevant documents from a corpus and a reader that extracts the answer from the retrieved documents [Zhu et al., 2021]. The retriever

¹<https://medlineplus.gov>

²https://www.nlm.nih.gov/medline/medline_overview.html

³<https://www.tripdatabase.com>

⁴The task of multimodal QA which typically consists of answering a question on a given image also exists.

component works similar to traditional IR systems, with both lexical [Chen et al., 2017] and dense [Karpukhin et al., 2020] systems being used to retrieve evidence for the reader. Note that the retrieval usually yields passages instead of full documents, that are then passed to the reader to extract the answer from the retrieved evidence [Zhu et al., 2021].

Nassiri and Akhloufi [2023] highlight the need for an additional step of query preprocessing in the retrieval process. A common aspect picked up in preprocessing for QA systems is the transformation of natural language questions into keyword queries. Given the majority of queries used in search engines are not posed in natural language, most retrieval systems are designed to work with keyword queries [White et al., 2015]. While there has been done some work on turning questions into queries (either using an extractive [Musa et al., 2018, Ni et al., 2019] or generative approach [Iovine et al., 2022]) there is no clear evidence for either keyword or question form being better for the retrieval process with published results going both ways [Iovine et al., 2022, Pugachev et al., 2023, White et al., 2015].

The appropriate working of the reader step, that turns retrieved documents into a concise answer, depends on the type of the question being asked as well as the type of answer expected [Rogers et al., 2023]. Boolean questions that expect an answer of either "yes" or "no" are typically tackled by training a classifier to derive an answer from the retrieved source [Clark et al., 2019]. Similar approaches exist for multiple choice questions, where the reader is trained to select the correct answer from a list of possible answers based on a given document [Lai et al., 2017]. Questions without predefined answer options are usually further divided into factual and non-factual questions.⁵ While factual questions typically have a single correct answer, non-factual questions can have multiple correct answers or no correct answer at all. Answers to the former can consist of a short sentence, but answers to non-factual questions are usually more complex and longer given the need for further explanation or reasoning [Cortes et al., 2022]. Additionally, answers to non-factual questions are usually more dependent on the context of the question, further underlining the importance of additional context in the answer [Breja and Jain, 2022]. While question words such as 'who', 'where' or 'what' can be indicative of factual questions, the distinction between both question types is fuzzy. Especially definitory questions starting with 'what' can well be non-factual questions, given multiple ways to interpret the question and thus more than one correct answer [Cohen et al., 2018].

Factual QA can be tackled using extractive methods, where a single

⁵In the scientific literature these questions are usually called 'factoid' and 'non-factoid' questions. However, to prevent confusion with the term 'factoid', which may denote trivial or unsubstantiated information, we opt to use the term 'factual' instead.

consecutive span of text is extracted from a retrieved document by classifying tokens as either the beginning or end of the answer span [Chen et al., 2017]. While the same approach can be used for non-factual questions, an addition to reflect the need for longer and more complex answers is the introduction of multiple-span extraction [Zhu et al., 2020].

2.2.1 Generative Question Answering

Roberts et al. [2020] use generative large language models in the field of question answering. They fine-tune a T5-based sequence-to-sequence model [Raffel et al., 2020] for the QA-task and show that competitive results are possible even without supplying any additional information to the model and relying solely on knowledge embedded into the model's weights. Izacard and Grave [2021] add a retrieval step to the generative QA process. They encode multiple retrieved passages separately, combining them and the question only in a specially trained decoder when generating the final answer. This approach, dubbed "Fusion-in-Decoder", allows for large amounts of context documents to be passed to the model. Lewis et al. [2020] introduce the concept of retrieval-augmented generation (RAG): They utilize a fine-tuned sequence-to-sequence model for the generation, feeding context and question into the model by concatenating them. While the performance is superior to purely parametric QA-systems, RAG has the additional benefit of being a highly flexible approach as both the retrieval source as well as the model can be exchanged to fit the specific needs of the task at hand.

While these approaches need additional task-specific training data, retrieval augmentation has been shown to work even with retrieved contexts only being introduced at inference time, using an off-the-shelf pretrained model [Lazaridou et al., 2022, Shi et al., 2023, Xu et al., 2023b]. The usage of frozen models increases the flexibility of RAG even further, additionally allowing researchers to increase the focus on other details of the generation process [Gao et al., 2023]. This includes the usage of task-specific few-shot prompting [Brown et al., 2020] or the introduction of additional retrieval steps, where the model is periodically supplied with new context documents [Jiang et al., 2023b]. A preprocessing step for the retrieved passages has been shown to improve the QA performance, as it can reduce the amount of noise in the passed context [Yang et al., 2023].

2.2.2 Evaluation of Question Answering Systems

Evaluation of QA systems with limited answer options is fairly straightforward as the systems' answers can easily be compared to the correct answer. The QA performance is then expressed as accuracy or, in the case of open but short

answers, as percentage of exact matches [Rajpurkar et al., 2016]. QA settings that expect longer answers are harder to evaluate automatically given an answer can well be correct even if it does not match the exact wording of the correct answer [Xu et al., 2023a]. One possible approach is to use metrics that measure term overlap between the generated answer and a given ground truth. Such metrics include BLEU [Papineni et al., 2002], ROUGE [Lin and Och, 2004], and METEOR [Banerjee and Lavie, 2005]. These metrics are limited in their ability to capture the semantic similarity between the generated and the correct answer, given they all depend on a lexical overlap between candidate and reference answers [Xu et al., 2023a]. These limitations become even more severe when evaluating generative systems, given the complete lack of lexical guiderails when compared to extractive systems [Chen et al., 2019]. The use of text embeddings on term [Zhang et al., 2020] or answer level [Aynedinov and Akbik, 2024] allows for a more semantics-aware evaluation of QA systems. While still relying on comparison of generated and reference answers, the use of embeddings has shown to correlate highly with human preference regarding the quality of the answer [Aynedinov and Akbik, 2024].

The inherent ambiguity of non-factual questions raises the question of how to conceptualize the quality of answers. Cambazoglu et al. [2021] show that the perception of 'usefulness' of a long-form answer to human annotators can be broken down into the categories of relevance (the answer actually addresses the question), correctness (there is no misinformation in the answer), completeness (the answer does not leave out any important information), and comprehensiveness (the question is understandable to the asker). They find automatic metrics to focus on relevance and correctness, with the other aspect not being captured by the automatic approaches.

The fine-grained automatic evaluation of QA (and more specifically QA using RAG) systems has been addressed using generative LLMs. Es et al. [2023] propose the RAGAs framework that measures a system's answer relevance and correctness. While the relevance metric uses a simple embeddings-based similarity metric to make sure the given answer and the expected answer are semantically similar (Aynedinov and Akbik [2024] use the same approach for the evaluation of generative models), correctness is measured entirely based on the output of an additional generative language model, that is asked to compare the generated answer to the correct answer. **OpenAI Evals**⁶ is a similar approach and bases its metrics entirely on the output of a generative model that is then parsed to extract the desired metrics. While the use of generative models for evaluation offers a great flexibility given the prompt can be adjusted to fit the specific needs of the task, it also hinges greatly on the

⁶<https://github.com/openai/evals>

model’s capability to provide fair results. Wang et al. [2023] show that a purely LLM-based evaluation can easily be cheated. An additional perspective on the QA systems is the ability to refuse to answer a question if the retrieval process yields insufficient information. Current black-box LLMs struggle to provide a refusal even if it would have been appropriate to do so [Chen et al., 2024].

2.2.3 Question Answering in the Medical Domain

As described before, medical documents are commonly quite complex and difficult to understand for consumers. This motivates the use of automatic QA systems that can help automatically find answers to medical questions, lowering the barrier of entry for consumers to access medical information [Jin et al., 2023, Welivita and Pu, 2023]. Our observation regarding medical IR holds true for the QA task as well: Given the high stakes of the medical domain, the trustworthiness of the information retrieved is of utmost importance [Xiong et al., 2024]. Additionally, the large difference between professional and layperson questions both with regards to the used terminology as well as the character of the expected answer may pose a challenge for the QA systems [Liu et al., 2011, Welivita and Pu, 2023].

Question answering in the medical domain is typically split into three categories: scientific, clinical, and consumer health QA. Clinical QA systems can help make decisions in a clinical setting, such as diagnosing a patient or suggesting a treatment plan based on a patient’s symptoms or medical history. Scientific QA systems can help researchers find answers to scientific questions, such as finding the latest research on a specific topic or understanding the results of a study. Those two categories are typically aimed at medical professionals and researchers. A common retrieval source for these tasks are abstracts of biomedical papers (PubMed) [Jin et al., 2023]. Consumer health QA systems, on the other hand, are aimed at consumers who are looking for information about their health or medical conditions. These systems typically focus greatly on making answers understandable to laypersons [Welivita and Pu, 2023]. Systems for medical and health-related QA typically follow similar approaches to those for open QA, differing mostly in the usage of custom retrieval corpora, question preprocessors, and language models fine-tuned on medical data [Mutabazi et al., 2021].

Similar to the search task, medical question answering systems make heavy use of BERT-based models [Devlin et al., 2019, Singh and Susan, 2023]. Alzubi et al. [2021] use BM25 to first retrieve relevant abstracts for factual questions about Covid-19 from PubMed. The reader stage then splits these documents into paragraphs and extracts answers using a finetuned DistilBERT-Model. This system was developed in light of the Covid pandemic to keep doctors and

researchers up to date with the latest research findings on Covid-19.

Answering Health-Related Consumer Questions

In 2022, the TREC Health Misinformation Track⁷ featured an answer prediction task for consumer health questions. Participants are first asked to retrieve relevant documents from the web, before using them to infer the answer to a given yes-no question [Clarke et al., 2022]. Pugachev et al. [2023] forego any dedicated medical components and build a functioning QA system for these questions using off-the-shelf components. This includes retrieval from PubMed before using BERT and RoBERTa-based classifiers tuned on an open domain boolean QA dataset for the answer generation.

Demner-Fushman et al. [2020] build the Consumer Health Information and Question Answering system (CHiQA)⁸ to address a wider range of consumer questions. The system first tries to correct any grammatical or spelling mistakes in the question before using a combination of feature-based SVMs and rule-based approaches to identify the type and focus of the question. This information is then used to retrieve relevant documents from a corpus of consumer-facing websites supplied by the US National Institute of Health (NIH). Terms related to the type and focus of the question get a weight boost in the BM25 retrieval. Additionally, the retrieval makes use of question-entailment based re-ranking, where given questions are compared to questions for which relevant documents are already known. If similar questions have been seen before, their respective relevant documents get a boost in the ranking. The retrieved documents are then shown to the user with no additional extraction step. Zhu et al. [2020] approach the problem of extracting answers for non-factual questions from given context documents by introducing a multi-span extraction model. They use an inter-sentence-attention enhanced sentence-based classifier to extract multiple spans from context documents. This allows them to create long-form answers, consisting of sentences located at different positions in the context.

Generative Question Answering in the Medical Domain

As described in the introduction, there have been promising results with the usage of generative LLMs for answering medical questions, with Google’s Med-PaLM [Singhal et al., 2023a,b] being the largest domain-specific model. Guo et al. [2023] adapt a smaller language model for purely generative answers, showing that even smaller models can provide competitive results in the medical

⁷The track has been running since 2019 (originally known as the "TREC 2019 Decision Track"), but did not include a question answering task.

⁸<https://chiqa.nlm.nih.gov>

domain when fine-tuned appropriately. Wu et al. [2023] also instruction tune a small model for medical QA and highlight the ability to provide competitive performance across multiple question types such as non-factual but also multiple choice questions. Given the high stakes of the medical domain, Xiong et al. [2024] highlight the importance to enrich the generation process with up-to-date and reliable information to limit the amount of hallucinations. A retrieval process additionally helps the patients by providing relevant documents to offer users additional context if they wish to delve deeper into the topic.

Lozano et al. [2023] collect medical questions from titles of scientific papers and manually extract answers from the text. This data is then used to evaluate their newly introduced RAG-based QA system⁹ aimed at medical professionals and scholars. The system relies on OpenAI’s inference API in every step: First, questions are turned into keyword queries by prompting GPT3.5 to do so. These queries are then used to retrieve relevant papers from PubMed via the available search API. GPT3.5 is then used again to filter out any false positives from the retrieved documents. The final answer is generated by letting GPT3.5 first summarize each abstract before using the same model to create a synthesis of all relevant abstracts. Zakka et al. [2024] build a RAG system for clinical questions that aims to help medical practitioners. They use OpenAI’s inference API to generate answers based on documents retrieved from a vector database consisting of a mix of textbooks, scholarly literature, and trusted documents from the web. They completely forego the struggles of automatic evaluation by evaluating the systems purely using manual judgments from medical professionals. The aspects of factuality, completeness, as well as overall quality of the answers are evaluated using expert rankings.

Li et al. [2023] apply RAG in a more consumer-focused setting. They create a medical chat bot that retrieves additional information from Wikipedia and enriches the user’s prompt with additional context similar to a traditional RAG approach. However instead of focusing on a QA task, the system is designed to provide a more conversational experience to the user, closely simulating a doctor’s visit. This is underlined by the training being based on a dataset of interactions between patients and doctors. They underline the importance of the retrieval aspect as it allows to access information that was not present in the training data, thus increasing the system’s flexibility. The generated answers are subjected to a qualitative evaluation by the authors, who compare the system’s answers to other generative systems and find their system to provide more coherent answers. Additionally, an automatic evaluation is performed on the iCliniq-10k¹⁰ dataset using BertScore as a semantic similarity metric. While the evaluation follows a non-factual QA setting, the iCliniq questions

⁹<https://www.clinfo.ai>

¹⁰The dataset is available at <https://github.com/KentOn-Li/ChatDoctor>

are conversational in nature, commonly consisting of multiple sentences and offering an in-depth look into the patient’s situation. The dataset also features multi-turn QA dialogs, where follow-up questions are asked.

2.3 Summary

Medical IR is a well-established field with a multitude of datasets and benchmarks. The specific domain comes with its own challenges, such as the high importance of trustworthiness of retrieved information as well as a strong lexical mismatch between professional and layperson queries. The field of question answering has picked up generative models in the past years, with the introduction of retrieval-augmented generation systems combining retrieval and generation steps to provide long-form answers to non-factual questions. This trend is also present in the medical domain, with the introduction of medical QA systems that provide answers to clinical or scientific questions.

Regarding the task of answering consumer’s health questions, we find that there is a need for additional research into consumer health QA [Welivita and Pu, 2023]: While our literature review shows that e.g., both the selection of the retrieval source (e.g., consumer-facing websites such as WebMD as used by Zhu et al. [2020]) and preprocessing of consumers’ questions [Demner-Fushman et al., 2020] have been the focus of prior work, we find very little work on the usage of retrieval augmentation in the field of consumer health QA. While Li et al. [2023] use retrieval augmentation to enrich LLMs with additional context for consumer questions, their large-scale evaluation uses a conversational dataset that differs from a one-turn QA setting. This motivates our work on filling this gap by first developing a RAG system, before then evaluating the system’s ability to answer non-factual questions.

Chapter 3

Dataset

Our literature review has shown that while the field of consumer health question answering has been addressed before, there are still aspects that need to be worked on. Our work proposes a RAG model for the task of consumer health question answering. Given the challenges for the evaluation of generative QA models we described before, we need a dataset that allows us to evaluate the performance of our model in a meaningful way. We need a dataset that contains natural language, non-factual questions that require long form answers. The provided reference answers should be trustworthy and understandable for laypersons. In this chapter we first conduct a review of existing datasets and select the most suitable ones for our task. We then augment and filter this dataset to better fit our needs. Finally we provide some descriptive analysis of the dataset we created.

3.1 Existing Datasets

We collect a set of candidate datasets of consumer health question and answer pairs using summary studies [Breja and Jain, 2022, Jin et al., 2023, Welivita and Pu, 2023] and our own research. Only datasets consisting of non-factual questions with long-form reference answers are included. Additionally, we exclude any multi-turn conversational datasets, as we focus on the single-turn QA setting. We provide a short overview for each inspected dataset and finally explain our decision on what dataset we use for our further research.

The **TREC LiveQA track** addresses the task of consumer health question answering. All questions are inquiries sent via email to the U.S. National Library of Medicine (NLM). Each question is represented by a subject line and the body text, usually consisting of multiple sentences providing background information to the consumer and a question. Some emails include multiple sub questions. The provided answers are (partially) manually extracted from

documents retrieved from PubMed [Abacha et al., 2017]. While these questions generally are non-factual and open-ended, the given answers stem from scientific literature and are not necessarily understandable for laypersons. The long form questions come with additional challenges given the possible noise in the text of the email body.

Abacha et al. [2019a] introduce the **MedicationQA** dataset. It consists of anonymized questions sent by consumers to MedLine Plus.¹ The reference answers are manually collected by medical experts from trustworthy websites. While many questions are non-factual and open-ended, the dataset has a clear focus on medication-related questions, ruling it out for our use case. **MedQuAD** is a dataset of over 47,000 health-related question-answer pairs collected from 12 NIH websites.² While the questions cover a wide variety of health-related topics and generally can be characterized as non-factual, the question and answer pairs are automatically extracted using a pattern matching approach [Abacha and Demner-Fushman, 2019]. **MEDIQA-QA** focusses on ranking answers for consumer questions. The provided answers were generated by the ChiQA³ system. While the candidate answers are ranked by medical experts, there is no clear reference answer provided for each question [Abacha et al., 2019b].

Zhu et al. [2020] introduce **MASH-QA** (short for "Multiple Answer Spans Healthcare Question Answering"). All questions are sourced from WebMD⁴ articles about a wide range of health topics. These pages are written by medical professionals and feature a question and answer section. Each question is answered by manually produced extracts from the corresponding article. This allows for easy evaluation of extractive QA systems. Zhu et al. [2020] highlight the non-factual nature of the questions and the need for long-form answers. The latter is underlined by the fact that the extracted answers are often multiple sentences long and stem from multiple different paragraphs in the source text (thus the name of the dataset). Given that MASH-QA closely matches our requirements, we decide to use it as the basis for our dataset. While the dataset is originally created for extractive QA approaches and only provides benchmark scores for extractive QA models, we believe that it can be adapted for usage in the field of retrieval-augmented generation. Furthermore, the provided context passages can be used for ablation studies were the retrieval step is omitted and the model has to generate an answer based on the given context.

¹<https://medlineplus.gov>

²<https://github.com/abachaa/MedQuAD>

³<https://chiqa.nlm.nih.gov>

⁴<https://www.webmd.com/>

Table 3.1: Datasets used to source narratives for our test dataset.

Track	Years	n Topics	n used
CLEF Consumer Health Search	2021	55	5
" eHealth Task 3	2013 — 2014	100	0
Neuclir	2022	60	1
Touché Causal Questions	2023	50	0
" Comparative Questions	2020 — 2022	150	1
" Controversial Questions	2023	50	1
TREC Common Core Track	2017, 2018	250	2
" Covid Track	2020	50	0
" HARD Track	2003, 2005	100	0
" Health Misinformation Track	2019 — 2022	200	7
" Microblog Track	2015	255	1
" Novelty Track	2002 — 2004	150	0
" Podcast Track	2020, 2021	100	1
" Robust Track	2004, 2005	300	1
" Terrabyte Track	2004 — 2006	150	1
" Web Track	1999 — 2002, 2009 — 2014	500	6
Total		2,520	27

3.2 Creating the MashTREC Dataset

We want to be able to evaluate each step of the QA process, which includes judging the relevance of retrieved documents passed as context to the answer generation. To simplify the evaluation and improve the quality of relevance judgements we decide to augment the MashQA dataset to follow the standard TREC format [Soboroff, 2021]. This involves adding narratives to each question that further describe what information relevant documents must include. To ensure a high quality of these narratives we opt to reuse existing narratives from topics that have been used before to evaluate IR pipelines.

For this we first collect all available topics from the TREC topic collection⁵ as well as topics available in the python package `irdatasets` [MacAvaney et al., 2021].⁶ The available topics are not limited to health-related tasks, but cover a wide variety of topics and retrieval contexts. We then filter out those topics that do not have any additional narratives or descriptions that

⁵<https://TREC.nist.gov/data.html>

⁶The list of available datasets can be found at <https://ir-datasets.com>

contain additional information about relevance. Note that some topics supply additional information in a field called "description" instead of "narrative". We include these as well.

To help us find matching narratives for our test questions we embed all questions from MashQA as well as all narratives using `all-mpnet-base-v2` model.⁷ We then compute the pairwise cosine similarity between questions and narratives⁸ and keep only those pairs with a similarity higher than 0.7. We choose this threshold arbitrarily, given this filter only serves to reduce noise for our manual matching process of the 175 question-narrative pairs that remain. Manual inspection of the results show that the resulting pairs are still fairly noisy: First we find that the narratives are partially too general or too specific to be useful for judging retrieval results for the given questions. Second, some questions match multiple narratives, which is not ideal as we want to have a one-to-one mapping between questions and narratives. We manually remove those pairs that are not suitable. Additionally, we remove pairs with question that could easily be answered with "yes" or "no" as we want to focus on open-ended questions. This process yields 27 question-narrative pairs (cf. Table 3.1 for the source of all matched narratives) that make up the MashTREC test dataset that we later use to evaluate both our retrieval as well as our generation process.

3.2.1 Creating a Training Dataset

As shown in the previous section, finding matching narratives for our test questions is a difficult task. We only find 27 matching narratives for the 3,587 questions in the MashQA test dataset. Given our collection of narratives only consists of 2,520 topics, an additional matching process for a training dataset would likely yield very few matches that would possibly be very similar to those in the test dataset. Since a high quality of relevance judgments is clearly more important for the test dataset, we decide not to use narratives for our training dataset. Instead of using the whole MashQA dataset, we opt to only use a subset of 50 questions. This has several advantages: First, the dataset is more manageable in size. Second, we can ensure that the dataset is similar to the test dataset by using stratified sampling based on our test questions. Finally, we can enforce a strict separation between training and testing by not using questions for training that are too similar to those in the test dataset.

To create the training dataset we first randomly sample 1,000 questions from the MashQA training dataset, stratifying by the first word in the question.

⁷<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁸We use the CLS-token to produce sentence-level embeddings for both questions and narratives.

Table 3.2: Prefixes of the questions in our dataset.

Prefix	Training	Test
What	20 (40%)	14 (52%)
How	18 (36%)	11 (41%)
Other Prefixes	12 (24%)	2 (7%)

We use the classes "What" with a weight of 0.5, "How" with a weight of 0.4 and combine all other prefixes with a weight of 0.1. This ensures that the question types in our training dataset are similar to those in the test dataset.

We then embed both the test and the sampled training questions using `all-mpnet-base-v2`. We compute the pairwise cosine similarity between each test question and each sampled training question⁹. We rank each training question according to its highest similarity to any test question and keep only the bottom 500 questions. This filters out the questions that have close matches in the test dataset. We sample the remaining 500 questions using the same stratified sampling as before and end up with a training dataset of 50 questions.

3.2.2 Characteristics of MashTREC

We call our adapted version of the MashQA data set "MashTREC" to refer to the two main sources used. As shown in Table 3.2, both the training and test datasets are dominated by definitory questions starting with "What". As shown in the literature review this is not automatically indicative of factual questions. The second large group are questions starting with "How". The distribution of prefixes is similar in both datasets, which is expected due to the stratified sampling. The small difference in the distribution of prefixes is likely due to the filtering process.

We provide two examples of questions from our test dataset that start with "How" and "What" in Table 3.4. The question presented, which begins with "What", is clearly definitional in nature: The provided reference answer is several sentences long and provides examples to illustrate the explained concept. Thus, the question is non-factual and fulfills our requirements. The presented narratives are a result of the automatic matching process. Although they were both originally created for slightly different questions, they fit our questions very well. This is representative for our full test dataset.

The non-factual property of the MashTREC questions is underlined by the fact that the reference answers are all several sentences long (cf. Table 3.3). While the sentences in the training variety of our dataset are marginally shorter,

⁹We use the CLS-tokens for sentence-level embeddings.

Table 3.3: Length of the reference answers in our dataset. Terms are defined as words separated by whitespace.

	Characters	Terms	Sentences
Test	413.3	70.6	4.8
Training	334.3	57.2	3.1

a manual inspection of the questions shows that they are all open-ended and require a long-form answer.

Each MashQA question is annotated with the URL of the source article. We extract a category (e.g., "Migraines Headaches" or "Skin Problems and Treatments") from URL's path and use it to classify each question. We find that the questions in our filtered dataset cover a wide variety of topics (17 different question categories in test and 26 in training). The most common categories in our test dataset are "Migraines Headaches" and "Beauty" (three questions each). The most common category for our training dataset is "Oral Health" with six questions. Although the exact medical fields covered by the questions are very broad, all questions can be answered from the perspective of a general practitioner. This assertion is supported by our expert study for the generative step late in the process. Doctors from different specialties indicated only minor problems in terms of lack of expertise to answer the questions. The distribution of categories in our dataset is available in Appendix A.

Table 3.4: Sample questions from the MashTREC test dataset. Narratives are result of a semi-supervised matching process to topics following the TREC format. The questions themselves and their reference answers are taken from the MashQA dataset.

Question	How does a massage work to treat cellulite?
Narrative	Cellulite is the dimpling of skin over an area of the body. Massage is a process of rubbing and applying pressure to the body with the intent of either reducing tension in muscles or producing other benefits. To "get rid of cellulite" would be to make the skin smooth without dimples over a long-term period.
Narrative Source	TREC 2022 Health Misinformation Track
Reference Answer	It can liquefy fat, cut connective tissue to loosen puckering, boost collagen growth and skin tightening, increase blood flow, and lessen fluid retention. Endermologie (or lipomassage or endermologie) is a specific type of mechanical massage. A machine with low-pressure suction kneads your skin between two spinning rollers. The theory is that the deep massage will break up the connective tissue that causes dimples. Most studies show that massage techniques, including endermologie, make your skin look better for a short time but offer no long-term benefit. Some experts worry that the suction can cause your skin to slacken prematurely, making it look worse. Focus on your legs, hips, and backside.
Category	Beauty
Question	What is the meaning of metabolism?
Narrative	A relevant document will contain specific information on the catabolic and anabolic reactions of the metabolic process. Relevant information includes, but is not limited to, the reactions occurring in metabolism, biochemical processes (Glycolysis or Krebs cycle for production of energy), and disorders associated with the metabolic rate.
Narrative Source	TREC 2004 Robust Track
Reference Answer	Metabolism refers to all the chemical reactions taking place in the body to convert or use energy. A few major examples of metabolism include: Breaking down the carbohydrates, proteins, and fats in food to release energy. Transforming excess nitrogen into waste products excreted in urine. Breaking down or converting chemicals into other substances and transporting them inside cells.
Category	A To Z Guides

Chapter 4

Retrieval

In this chapter we focus on the retrieval process of our system. Our goal is to create a retrieval system that is able to find relevant and trustworthy information that can then be used to answer health-related questions. The system should produce short text snippets that we can then feed into our answer generation system.

We first address the challenge of health misinformation and develop our approach to avoid this problem. We then discuss several approaches to handle the need for preprocessing queries (or questions) for the retrieval process, which was discussed in our literature review earlier. Finally, we follow the examples discussed earlier and combine lexical and dense retrieval systems to create a retrieval pipeline. This pipeline is then evaluated on both our MashTREC dataset and the TREC 2022 Health Misinformation topics.

4.1 Retrieving from Reputable Sources

The trustworthiness of the retrieved information is a critical aspect of the retrieval process. Especially in the medical domain, untrustworthy information can have serious consequences. While we have previously discussed approaches to assessing the trustworthiness of documents, we choose to take a different approach. Instead of letting our retrieval system decide the trustworthiness of retrieved documents, we solve the challenge of trust in our retrieval results by retrieving only from sources that we consider a priori to be trustworthy.

One way to ensure reliable information in the retrieved documents is to use a dataset of scholarly articles, such as PubMed [Jin et al., 2023]. This ensures that the information is peer-reviewed and up to scientific standards. However, this approach has the disadvantage that the language used in the retrieved documents can be too complex for laypeople to understand. We aim to provide answers to questions that are understandable to the general public.

Therefore, we choose to retrieve from a corpus of webpages that are known to provide trustworthy information while still being accessible to a wide audience.

We do this by first creating a list of reputable sources and retrieving only from these sources. This follows the example of the CLEF eHealth Evaluation Lab from 2013 to 2015 [Palotti et al., 2015]. The task on user-centered health information retrieval used a corpus provided by the KHRESMOI project.¹ The EU-funded project aimed to build a multilingual and multimodal search engine for the medical domain. To ensure trustworthiness of the retrieved documents, KHRESMOI only crawled webpages that were HONcode certified by the Health On the Net Foundation² (HON). This certification was awarded to health-related websites that met a set of standards, ranging from good privacy practices to only providing evidence-based information [Boyer et al., 1998]. Additionally, only websites publishing content by qualified editors were considered. HON also underlined the need for information to be understandable and accessible to the public. The foundation is defunct as of December 2022 and the HONcode certification is no longer available to new websites.

Although we could theoretically use the corpus used by the CLEF lab to limit our retrieval to trustworthy sources, we choose to create our own corpus. We do this for two reasons: First, the CLEF Lab corpus is not publicly available. Second, the corpus is outdated and may not reflect the current state of medical standards. We base our corpus on the Clueweb22-B [Overwijk et al., 2022] crawl consisting of 200 million webpages. This collection is then filtered to only include webpages designated to be trustworthy. We use the following sources to collect trustworthy domains:

- While there is no publicly accessible list of all websites certified by the HON foundation, a **HonCode browser extension** exists.³ This open source application notifies users when they visit a HONcode certified website and marks links to certified websites. We find a list of 6,581 certified domains at <https://www.honcode.ch/HONcode/Plugin/liste.txt>.⁴ While the list is no longer updated due to the defunct status of the HON foundation, the last update still encompasses the crawling period of the Clueweb22-B collection.
- The **KHRESMOI Reference Corpus** [Khresmoi, 2016] is a collection of 1,083 documents from the medical domain. It contains the full

¹The project’s website is not available anymore, but there is a snapshot from 2022 available: <https://web.archive.org/web/20220205054245/http://www.khresmoi.eu/>

²The last available snapshot of HON’s website is available at <https://web.archive.org/web/20230709100236/http://www.hon.ch/en/>

³<https://github.com/healthonnet/hon-honcode-extension/>

⁴Thanks to Jan Heinrich Reimer for pointing this out!

text of documents that are annotated with mentioned medical entities. The corpus was created to serve as training data for automated entity annotation methods. The used documents come from two distinct sources: first, it consists of Wikipedia pages and secondly of webpages crawled by KHRESMOI project. For our use-case, we extracted the documents' sources, yielding a total of 77 unique domains.

- The **MedQuAD** collects health-related question and answer pairs from the websites associated with the US National Institutes of Health (NIH).⁵ The list of domains is available on the project's GitHub repository.⁶ Overall, 9 governmental domains are included in the dataset.
- **OpenMD**⁷ is a freely accessible online search engine for health-related information. Additionally, it offers a collection of trusted homepages offering medical information⁸ directed at consumers. All websites are reviewed by medical professionals. The listed websites are grouped by the topic that is mainly discussed on each website. We scrape all available directory pages and extract the linked sites' domains, resulting in a total of 812 entries.

Some sources we use include URLs instead of domains. In this case we extract the domain from the URL by removing any path or query information while keeping all subdomains. Deduplication of the domain list results in a total of 7,309 unique domains. Filtering the Clueweb22-B corpus to only include documents from these domains results in a corpus of 21,889,770 documents. We deem this to be a sufficient amount of documents to retrieve from.

4.2 Preprocessing

As discussed in the literature review, it is unclear whether retrieval using natural language questions or keyword queries is preferable in terms of retrieval effectiveness [Iovine et al., 2022, Pugachev et al., 2023, White et al., 2015]. Given the importance of query preprocessing in the medical domain [Welivita and Pu, 2023], we still develop two approaches to preprocess natural language questions into keyword queries.

While one of our proposed systems works by selecting the most important terms from a question, the other system is able to completely rewrite the

⁵<https://www.nih.gov>

⁶<https://github.com/abachaa/MedQuAD>

⁷<https://openmd.com>

⁸<https://openmd.com/directory>

question into a keyword query format, possibly introducing completely new terms. We describe the creation of both systems in the following sections before performing an ablation study, evaluating the impact of the preprocessing techniques on the retrieval effectiveness of a QA system.

4.2.1 Essential Term Extraction

Khashabi et al. [2017] claim that QA systems can be distracted by long-form natural language questions including redundant terms. They introduce the concept of essential question terms. A term’s essentiality is hereby defined as how necessary it is to see it when answering a given question. This concept is further specified by three additional rules:

1. Exchanging an essential term changes the meaning of the questions.
2. Dropping a non-essential term does not change the correct answer of a question.
3. It is not important whether or not a question is still phrased grammatically correct after dropping a term.

Khashabi et al. [2017] introduce a dataset of 2, 223 elementary school science exam questions. For each question a total of five crowd sourced workers on Amazon Mechanical Turk were asked to judge each term’s binary essentiality. The majority vote is then used to produce a term-level essentiality label. Classifiers trained using the so produced dataset have been used to retrieve evidence in multiple choice QA systems before by combining a question’s essential terms with each answer option to retrieve evidence [Musa et al., 2018, Ni et al., 2019].

We propose an extractive query preprocessor for the open QA setting that we train using the Essential Terms dataset. We train our own transformer based token classifier that is then used to extract a question’s essential terms. The extracted terms are then used as a keyword query to retrieve relevant passages to answer the question. We fine-tune a transformer on the classification task using a hyperparameter search on the essential terms dataset. We follow Khashabi et al. [2017] by using the majority vote over all manual annotations to produce binary relevance labels for each term. The essential terms dataset uses words separated by whitespace as terms, so we preprocess the dataset by first tokenizing it using our base model’s tokenizer and then transferring the original term labels to the newly created tokens.

We perform a random search using the distributions defined in Table 4.1. We perform a total of 75 runs using a random 8:1:1 split for training, validation, and testing following Musa et al. [2018]. The trained models are compared by

Table 4.1: Hyperparameter value ranges for training of an encoder based classifier to identify essential terms in a question. All runs use a uniform distribution for sampling from the value range.

Parameter	Value Range
Base Model	{distilbert/distilbert-base-uncased, sentence-transformers/all-MiniLM-L12-v2, FacebookAI/roberta-base}
Weight Decay	0 — 0.5
Learning Rate	1e-8 — 1e-3
Epochs	1 — 30
Batch Size	1 — 10
Class Weights	{balanced, imbalanced}
Scheduler	{cosine, linear}
Warmup	0 — 5000

Table 4.2: Essential terms classification results. We compute F_1 -score, precision (Pr), recall (Re), and accuracy (Acc) per question and macro average the results. The best value per column is printed in bold.

Approach	Acc	Pr	Re	F_1
ET Classifier [Khashabi et al., 2017]	0.75	0.91	0.71	0.80
ET Net [Ni et al., 2019]	-	0.74	0.90	0.81
NCRF++ [Musa et al., 2018]	0.88	0.73	0.80	0.77
Our Approach	0.92	0.86	0.85	0.84

their F_1 -score on the validation dataset. Multiple runs reach a perfect score of 1 on both the training and the validation dataset, therefore we choose our model based on the lowest validation loss. Our selected model uses the RoBERTa base model and is trained for 13 epochs.⁹

We evaluate our best model using our test dataset and compare it to prior approaches. Table 4.2 shows that our model outperforms all prior work in terms of accuracy and F_1 -score.

The model can then be turned into a query preprocessor: We first run the given natural language question through our classifier. By only keeping tokens deemed "essential" we can then transform the question into a keyword query. Note that this does not enable complete rewriting of queries, as the approach

⁹Batch size is 3, the learning rate is 2.543×10^{-5} using cosine scheduling. The class weights are balanced and the weight decay 0.21. We use 2790 warmup steps.

Table 4.3: Comparison of the output of term classifiers for the question: Emily made a cup of tea and stirred it with a spoon. The spoon became warm. How was the heat from the tea transferred to the spoon?. Note that this question comes from Musa et al. [2018] and is not present in the original dataset. That means that there is no ground truth for it.

Approach	Query
NCRF++ [Musa et al., 2018]	tea heat tea transferred
Our Approach	tea stirred spoon warm heat transferred

works purely extractive and is not able to introduce new terms. Table 4.2 compares our model to one prior approach when used as a query preprocessor. Musa et al. [2018] are the only ones to provide an example output, which is why we compare our approach to theirs. Our approach extracts more terms from the question while avoiding duplication of terms. However, there is no evidence that one approach is superior with regards to retrieval effectiveness.

4.2.2 Sequence-to-Sequence Preprocessing

As discussed before, medical information retrieval has the problem of language mismatches: There are many ways to express the same information need in a query, with different phrasing possibly leading to different retrieved documents. A purely extractive approach like the one described above is not able to apply such a rephrasing. To counter this pitfall we propose a sequence-to-sequence model that is trained to fully rewrite a question into a query. This model can then be used as a preprocessor to retrieve evidence for questions.

Dataset

We do not come across any large scale datasets of query-question pairs in the medical domain.¹⁰ To train a sequence-to-sequence model to rewrite questions into queries, we need such a dataset. We decide to create our own dataset using a semi-supervised approach by combining questions from the MashQA training dataset with queries from the Tripclick dataset.

The Tripclick dataset [Rekabsaz et al., 2021] contains click logs of the Trip database.¹¹ The Trip database is a search engine for medical professionals

¹⁰Note that the TREC Health Misinformation dataset features both natural language questions and keyword queries for each topic. However, only 150 such pairs are available. Additionally, we use the 2022 edition of the track in our ablation study, ruling it out as training questions.

¹¹<https://www.tripdatabase.com>

containing click data for 1.6 million health-related queries. The queries are mainly posed by medical professionals, so the language in the queries is highly specific and likely differs from the language used by consumers. The HEAD variant of the dataset contains queries that represent the most frequent 20% of the search engine's traffic.

We start our approach by first embedding all questions and queries using the all-MiniLM-L12-v2 model.¹² The pairwise similarity between all questions and queries is then computed using cosine similarity.¹³ We select the top five matches per question as match candidates. This gives us a total of 137,835 potential question-query pairs (five pairs for each of the 27,567 questions from the MashQA training variant). We exclude any pairs containing questions that we already use in our MashTREC dataset. Our goal is now to establish both a similarity metric and a threshold score that allows us to differentiate between matches and non-matches. The task differs from a traditional classification problem as we are using the results for the training of a downstream model. Instead of finding a similarity threshold that gives us a good class separation, we want to maximize the number of matches while still ensuring high quality matches. In practice this means that we want to maximize recall for a fixed precision. To find a threshold that fulfils this requirement we manually annotate part of the candidate pairs as either "Match" or "No Match". Using these manual annotations we can then estimate the precision and recall for different thresholds and similarity metrics, allowing us to choose the best option.

Annotation: The annotation is conducted by six annotators using the open source annotation tool Doccano.¹⁴ A screenshot of the user interface is given in Figure 4.1. No annotators are medical professionals. One is a PhD student in computer science, one is a graduate student in computer science, one is an undergraduate student in computer science and the other annotators are all graduate students in social sciences. None but one have prior experience with annotation tasks. All annotators are native speakers of German with fluent but not native knowledge of English. The annotators are asked to annotate whether the presented question and query both represent the same information need. All pairs where either the query or the question is too general or too specific will be rejected. The exact wording of our annotation guidelines can be found in Appendix B.

We use a two-stage annotation process to ensure high annotation consistency. In the pilot phase all annotators are presented with the same 250 question-query pairs (five queries per question). Uncertainties in question-query pairs where

¹²<https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2>

¹³We once again use the CLS-token to get sentence-level embeddings.

¹⁴<https://github.com/doccano/doccano>

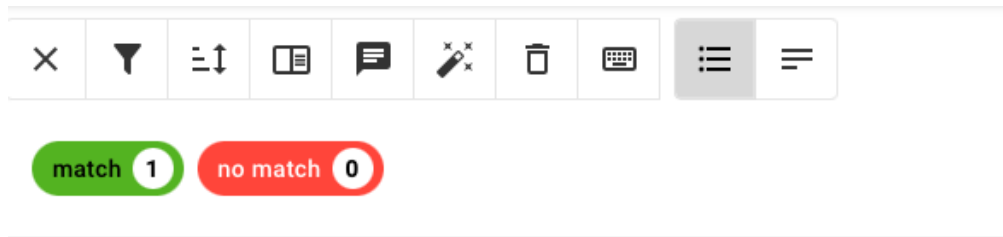


Figure 4.1: Screenshot of the annotation interface in Doccano. The annotator is presented with a question and a query and has to decide whether both represent the same information need.

there is disagreement about the match are then resolved in a group discussion. We also use this discussion to clarify any ambiguities in the annotation guidelines. In the main phase of the annotation, the participants then annotate 625 pairs each. 375 of these pairs are individual for each annotator. The rest of the pairs are the same for each participant. This leads to a total of 2,250 annotations by a single annotator and 250 shared annotations resulting in 2,500 annotations in total. The shared pairs allow us to measure agreement in the main phase as well. The inter-rater agreement can be measured using Fleiss' κ , which compares the observed agreement to the expected agreement. For the pilot study we reach an agreement of $\kappa = 0.45$ what constitutes a moderate agreement [Landis and Koch, 1977]. For the main study we reach an agreement of $\kappa = 0.6$ which, although higher, once again constitutes a moderate agreement.

In total, the annotation process produces 2,487 annotated pairs. The missing 13 pairs had to be dropped due to user errors while using Doccano. We drop a further 55 pairs from the shared pairs where no consensus across all annotators was reached. As a lot of matches follow the pattern of a question asking *what is/are [subject]?* and a keyword query of *[subject]*, we add an additional 60 pairs from the candidates that follow this pattern but were not selected for annotation. This results in 2,468 pairs of which 297 (12%) are matches.

Semi-Supervised Matching: Our decision to use all-MiniLM-L12-v2 and cosine similarity metrics for creating the candidate matches was somewhat arbitrary. To determine the best possible configuration of base model and similarity metric to use when automatically creating a large amount of high quality matches (by maximizing the recall for a fixed precision) we use our annotated pairs to estimate recall and precision values for a range of different combinations of base model, similarity metric, and threshold.

As similarity metrics we try the cosine similarity, the euclidean distance, the inner product, and the manhattan distance. As base models for creating the embeddings we evaluate the following models:

- BioBert¹⁵
- miniLM¹⁶
- LaBSE¹⁷
- T5¹⁸
- MPNet-v2¹⁹

For each combination of similarity metric and base model we compute the similarity of each labeled pair. Combined with the manually created labels this allows us to create a precision-recall curve by varying a similarity threshold and only selecting those pairs with a high enough similarity. Manually inspecting the selected pairs for each fixed precision value shows us that a value of $p = 0.75$ offers a good balance between a large enough result set and high quality matches. We use the configuration that achieves the highest recall at this precision value for the creation of our semi-supervised dataset. The model is a combination of the mpnet-v2 model and the cosine similarity metric. The model achieves a recall of 0.47 at a precision of 0.75 for a similarity threshold of 0.83. Figure 4.2 shows the precision-recall curves for all evaluated models. The chosen configuration and the precision value of 0.75 are marked in the plot. Our chosen configuration is not only the one with the highest recall at this precision value but also the one with the highest overall Area Under the Curve score (AUC) of 0.95. This indicates the best tested performance in the detection of matches when used as a classifier.

Using the selected configuration to filter all previously generated candidate matches results in a total of 340 pairs. This constitutes the result of our

¹⁵<https://huggingface.co/pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb>

¹⁶<https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2>

¹⁷<https://huggingface.co/sentence-transformers/LaBSE>

¹⁸<https://huggingface.co/sentence-transformers/sentence-t5-base>

¹⁹<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

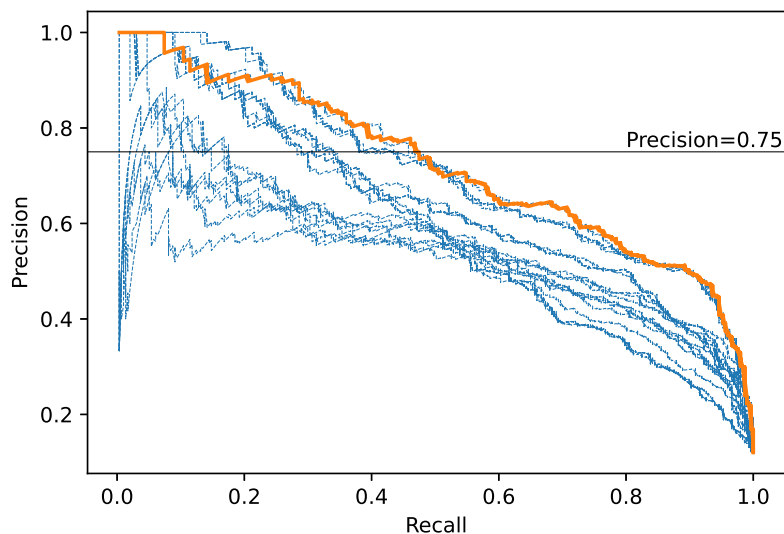


Figure 4.2: Precision-Recall curves for the classification of query-question matches. All evaluated similarity metrics and base models are shown, with our chosen configuration being marked in bold orange.

Table 4.4: Example pairs from our semi-supervised question-query dataset

MashQA Question	Tripclick Query
How is diverticulitis treated?	diverticulitis treatment
Can echinacea help with a cold?	echinacea and common cold
How is cancer-related pain typically treated?	cancer pain management
Can acupuncture help treat insomnia?	acupuncture insomnia

semi-supervised dataset creation. Table 4.4 offers a few examples of the created pairs. While some of the queries only consist of terms also occurring in the question, others are more complex and contain additional terms, also introducing new technical terms such as "pain management" to the query. This illustrates the potential of the sequence-to-sequence model to introduce expert language to the query.

Dataset Finalization: Because the yield of our semi-supervised dataset creation is lower than expected, we decide to add additional pairs to the dataset. To do this, we use the Essential Terms dataset, which is also used for the training of our the extractive preprocessing approach. This adds an additional 2,223 question-question pairs. Note that these additional pairs are

Table 4.5: Hyperparameter prior distributions for training of an sequence-to-sequence model to rewrite questions into queries. All runs use a uniform distribution for sampling from the value range.

Parameter	Value Range
Base Model	t5-small, t5-base
Weight Decay	0 — 0.75
Learning Rate	1e-8 — 1e-3
Epochs	1 — 20
Batch Size	1 — 15
Scheduler	{cosine, linear}
Prompt	{ <code>"Turn question into query", "Define a search query for", "Turn the following question into a clear and concise keyword query"}</code> }
Maximum Length	15 — 64
Repetition Penalty	0.5 — 2.0

not specific to the medical domain and consist only of extractive keyword queries. We still deem the amount of medical data to be sufficient for training our sequence-to-sequence model to work on medical questions.

Training

We fine-tune a T5-based sequence-to-sequence model on the dataset using a random hyperparameter grid search to transform the given natural language questions into keyword queries. The prior distribution for each hyperparameter can be found in Table 4.5. We perform a total of 50 runs using a random 8:1:1 split for training, validation, and testing. When splitting the data, we stratify according to whether the query-question pair comes from our semi-supervised dataset or the essential terms dataset. The best model is trained for 8 epochs using a linear scheduler and reaches a ROUGE-L score of 0.86 on the validation dataset. It uses the `t5-base` model and the longest tested prompt: `Turn the following question into a clear and concise keyword query.`²⁰

Evaluation

We evaluate our fine-tuned models using the ROUGE metrics [Lin and Och, 2004]. Originally devised as a family of metrics for automatic evaluation of

²⁰It uses a batch size of 7, a learning rate of 2.646×10^{-4} , and a weight decay of 0.16. The repetition penalty is 1.146,

Table 4.6: Performance of our fine-tuned sequence-to-sequence model as measured with ROUGE-L and ROUGE-1.

Model	Our Dataset		QSP-Pairs	
	ROUGE-L	ROUGE-1	ROUGE-L	ROUGE-1
Ours	0.85	0.85	0.63	0.64
Iovine et al. [2022]	-	-	0.73	0.78

machine translations, it has been used before to evaluate generation of keyword queries from natural language questions [Iovine et al., 2022]. Given a reference X of length m and a generated string Y the ROUGE-L score is defined as

$$R_L(X, Y) = \frac{LCS(X, Y)}{m}$$

with $LCS(X, Y)$ being the length of the longest common subsequence of X and Y . ROUGE-1 refers to the recall of unigrams between generated string and ground truth.

We select the fine-tuned model with the highest ROUGE-L score on the validation dataset. The model is then evaluated on both our test dataset and the test dataset provided by Iovine et al. [2022], which contains question-query pairs from an open domain. The results can be found in Table 4.6. We note that our approach underperforms the approach by Iovine et al. [2022] on the QSP-Pairs dataset. However, the performance on our own dataset is significantly higher. We attribute this to the fact that the datasets differ in their nature. Our dataset has a significant bias to health-related questions. Due to the fact that Iovine et al. [2022] do not publish model weights, we are unable to compare the performance on our dataset.

4.2.3 Comparing Preprocessing Techniques

After completing our preprocessing techniques we compare the two approaches. Table 4.7 shows examples of generated queries. Note that both proposed approaches generate keyword queries that are very close to the original question and mainly remove stopwords. Albeit being able to completely transform the question, 54% of terms generated by the sequence-to-sequence model were already present in the input. In part, the generative approach only augments small details, e.g., by transforming "migraines" into the singular "migraine".²¹

²¹The American Migraine Foundation explicitly favours the singular as it better captures the chronic character of the disease: <https://americanmigrainefoundation.org/resource-library/migraine-terms/>

Table 4.7: Examples of queries generated by the two preprocessing techniques.

Which medications can treat alzheimer’s disease?	
Essential Terms	medications treat alzheimer’s disease
Sequence-to-Sequence	alzheimer’s disease treatment
How can acupuncture help treat migraines?	
Essential Terms	acupuncture treat migraines
Sequence-to-Sequence	acupuncture migraine
How is fatigue related to multiple sclerosis?	
Essential Terms	fatigue related multiple sclerosis
Sequence-to-Sequence	multiple sclerosis fatigue

Table 4.8: Effect of preprocessing approaches on query and term length.

System	# characters	# terms	term length
Original	45.5	7.4	5.4
Essential Terms	26.5	3.3	7.3
Sequence-to-Sequence	21.5	2.6	7.5

Overall both approaches result in similar queries: 22% of all questions in the test set are transformed into the same query. However the results of the generative approach seem to be slightly shorter and contain fewer terms. Table 4.8 confirms this impression as both the overall length as well as the term count is reduced. However, we observe slightly longer terms in the sequence-to-sequence approach. We hypothesize that this might be due to a more scientific language used in the queries of the Tripclick dataset that was used to train the model.

4.2.4 Ablation: Query Preprocessing on TREC 2022 Health Misinformation Track

To evaluate the impact of preprocessing on the retrieval effectiveness we use the TREC 2022 Health Misinformation Track. The web retrieval task is on the retrieval of documents for consumer health search from the C4 noclean²² corpus. For all topics, a yes-no question in natural language is given. The retrieval systems are evaluated according to their likelihood of retrieving helpful documents and their resilience against retrieving harmful documents. This

²²<https://huggingface.co/datasets/allenai/c4>

Table 4.9: Retrieval effectiveness scores of our preprocessors on the TREC 2022 Health Misinformation Track. For compatibility with helpful documents, a higher score is better. For harmful documents, lower is better.

System	Compatibility			
	nDCG	Harmful	Helpful	Difference Helpful, Harmful
Sequence-to-Sequence	0.17	0.09	0.13	0.04
Essential Terms	0.19	0.11	0.17	0.06
Stopword Removal	0.21	0.10	0.16	0.06
Original Question	0.22	0.11	0.18	0.07

is measured using the compatibility metric. Additionally we evaluate the systems’ retrieval ability to retrieve useful documents using the normalized discounted cumulative gain (nDCG) metric. All used metrics are explained in greater detail in Section 4.4. We combine our preprocessing approaches with the default lexical BM25 retrieval model from Elasticsearch without any additional reranking. As we have observed that our preprocessors seem to remove stopwords, we additionally evaluate the impact of a preprocessor consisting solely of stopwords removal. We implement this using the NLTK library’s english stopwords list [Bird et al., 2009].

Table 4.9 shows the results for each system. We observe that the scores for all tested systems are very close to each other. The sequence-to-sequence model shows a slight advantage over the other systems in terms the likelihood to retrieve harmful documents while the original question performs best with regards to the retrieval of helpful documents. Using a question as query also performs best in terms of nDCG. We do not measure any significant differences between the systems ($p < 0.05$). Compared to the other runs submitted for the shared task, all of our measured metrics would place our systems very low in the ranking [Clarke et al., 2022]. We attribute this to the fact the we only use a very simple retrieval model, because only the impact of our preprocessing techniques is of interest at this stage.

While this ablation study does not show an improvement in retrieval effectiveness, we also do not observe a decrease. Note that the Misinformation Track only features yes-no questions while our models were trained on multiple choice (Essential Terms) and a mix of the former and non-factual questions with open answers (Sequence-to-Sequence). This may have led to our models not being able to fully leverage their potential. For this reason we decide to continue with both approaches in the next steps of our research.

4.3 Creating a Retrieval Pipeline

Given our focus in this work lies on creating a QA system, we opt against creating our own retrieval system from scratch. Instead, we decide to combine existing retrieval approaches in a pipeline for our use case of retrieving relevant documents for health-related questions. We then perform a grid search over different retrieval settings and optimize the pipeline with regards to retrieval effectiveness. For this we divide our retrieval process into multiple stages:

1. The first stage uses lexical retrieval to retrieve a large amount of documents using the original question.
2. We then perform a preprocessing step to transform the given question into a keyword query.
3. We perform an additional lexical retrieval step by reranking the documents retrieved in the first step using BM25.
4. We perform a rank cutoff to reduce the amount of documents to rerank.
5. We split the reranked documents into paragraphs using a sliding window approach. This step is fixed across all runs in the grid search.
6. For the second reranking step we use a dense retrieval model. We try both monoT5 [Pradeep et al., 2021] and ANCE [Xiong et al., 2020] in this step.
7. We perform another rank cutoff to reduce the amount of documents.
8. Lastly we rerank the paragraphs using duoT5 [Pradeep et al., 2021].

We use the Clueweb22-B [Overwijk et al., 2022] corpus that consists of 200 million documents for the initial retrieval. We filter out all documents that are not from reputable sources as described in the previous section. This leaves around 22 million documents in the corpus. The documents are indexed and retrieved using Elasticsearch with the search engine’s default BM25 retrieval parameters. This step is fixed and not part of the grid search as we do not have access to change the retrieval model of the search engine. We retrieve an initial set of 500 documents for each question.

For the preprocessing step we use the sequence-to-sequence model and the keyword extraction model described in the previous section. Additionally we evaluate the search using the original question as a query. While the first reranking step with BM25 after using the same retrieval system for the initial retrieval might seem redundant, we use this step to vary BM25’s parameters.

Table 4.10: Value ranges used in the gridsearch for all steps in the retrieval pipeline. We also evaluate skipping the last reranking step entirely. Our selected pipeline configuration is marked in bold.

Parameter	Value range
Preprocessing	
System	{Sequence-to-Sequence, Essential Terms, None }
First reranker	
BM25 k_1	{0.6, 2.5, 4.4 }
BM25 b	{0.15, 0.45, 0.75 }
Second reranker	
Rank Cutoff	{ 50 , 100}
Model	{ monoT5 , ANCE-FirstP}
Third reranker	
Rank Cutoff	{5, 10 }
Model	{ duoT5 , None}

For this, we use similar parameter ranges to those used by Lin [2018]. We somewhat reduce the range of parameters to limit the amount of configurations we have to evaluate.

In total the combinations of configurations result in a total of 324 different runs. The used parameter ranges are shown in Table 4.10. All retrieval pipelines are implemented using the Python library `pyterrier` [Macdonald and Tonellotto, 2020]. For the ANCE reranker we use the `pyterrier_dr` package.²³ Both `monoT5` and `duoT5` are implemented using `pyterrier_t5`.²⁴

4.4 Evaluation

We evaluate our systems purely with regards to retrieval effectiveness, ignoring runtime and resource usage. For the grid search we use the training topics from our MashTREC dataset. The best performing pipeline is then evaluated on the test set. All evaluations are performed using manual annotations of the retrieval results up to a depth of 5. Similar to the evaluations performed in the TREC Health Misinformation track [Clarke et al., 2020a] we rate retrieved snippets

²³https://github.com/terrierteam/pyterrier_dr

²⁴https://github.com/terrierteam/pyterrier_t5

Table 4.11: Scoring system for combined relevance and correctness judgements.

Description	Score
The snippet is relevant and correct.	4
The snippet is partially relevant and correct.	3
The snippet is relevant and correctness is unknown.	2
The snippet is partially relevant and correctness is unknown.	1
The snippet is not relevant.	0
The snippet is partially relevant and incorrect.	-1
The snippet is relevant and incorrect.	-2

not only with regards to their topical relevance but also their correctness. Since we only offer narratives for topics in the test dataset, we create in depth annotation guidelines for the relevance judgements to help the annotators create high quality annotations. The guidelines are available in Appendices C and D.

We use a three point scale for relevance judgements: "Relevant", "Partially relevant", and "Not relevant". All snippets deemed either "Relevant" or "Partially Relevant" are additionally annotated with regards to their correctness. Because the manual relevance judgements are not performed by medical professionals, we define correctness as whether or not the snippet aligns with the information presented in the reference answer from MashQA. Due to the open nature of the questions, we judge a snippet as "Correct" if there is any overlap between the snippet and the reference answer. Only if there are clear contradictions between snippet and reference, the snippet is judged as "Incorrect". All other snippets are annotated as "Correctness Unknown".

We derive a numerical score from the combinations of the relevance and correctness judgements. The resulting scoring system that is heavily inspired by the TREC Health Misinformation track [Clarke et al., 2020a] and follows the idea of rewarding correct information and penalizing incorrect information. The systems prefers clearly irrelevant results to those that are relevant but contain misinformation. The full mapping is shown in Table 4.11.

4.4.1 Metrics

To rank the retrieval systems we compute several metrics that use both the relevance and correctness judgements. For all metrics we use the Python package `ir_measures`.²⁵

²⁵https://github.com/terrierteam/ir_measures

Table 4.12: Illustrating the counts of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) in a retrieval scenario.

Retrieved	Relevant	
	Yes	No
Yes	TP	FP
No	FN	TN

Precision and Recall

Precision and recall provide a way to evaluate the binary relevance of retrieval results. To do this, we binarize our relevance judgments by considering only "relevant" and "partially relevant" as relevant and "not relevant" as not relevant.

Each document in the corpus can be classified as either true positive (TP), false positive (FP), false negative (FN) or true negative (TN) as described in Table 4.12. The precision P measures the proportion of retrieved documents for a topic that are relevant:

$$P = \frac{TP}{TP + FP}.$$

Recall is the proportion of relevant documents for a topic that are retrieved:

$$R = \frac{TP}{TP + FN}.$$

To combine precision and recall into a single score we calculate their harmonic mean, the F_1 -score:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}.$$

Note that the metrics do not take the position of the relevant documents into account and are highly dependent on the depth of the evaluation. We macro-average the precision, recall, and F_1 -score over all topics to get the final scores for each system.

Normalized Discounted Cumulative Gain (nDCG)

Precision and recall metrics can only be computed using binary relevance judgements. Additionally, these metrics do not take the position of the retrieved documents into account. Therefore we also calculate the normalized discounted cumulative gain (nDCG) [Järvelin and Kekäläinen, 2000, 2002] for the retrieval

results. Compared to precision and recall that only allow for binary relevance judgements, nDCG allows for graded relevance judgements. Furthermore the metric takes the position of each retrieved document into account. The further down the list a relevant document is, the less it contributes to the nDCG score. To compute the nDCG for one topics at depth p we first calculate the discounted cumulative gain (DCG):

$$\text{DCG}_p = \sum_{i=1}^p \frac{\text{rel}_i}{\log_2(i+1)}$$

with rel_i being the relevance score of the document at position i . Because different topics might have different numbers of relevant documents, we normalize the DCG by the ideal DCG (IDCG) that is reachable with the retrieved results (based on all relevance judgements we have for each topic) to get the nDCG:

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p}.$$

We use the relevance score 2 for relevant documents, 1 for partially relevant documents and 0 for not relevant documents. The resulting nDCG scores are then averaged over all topics to get the final nDCG score for each system.

Compatibility

The metrics discussed so far only take the relevance judgements into account. While one may use the scores defined in Table 4.11 to calculate an nDCG score that takes the correctness judgements into account, we follow Clarke et al. [2020a] and instead opt for the compatibility metric [Clarke et al., 2020b]. The scores mapped to each judgement are somewhat arbitrary. Changing the scores could lead to very different nDCG scores. While we do not necessarily trust the exact scores we chose, we do trust the relative order of the scores. The compatibility score takes that into account by comparing the order of retrieved documents to a theoretical ideal order as defined by our preferences.

For each ranking of retrieved documents R , one can define an ideal ranking I . Within the ideal ranking all documents (not only those retrieved in R) are sorted according to a given preference order. The ideal ranking I is specific for each real ranking R : Documents that are in R always rank higher than documents that are not in R and have the same preference score. At each preference rank the ideal ordering starts with the documents that are in R in the retrieved order and then continue with the documents that are not in R in an arbitrary order. The rank biased overlap (RBO) [Webber et al., 2010] that measures the similarity between the two rankings can then be calculated:

$$\text{RBO}(R, I) = (1 - p) \sum_i^{\infty} p^{i-1} \frac{|R_{1:i} \cap I_{i:1}|}{i}$$

with $R_{1:i}$ and $I_{i:1}$ being the top i documents in R and I respectively. The parameter p is a weight that determines how much the RBO score is biased towards the beginning of the rankings. We use $p = 0.98$ as suggested by Webber et al. [2010]. As we only judge the top five documents for each topic, we only calculate the sum up to $i = 5$. The resulting similarity scores across all topic are then averaged to get the compatibility score for each system.

We use the metric in two ways to evaluate the retrieval systems. First we calculate the overall compatibility score using the relevance scores as preference scores (we call this the "Overall Compatibility" of a system). This allows a good overall evaluation of the retrieval systems taking both relevance and correctness into account. Second, we use the metric to place the evaluated retrieval systems in a two dimensional space: We can measure the compatibility to documents considered helpful separately from the compatibility to documents considered harmful. In theory this allows us to select a retrieval system that retrieves documents that are helpful while avoiding harmful documents (indicated by a high compatibility with helpful documents and a low compatibility with harmful documents). For this sake we divide our judgements into two groups: All documents with a score higher than 0 are considered helpful and all documents with a score lower than 0 are considered harmful. We take the absolute value of the score for harmful documents. We then calculate the compatibility score for helpful and harmful documents separately. A good retrieval system should have a high compatibility score for helpful documents and a low compatibility score for harmful documents. This approach to evaluate retrieval systems is taken from the TREC Health Misinformation track [Clarke et al., 2022].

The overall compatibility is the main metric in our grid search. However, we still report all other measured metrics to give a better understanding of the retrieval systems.

4.4.2 Grid Search

The retrieval results for all systems in our grid search are judged by five annotators. These annotators are one PhD student of computer science, one graduate student of computer science, two bachelor students of computer science and one graduate of social sciences. All annotators are fluent in English and one is a native speaker. Only the first two annotators reported prior experience with text annotation. The guidelines for the annotation process are available in Appendix C.

We use a two-step annotation process to ensure high quality annotations with a high level of agreement between annotators. The first stage consists of a shared annotation of 103 query-document pairs.²⁶

This allows us to measure the initial agreement between annotators and resolve any disagreements in a group discussion. We measure the agreement using Fleiss' κ : We reach a value of $\kappa = 0.31$ which is considered fair agreement [Landis and Koch, 1977]. When evaluating agreement on correctness and relevance separately, we reach values of $\kappa = 0.32$ for relevance ($\kappa = 0.44$ for binarized relevance where partially relevant documents are considered relevant) and $\kappa = 0.27$ for correctness. In the second phase of the annotation process we split the remaining document pairs between the annotators. Additionally, 99 query-document pairs are shared by all annotators.²⁷ We measure the agreement using Fleiss' κ again: The overall agreement increases to $\kappa = 0.43$, agreement on relevance is $\kappa = 0.63$ (0.74 for binarized judgements) and agreement on correctness is $\kappa = 0.47$.

Note that during this initial phase of the annotation process we did not use the category "Correctness Unknown" and only used "Correct" and "Incorrect". All documents that did mention any of the information presented in the ground truth were labeled as "Incorrect", even if there was no contradiction between the snippet and the ground truth. We hypothesize that this unintuitive labelling led to the noticeably lower agreement with regards to correctness. To resolve this issue we added the category "Correctness Unknown" and reannotated the correctness for all documents that were labeled as "Relevant" or "Partially Relevant" and "Incorrect" initially. This reannotation was done by the two annotators with prior experience. We derive unambiguous judgments from shared annotation by majority vote.

Results

The overall distribution of our evaluation metrics is shown in Figure 4.3. In particular, we see a large variance between the systems for the precision measured at the first rank.²⁸ Because we plan to use the retrieval results as input for an answer generation system, we are interested in the precision at the first rank. This is due to the fact, that only a limited amount of context documents can be passed to the generation step of the pipeline. Figure 4.4

²⁶We decide to let each annotator make relevance judgments for all retrieved documents of three queries. We choose this number because it gives us a good number of query-document pairs, close to 100.

²⁷Again, we create shared relevance judgements for the retrieval results for three queries. Since the number of unique results per query varies, there are slightly fewer shared pairs in the main annotation process.

²⁸That is the percentage of relevant results retrieved at rank 1.

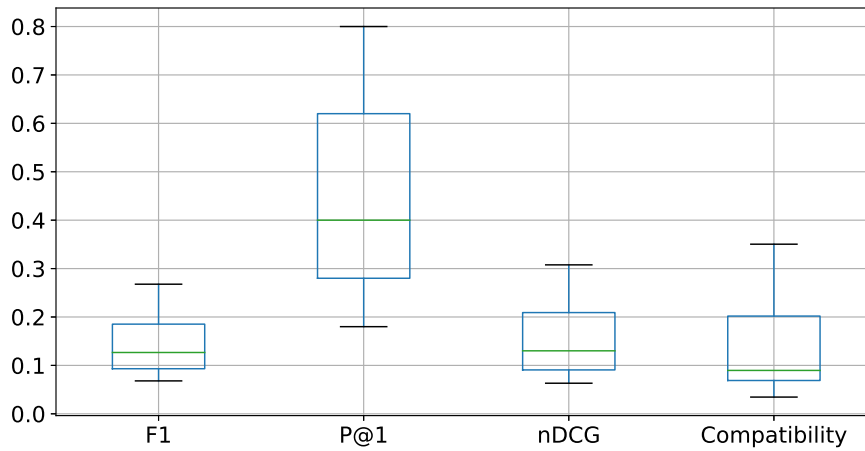


Figure 4.3: Distribution of relevance-focussed retrieval effectiveness metrics for all systems evaluated in the grid search.

shows how the different hyperparameters influence the retrieval effectiveness as measured by the overall compatibility. For the parameters of our initial BM25 step we observe that there is a clear trend towards higher compatibility scores for higher values of b and lower values for k_1 . While the median scores for all preprocessors are very close to each other, there are some runs using no preprocessing that score noticeably higher than any run using a preprocessor. While no clear trend for any choice is visible for the rank cutoffs between each reranker, the choice of the reranker itself seems to have a large influence on the overall compatibility. There are more high scoring systems that use monoT5 than ANCE. The additional duoT5-based reranking step also increases the

Table 4.13: Hyperparameters of the best performing retrieval system in the grid search.

Hyperparameter	Value
Preprocessor	None
BM25 k_1	4.4
BM25 b	0.75
First Rank Cutoff	50
First Reranker	monoT5
Second Rank Cutoff	10
Second Reranker	duoT5

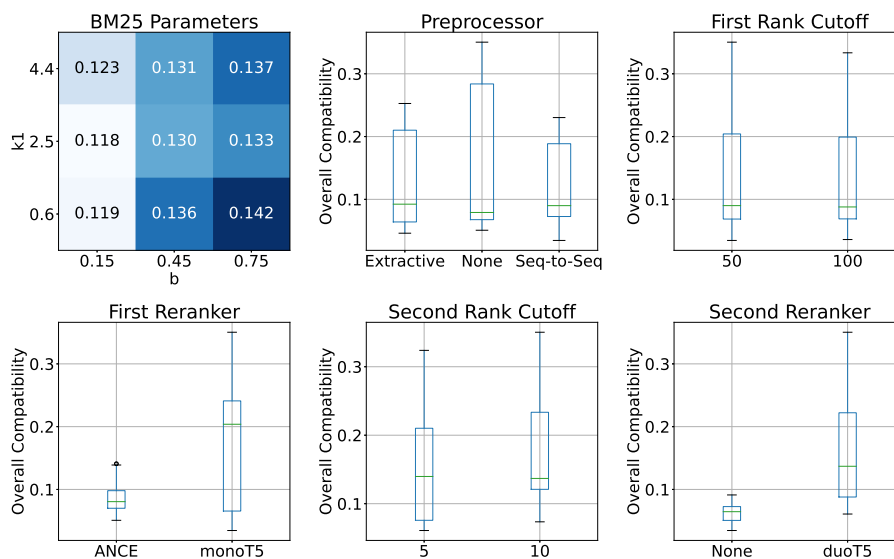


Figure 4.4: Distribution of overall compatibility scores for each step of the retrieval pipeline.

compatibility. These observations are confirmed when we look at the best performing system in the grid search with regards to the overall compatibility (cf. Table 4.13). With the exception of k_1 , all hyperparameters of our chosen system follow all trends described above. The retrieval pipeline scores highest or high in nearly all of the evaluation metrics we measured (cf. Table 4.14).

The notable exception to these results is the compatibility with harmful documents. Our chosen system scores relatively high in this metric which might indicate a high likelihood of retrieving harmful documents. We investigate this further by looking at the spacing of our system in the two-dimensional space of compatibility to helpful and harmful documents as discussed before. Figure 4.5 shows the positioning of all evaluated systems in this space. Ideally, we would like to see a high compatibility with helpful documents and a low compatibility with harmful documents, indicated by a dot in the bottom right corner of the plot. Overall, the 2-dimensional scores of the systems seem to indicate, that a high compatibility with helpful documents is often accompanied by a high compatibility with harmful documents. The bottom right corner of the plot is mostly empty. However, we also note that the compatibility with harmful documents is generally very low. The average compatibility score with harmful documents is 0.02. Additionally, investigating the compatibility scores for

Table 4.14: Evaluation metrics for the best performing retrieval system in the grid search. The rank column shows the rank of the system compared to all other evaluated systems. The ranking for the compatibility with harmful documents is inverted. In total, 324 systems are evaluated.

Metric	Value	Rank
Overall Compatibility	0.35	1
Compatibility with Helpful Documents	0.38	2
Compatibility with Harmful Documents	0.03	260
Δ of Harmful and Helpful Compatibility	0.35	2
F_1	0.27	1
nDCG	0.31	1
P@1	0.78	2

each topic (cf. Figure 4.6) we see that there is no compatibility with harmful documents whatsoever for most topics. We retrieve harmful documents for only a few topics, in fact our chosen system only retrieves harmful documents for 2 out of 50 topics. So while our selected system scores high in the compatibility with harmful documents, this is still not indicative of a high likelihood of retrieving harmful documents. Similar to Clarke et al. [2022], we additionally measure the difference between the compatibility with helpful and harmful documents. For this metric, our system scores second highest in the grid search.

4.4.3 Test Set

We retrieve passages for each test topic with the best performing system as determined by the grid search. Again, we evaluate the retrieval up to a depth of 5. Our annotation guidelines are very similar to those used for the grid search. However, we add the narratives taken from the TREC topics to each question to give the annotators a better understanding of the context of the question. The guidelines are available in Appendix D. Three annotators that already participated in the grid search annotation process are used to judge the retrieval results. Since the annotators are already familiar with the annotation process, we only conduct one round of annotation. All annotators annotate the same query-document pairs, resulting in three judgements for each document. Overall we reach a Fleiss’ κ of 0.21 which constitutes a fair agreement [Landis and Koch, 1977]. The agreement on the binarized relevance judgements is $\kappa = 0.46$ and there is an agreement of $\kappa = 0.34$ on the correctness.

As all documents are judged by three annotators, we derive unambiguous judgements from shared annotation by using a majority vote. We determine

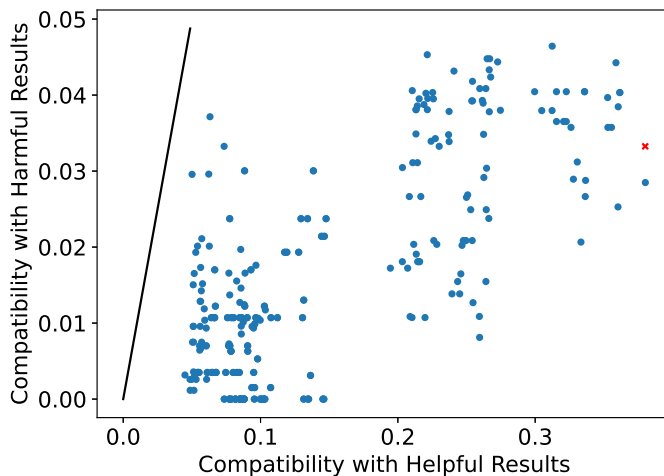


Figure 4.5: Aggregated compatibility scores for each retrieval system in the grid search. Each dot represents the mean of the compatibility scores across all topics for one system. The retrieval system chosen as the best performing system is highlighted with a red cross. The line indicates an equal compatibility with helpful and harmful documents.

the majority vote separately for relevance and correctness and then combine the two to get the final judgement for each document. We then calculate the evaluation metrics as described before.

Results

Table 4.15 shows the evaluation metrics for our selected retrieval system on the test topics. Notice that the system scores very high on all metrics - in fact, it significantly outperforms its own performance on the training topics on most of the metrics measured. This is surprising, as we would expect comparable performances on both datasets.

While the training set was sampled to follow a similar distribution of question prefixes as the test set, the details of the questions may be different. Note that the test set was created by matching questions to previous TREC topics, while the training set was selected based on prefixes only. Since the TREC topics we match to are likely hand-picked, there could be a difference in retrievability between our topic sets. This could lead to a difference in the retrieval effectiveness of our retrieval system. In addition, we only provide narratives for the test topics. Since the narratives typically help the annotators to better understand the context of the question, this could also lead to a higher probability of classifying retrieved documents as relevant. We note these

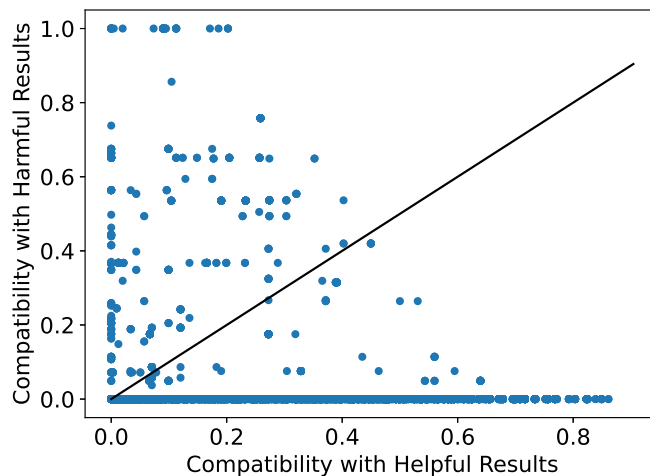


Figure 4.6: Compatibility scores for each topic in the grid search. Each dot represents the scores for the results of one system for one topic. Note that the overwhelming majority of the compatibility scores with harmful documents is 0. This is underlined by the average harmfulness compatibility score of 0.02. The line indicates an equal compatibility with helpful and harmful documents.

observations as limitations of our evaluation process, but continue with our chosen system as the overall results seem very promising.

4.4.4 Evaluation on TREC 2022 Health Misinformation

To get an additional perspective on the retrieval effectiveness of our chosen system we decide to conduct an additional evaluation. We use the TREC 2022 Health Misinformation Track as a benchmark. The track provides a set of 50 health-related topics that are expressed as yes-no questions. Judgements for retrieval from the C4 noclean dataset by Google²⁹ are provided by the track’s organizers. Similar to our prior evaluation, we measure compatibility to both harmful and helpful documents as well as the retrieval effectiveness with regards to usefulness of the retrieved documents.

The retrieval pipeline we produced in our grid search needs to be slightly modified for the evaluation: we originally designed the pipeline to retrieve passages, but the Misinfo Track only provides judgments at the document level. Therefore, we apply a transformation to our retrieval results that ranks each originally retrieved document (before applying the sliding window transformation that produces our passages) according to its highest ranked

²⁹<https://huggingface.co/datasets/allenai/c4>

Table 4.15: Evaluation metrics for our selected retrieval system on the test topics. The dagger[†] indicates a significant increase (or decrease for compatibility to harmful documents) compared to the metrics on the training topics ($p < 0.05$).

Metric	Value	σ^2
Overall Compatibility	0.81 [†]	0.05
Compatibility Helpful Documents	0.81 [†]	0.05
Compatibility Harmful Documents	0.00	0.00
F ₁	0.91 [†]	0.03
nDCG	0.95 [†]	0.01
P@1	0.89	0.10

Table 4.16: Evaluation results for our selected retrieval system in the TREC 2022 Health Misinformation Track. In addition to our results, we report the metrics for the best systems in terms of compatibility with helpful and harmful documents, respectively. The nDCG results and median scores are taken from Bondarenko et al. [2022], the best systems are taken from the official track report [Clarke et al., 2022].

System	Compatibility			nDCG
	Helpful	Harmful	Difference Helpful, Harmful	
Ours	0.19	0.15	0.04	0.50
hm22_ref_comb.vera_t5	0.35	0.09	0.26	-
webis-longck-ax-lin	0.11	0.07	0.08	0.49
<i>Median All Participants</i>	0.24	0.13	0.11	0.69

passage. Instead of discarding the documents filtered out by each rank cutoff, we reconfigure our system to rerank only the top k passages at each step. Thus, we produce a ranking up to a depth of 500.

We show the results of our evaluation in Table 4.16, comparing our system to the median of each metric across all runs for the shared task, as well as to the best systems with respect to both compatibility metrics. Overall, our system scores slightly below the median for each metric. A run with our system would have placed us in the bottom third in terms of the difference in compatibility scores for helpful and harmful documents.

While our approach does not rank highly, it is important to understand the limitations of the evaluation. Our grid search was designed to optimize retrieval effectiveness for the top five passages. We did not aim to produce a good ranking with a depth of 500, and we did not optimize the ranking of full

documents. In addition, we optimized our system on a dataset of non-factual questions, whereas TREC Health Misinformation contains yes-no questions.

4.5 Summary

In this chapter we have discussed the retrieval process for our system. We first addressed the challenge of health misinformation by only retrieving from trustworthy sources. Because we want to use sources that are understandable to the general public, we opt to retrieve from a corpus of webpages instead of scholarly articles. We create a list of reputable sources by combining several sources of trustworthy domains. This list is then used to filter the Clueweb22-B corpus to only include documents from these domains.

Then, we developed two approaches to transform questions from a natural language form into keyword queries. One preprocessor followed an extractive approach, using a classifier to identify the most important words in a question. The second preprocessor used a generative approach, allowing new words to be added to the query. Initial tests on the TREC Health Misinformation track showed no advantage in retrieval effectiveness for either approach.

We then built our own retrieval system by combining existing rankers and rerankers. We used a grid search to determine the best performing retrieval configuration on our training questions, using manual relevance judgements to determine the retrieval effectiveness. We selected the best system with regards to the overall compatibility score and note that it scores highly across most measured metrics. The notable exception to this is the compatibility to harmful documents. Given the low overall number of harmful documents we retrieved, we consider this to be an acceptable result.

When evaluating our chosen retrieval system on our test set, we notice a significant increase in some metrics compared to the training data. We attribute this to slight differences in the genesis of the topics. We perform an additional evaluation on the TREC Health Misinformation track. Our system performs poorly on this benchmark. We attribute this to the fact that we had to modify our system slightly to fit the TREC evaluation setup. In addition, we note that the questions in our MashTREC dataset are not factual in nature, while the questions in the Misinformation track are all yes-no questions.

Chapter 5

Generating Long Form Answers

In this chapter we describe the process of generating long form answers to our questions using the retrieved passages as context. We opt for a retrieval-augmented generation approach as first introduced by Lewis et al. [2020]. We decide to treat the generative model as a black box and focus on optimizing the input prompt as well as the document processing. Additionally this allows us to exchange the language model we use and report our system’s answer quality when using different LLMs.

First, we describe the construction of the generative component before evaluating the generated answers. Since the generated answers to the questions from our test set are the end result of our question answering pipeline, we provide an in-depth analysis of the generated answers. We perform both an automatic evaluation, which compares the generated answers to the reference answers from MashQA, and a manual evaluation, in which experts evaluate the quality of the generated answers. We conclude the chapter with an ablation study that evaluates our generative system without the retrieval component.

5.1 Prompt Engineering

To generate long-form answers to our questions, we use a large language model. We use the Mistral7B model [Jiang et al., 2023a] released by MistralAI in 2023 in the prompt engineering process. The model weights are available for free download.¹ We treat the choice of the exact phrasing of the prompt as a grid search problem, using different aspects of the prompt as hyperparameters that are varied. We experiment with different ways of providing context to the model, using system prompts, and one- or few-shot examples. However, we use Mistral7B for all runs in the grid search.

¹<https://huggingface.co/mistralai/Mistral-7B-v0.1>

We then evaluate the generated answers using automatic evaluation metrics and select the best performing prompt configuration to generate the answers for the test set. In total, the grid search evaluates 96 different prompt configurations. We optimize the following hyperparameters:

- Mistral7B provides the ability to add a system prompt to the input. This prompt is indicated by a special token and defines the general scheme of the generated answers. We experiment with three system prompts and evaluate the effect of using no system prompt at all.
- We experiment with using either one or three snippets as context for the model.
- We evaluate the usage of a summarizer to summarize the snippets before passing them as context to the generation model. Additionally, we try using no summarizer at all.
- We try different ways of providing examples to the model. We experiment with using either one- or three-shot examples combining any of five selected example triplets of question, context, and reference answer.

The different system prompts tested differ mainly in the specificity of information given to the model with regards to the expected content of the answer. While one system prompt is very specific and goes into great detail about the details of the QA setting and the expected content of the answer, the other two are more general and only provide a general scheme of the expected answer. The different system prompts we evaluated can be found in Appendix F. The longest system prompt explicitly asks the LLM to refuse giving an answer if the context does not provide enough information to answer the question. While we want the system to generate answers to all questions, rejecting to answer can be desirable, as it can help to avoid the answers not based on any information it found in the retrieved trustworthy documents.

To prime the model to generate answers that are similar to the retrieved snippets, we evaluate the use of one-shot and few-shot prompting. We use actual snippets retrieved by our chosen retrieval system as examples. We randomly select five topics from the training dataset that we retrieved relevant and correct snippets at rank one for and evaluate using either one or three examples that combine any of the five questions. The example questions are removed from the training set. One or three triplets consisting of question, snippet, and answer are then presented to the model before it is presented with the actual question and context from the test set.

For the summarization step we use the `facebook/bart-large-cnn` model.² We implement the summarization step using a pipeline provided by the Hugging Face Transformers library³ [Wolf et al., 2020]. Manual evaluation of the retrieved snippets showed that they often contain noise such as HTML-tags or incomplete sentences. During early testing, we found that a summarization step can help to distill relevant information from the retrieved snippets.

5.2 Evaluation

Following the evaluation of a RAG system for answering clinical questions presented by Zakka et al. [2024], we evaluate our generated answers on the three dimensions of factuality, completeness, and human preference. However, we differ from the proposed evaluation process by limiting the amount of human input required. We do this because our access to experts in the medical domain is limited and the number of systems to evaluate in the grid search is large. We decide to use only manual evaluation (and thus human preference) for the test set, and use purely automatic evaluation metrics for the grid search.

The overall evaluation process comes with two pitfalls. First, the automatic metrics compare the generated answers to a given reference answer. This assumes that the reference answer is of the highest quality and complete. While the answers provided by the MashQA dataset are written by medical professionals, annotators in the manual evaluation process of the retrieval system already noted that some retrieved snippets are more complete and understandable than the given reference answer. We still use automatic metrics because we perform an additional manual evaluation step later. This is then used to test the automatic metrics' reliability.

The second pitfall is the possibility of train-test leakage. Neither MistralAI nor OpenAI disclose the exact training data used for their models. This leaves the possibility that the model has seen the reference answer during training. We are aware of this possibility. In the manual evaluation process we decide to include the reference answer when measuring preference for the generated answers. This enables us to detect if the system's answers are of higher quality than the ground truth, indicating that the models are not just repeating the reference answer they already saw during training.

²<https://huggingface.co/facebook/bart-large-cnn>

³<https://huggingface.co/docs/transformers/>

5.2.1 Automatic Metrics

Our automatic evaluation of factuality and completeness is partially based on the RAGAs framework for automatic evaluation of RAG systems [Es et al., 2023]. As the proposed metrics require the use of OpenAI’s inference API, we decide to split our grid search into two stages. We use SemScore, which does not use any of OpenAI’s models, for the first stage to determine the top five configurations. The top five configurations are then evaluated further using RAGAs-based metrics. This split is necessary as the usage of OpenAI’s API is costly and we work with a limited budget.

We use a total of three different automatic evaluation metrics to evaluate the generated answers. While SemScore measures the semantic similarity between generated answer and the given reference, the RAGAs metric of Correctness and Completeness offer a more fine grained evaluation.

SemScore

SemScore is a metric that was developed for the automatic evaluation of instruction-tuned LLMs [Aynedinov and Akbik, 2024]. It works by comparing the generated answer to the ground truth by computing the semantic similarity as measured by the cosine similarity between the embeddings of the generated answer and the ground truth. Given two vector representations x and y , the cosine similarity is the cosine of the angle θ between x and y :

$$\cos \theta = \frac{x \cdot y}{\|x\| \|y\|}.$$

The produced value is in the range $[-1, 1]$, where 1 indicates a very high similarity and -1 a very high dissimilarity. Aynedinov and Akbik [2024] recommend to use the `all-mpnet-base-v2`⁴ base model to create the embeddings and measure the similarity of the inputs’ CLS-tokens. We follow this recommendation. Using the described configuration SemScore has the highest rank correlation with human preference judgments when comparing it to other automatic evaluation metrics such as ROUGE or BLEU [Aynedinov and Akbik, 2024]. The metric can capture high similarity even in the case of a complete rephrasing of the ground truth without any overlap in the words used. SemScore is identical to the Relevance metric as proposed by RAGAs⁵ with the only difference being the model used to create the embeddings.

⁴<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁵https://docs.ragas.io/en/latest/concepts/metrics/answer_relevance.html

Correctness

The Correctness metric, as provided by RAGAs⁶, takes an approach where question answering scenarios are treated as retrieval problems. Here, the reference answers are considered as a set of relevant statements that may or may not be present in the generated answer. Furthermore, a generated answer might include irrelevant statements. This perspective enables us to calculate the amount of true positives, false positives, true negatives, and false negatives, which can subsequently be combined using precision, recall, and F_1 -scores.

We first submit the triplet of question, reference answer, and generated answer to the OpenAI-API to extract statements only in the ground truth and statements only in the generated answer as well as statements in both (This creates the three classes of False Negatives, False Positives, and True Positives). For this we use the GPT3.5 inference endpoint. The exact prompts we use is available in RAGAs' Github repository.⁷ We then calculate the precision, recall for each generated answer. A system's Correctness score is the macro-averaged F_1 -score across all generated answers.

Completeness

We use OpenAI's GPT3.5 model to measure a generated answer's completeness with regards to three different aspects: We measure if the answer addresses all aspects of the question, if the answer omits any important content and if the answer contains any irrelevant content. To create these three scores we submit question, reference answer, and generated answer to the OpenAI-API and ask the model to produce three distinct scores for each of the three aspects. The triplet of scores is then aggregated using the arithmetic mean to get a single score. To get a system's completeness score, we average the scores of all the system's answers. The exact prompt we use is available in Appendix I.⁸

We must note that this metric has not been validated in a scientific setting before and thus may not be as reliable as the other used metrics. However, we include it in our evaluation to get a more fine grained evaluation of the generated answers.

⁶https://docs.ragas.io/en/latest/concepts/metrics/answer_correctness.html

⁷https://github.com/explodinggradients/ragas/blob/c7302724a6203966c7d0c16db8f09ed5ce2a1a75/src/ragas/metrics/_answer_correctness.py#L32

⁸Thanks to Dr. Alexander Bondarenko for providing the prompt as well as the results of the evaluation for this metric.

5.2.2 Grid Search

In the first step of our grid search we measure the SemScore of each configuration. Figure 5.1 shows the distribution of SemScores for all tested configurations. We notice that there is a large variance in the achieved scores. The top five configurations all have an aggregated SemScore of 0.58 and higher. All top five configurations use the longest tested system prompt. In addition, all top runs use one-shot examples, but the specific example used varies. A single context snippet is used in three out of five runs. The summarization step is also used in three out of five runs. Although the long system prompt used for all candidates allows the system to reject an answer in the case of irrelevant or missing context documents, this option is not used for any generated answer - even if the top retrieved snippet was rated irrelevant in our retrieval evaluation.

We then evaluate the top five configurations using the OpenAI-based metrics. Figure 5.2 shows the results for the second stage of our automatic evaluation. While SemScore and Completeness are highly correlated, the correlation between Correctness and the other two metrics is lower, indicating that both LLM-based metrics seem to measure distinct aspects of the generated answers. We also determine the 95% confidence interval for each metric.⁹ This reveals a high variance for each tested score, making it hard to determine the best configuration based on the automatic evaluation metrics alone.

Based on these observations, we decide against selecting our generation configuration based on a single metric alone. Instead, we decide to examine the hyperparameters of the selected candidates. We determine the mode for each hyperparameter used by the candidates and notice that a system having each parameter equal to the mode is already featured in the top five configurations. We decide to select this configuration as the result of the grid search. The selected configuration uses a very extensive system prompt and a single passage as context that is passed through a summarization model. The chosen one-shot example is noticeable because the used context snippet only contains part of the reference answer to the question. We hypothesize that this might be beneficial because it encourages the model to use both the context and the embedded knowledge to generate the answer. The exact prompt the selected system uses is available in Appendix G.

5.2.3 Test Set Evaluation

Our grid search tested prompt configurations based on answers generated by the Mistral7B model. Since we treat the language model as a black box and do not perform any additional fine-tuning, we can easily replace the model used for

⁹For this we assume a normal distribution of scores.

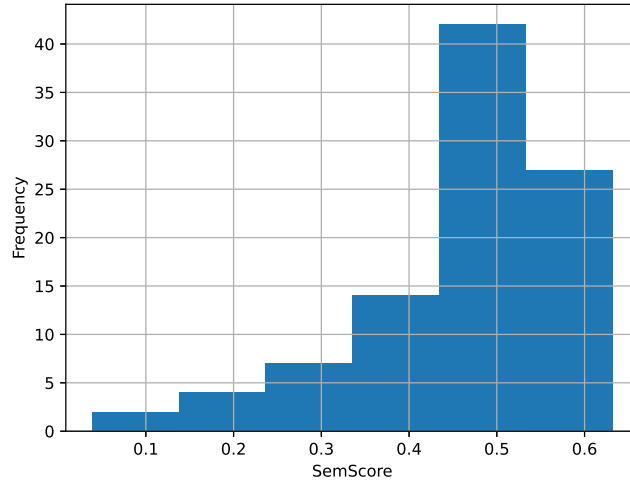


Figure 5.1: Distribution of SemScores for all tested configurations in the grid search. A system’s SemScore is the average of the SemScores of the generated answers to training questions.

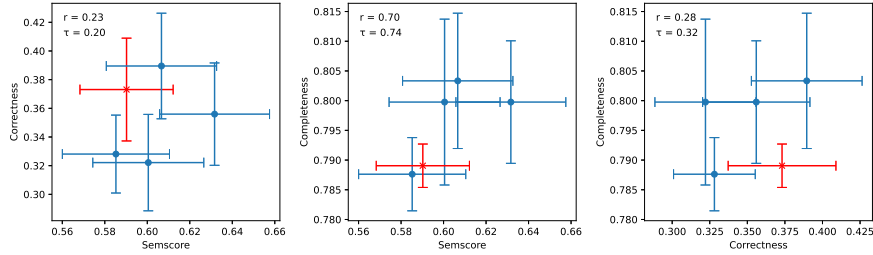


Figure 5.2: Pairwise comparison of all automatic evaluation metrics for the top five grid search configurations as decided by the SemScore metric. The additional bars show the 95% confidence intervals assuming a normal distribution. Our selected system is highlighted in red. The correlation of the aggregated metrics is measured using Pearson’s r and Kendall’s τ .

generation. This allows us to test the chosen configuration independently of the model used. We decide to evaluate the selected prompt configuration using not only the Mistral model, but also OpenAI’s significantly larger¹⁰ GPT4 model.¹¹ We use the same prompt for both models, only changing how the system prompt

¹⁰While OpenAI does not publish any information on the model’s size it is estimated to have about 1.7 trillion parameters: <https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/>. Mistral7B has 7 billion parameters.

¹¹We use OpenAI’s inference API with the model `gpt_4-0613`.

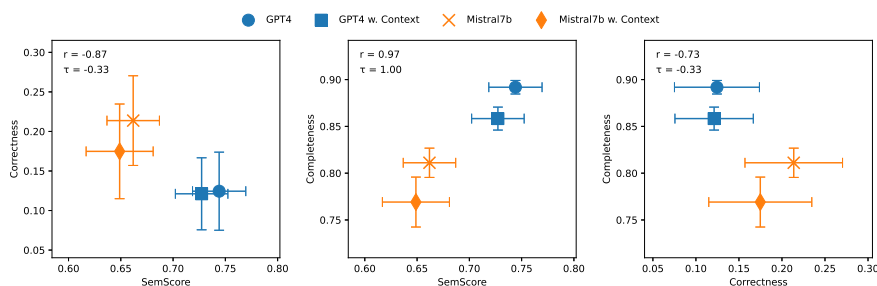


Figure 5.3: Pairwise comparison of our automatic evaluation metrics for the test set evaluation. The additional bars show the 95% confidence intervals assuming a normal distribution.

is passed. In addition, we generate answers using both models with a prompt that has no context and no examples. In these cases, the system prompt is updated to remove all references to context. The exact wording is available in Appendix H. This evaluation setting gives us four different generated answers and one reference answer per topic.

We first evaluate all generated answers using the same automatic metrics as in the grid search. We then have the generated answers evaluated by medical professionals and compare the generation results to MashQA’s reference answers.

Of all the responses generated, there is only one case where the system refuses to give an answer due to missing information in the passed passage. The pipeline in question uses GPT4. A manual review of the given context passage reveals that it is in fact not relevant to the given question, meaning that the model was correct in rejecting the answer. However, our manual relevance judgments found two other non-relevant passages that were provided as context, meaning that in some cases all of the tested systems failed to reject answers based on missing information in the context. While the one case of rejection is a desirable result, we decide to exclude the given answer from further evaluation to ensure comparability between the different pipelines.

Automatic Metrics

The pairwise distribution of our automatic evaluation metrics per system are shown in Figure 5.3. The value ranges for all scores for the pipeline using the Mistral model and retrieved context are similar to those measured in the grid search. Overall we notice that both GPT4-based systems outperform the Mistral-based systems regarding SemScore and Completeness. The Mistral-based systems outperform the GPT4-based systems regarding Correctness. For each model the configuration using no context outperforms its counterpart using context across all measured metrics. However, we find that regardless of whether

Table 5.1: Pearson correlation between non-aggregated automatic evaluation metrics across all systems for the test topics. No coefficient is significantly higher than 0 at the $p = 0.05$ level.

	SemScore	Correctness	Completeness
SemScore	-	0.05	0.17
Correctness	0.05	-	0.01
Completeness	0.17	0.01	-

context is used or not, the results are very similar if the base model used is the same. The chosen LLM seems to have a larger impact on the quality of the generated answers than the retrieval process.

We report the pairwise correlation between the non-aggregated automatic evaluation metrics in Table 5.1. These results show only very low positive correlations between the metrics. We perform t -tests to determine if the correlation coefficients are significantly different from 0. No measured correlation is significant at the $p = 0.05$ level. These results cannot be used to determine whether or not the metrics are capable to judge the quality of the generated answers. However, the lack of correlation shows us that if the metrics are usable to measure the generated answer’s quality, they do so in different ways. We aggregate all metrics using the mean per system. We then recalculate both the pearson as well as the kendall correlation between the aggregated metrics. The results are shown in Figure 5.3. While the SemScore and completeness metrics are highly correlated, RAGAs’ correctness metric does not correlate with the other two metrics. Additionally, it exhibits a very large variance.

Our automatic metrics show that the GPT4-based systems outperform the Mistral-based systems. Counterintuitively, the retrieval process has a negative impact on the quality of the generated answers. To get a better understanding of how the retrieval process influences the generation process, we investigate how the quality of the retrieved passages influences the quality of the generated answers. Table 5.2 shows how the topical relevance of the snippet that is used as context influences the metrics. We only calculate these metrics for runs that use context. While answers generated based on non-relevant passages perform lower across all metrics, there is no such trend between the finer-grained relevance classes. These results indicate that a good retrieval process may actually increase the quality of the generated answers. However, the results are only statistically significant for the Completeness metric.

We further investigate the influence of the context by measuring how similar the generated answers are to the given context. Given the overall high quality of our retrieval (as shown in the retrieval evaluation) we would expect that

Table 5.2: Automatic evaluation metrics of generated answers by the relevance of the supplied context. The dagger[†] indicates a significant (t-test with $p < 0.05$) increase in the metric compared to the corresponding lower relevance class. For the binarized relevance we count all partially relevant and relevant contexts as relevant.

Context Relevance	SemScore	Correctness	Completeness
Not Relevant	0.62	0.13	0.64
Partially Relevant	0.61	0.20	0.89 [†]
Relevant	0.71	0.14	0.82
<i>Relevant (Binarized)</i>	0.69	0.15	0.83 [†]

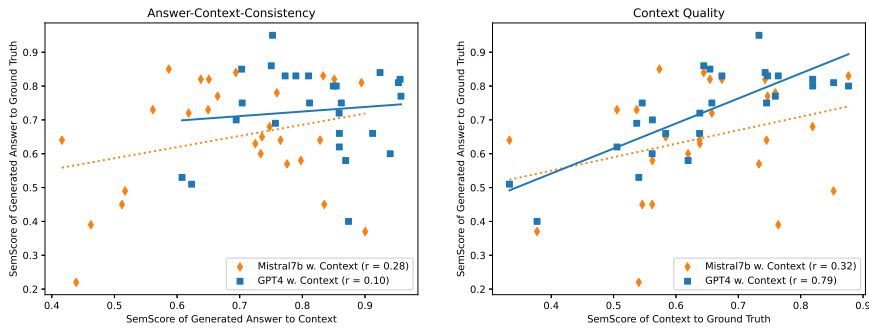


Figure 5.4: Influence of the contexts’ characteristics on the SemScore of the generated answers. The left plot shows the influence of the similarity between context and generated answer to the SemScore. The right plot shows the similarity between the context and the ground truth to the SemScore.

generated answers that are consistent with the retrieved context perform better than answers that are inconsistent with the context. We measure the similarity between the context and the generated answer using SemScore. The left plot in Figure 5.4 shows how the measured similarity influences the SemScore of the generated answers. With regards to the consistency between answer and context we do not notice an influence on the answer’s overall SemScore. It does not seem to matter whether or not the generated answer is similar to the given context as there is no influence on the answer’s SemScore. This is somewhat contradictory to our prior observation regarding the importance of the context for the generation process.

In our manual annotation of the retrieval results, we only evaluate the relevance of the retrieved passages with respect to the question - the alignment with the given reference answer is only measured as part of the correctness annotation. This means that even highly relevant documents may contain

different information from the reference answer. For this reason, we decide to measure an additional metric that measures the similarity between the retrieved context and the reference answer, acting as a *ex-post* measure of retrieval quality that checks whether the retrieved documents match our expected result or not. We measure this similarity using the SemScore metric. The second plot in Figure 5.4 shows the relationship between the *ex-post* context quality and the SemScore of the generated answers. The results for the Mistral-based system show a low correlation between the quality of the contexts and the SemScore of the generated answers. However, for the GPT4-based system, we observe a clear positive linear relationship between the two metrics. This means that the more similar the retrieved context is to the reference, the higher the SemScore of the generated answer. This indicates that the retrieval process has an impact on the generation process.

While the automatic evaluation metrics show that purely generative systems outperform systems that use retrieval for both tested models, the results of the finer-grained analysis show that the retrieval process can have a positive impact on the generation process. Our results seem to indicate that if we only retrieved documents that match our expected answers really well, the retrieval-augmented answers would be of higher quality than those generated without retrieval. This is a promising result as it indicates a potential for RAG to improve the quality of the generated answers in health-related QA.

However, this observation relies heavily on the fact that the automated metrics only compare the generated answers to the given reference answer. We have shown that contexts that align well with the reference answer can also lead to generated answers that align well with the reference answer. Note that this does not necessarily indicate that the generated answers are of a higher overall quality. Given the inherent ambiguity of non-factual questions and the possibility of multiple correct answers we discussed before, the generated answers might still be of higher quality than the reference answer if they do not align well with it. This underlines the need for a manual evaluation of the generated answers.

Manual Preference Annotation

We have our generated answers evaluated by medical professionals. We present the annotators with a question and a pair of two answers from different systems. The annotators are then asked to choose their preferred answer. While we provide some guidance on what to consider when choosing an answer, we do not give a strict definition of what a good answer is. Instead we ask the annotators to choose the answer that is closer to the answer they would give to their patients. The exact annotation guidelines are available in Appendix E.

Additionally, we give the annotators the option to choose "No Preference" if they feel that both answers are equally good or bad, and "Unsure" if they feel like they lack the expertise to make an informed decision. We use the open source tool Doccano to collect the annotations. One annotator has a doctorate and works as a trauma surgeon, the other annotator is a junior doctor in anaesthesia. As the questions in our test set are most likely to be assigned to general medicine, we do not make a specific assignment but randomly assign the same number of questions to each annotator for evaluation. For three questions, we let both annotators rate all answer pairs in order to measure the inter-coder agreement to test the reliability of our annotations.

We drop all annotations where the annotator chose "Unsure". Additionally we drop preference pairs where a context based system refuses to give an answer because of lacking information in the context. This is the case for one answer of the GPT4-based system. We calculate the agreement between both annotators using Fleiss' Kappa. The overall agreement is $\kappa = 0.14$ which is considered only slight agreement [Landis and Koch, 1977]. When ignoring any pairs where at least one annotator chose "No Preference" the agreement increases to $\kappa = 0.63$ which constitutes substantial agreement. This is likely due to the fact that the annotators differed greatly in their likelihood to choose "No Preference". One annotator chose that option in 38%, the other annotator used it only in 15% of all cases. Given that our chosen preference aggregation methods are capable to handle ties, we decide to include all annotations in our evaluation. We drop all shared pairs where there was no agreement between the annotators. Overall this leaves us with 225 expert annotations.

Table 5.4 shows the number of times each system was preferred over another, ignoring ties. The overall impression from the evaluation based on automatic metrics is confirmed: GPT4-based systems outperform Mistral-based systems, whether context is used or not. For GPT4, the system without RAG still wins the head-to-head comparison (7 wins, 6 losses, 10 ties). However, when comparing the Mistral systems, there is a slight preference for the system using context (9 wins, 7 losses, 6 ties). While the automatic metrics assume that the given reference answer is the best possible answer, we can abandon this assumption in the manual evaluation. In fact, the annotators frequently prefer the generated answers to the given reference answer. Table 5.3 shows the generated answers for a question where the annotators preferred every single generated answer to the reference answer. All answers, except for the one generated by Mistral7B without retrieval, are very careful in their wording and do not make any strong claims. This answer also stands out as its language seems less natural than the other answers. However, it is still preferred over the reference answer. The annotators did not note any preferences between the generated answers.

Table 5.3: Examples of generated answers for the test set evaluation. The reference answer is provided by MashQA. The generated answers are preferred by the annotators across all configurations. The medical professionals noted no preferences between the generated answers.

Question:	How effective is zinc for colds?
Reference Answer:	There are mixed reviews on zinc. Some studies show that zinc nasal sprays can help reduce how long your cold lasts and make your symptoms less severe. And because of the risk of a loss of smell, many experts recommend that you avoid zinc nasal sprays completely.
Mistral7B Context:	There have been many studies done regarding how well zinc works against common colds but they all seem to come back inconclusive as far as whether or not it actually helps fight off viruses like Rhinoviruses (which causes most cases). However one study did find evidence suggesting that taking high doses could reduce severity if taken within 24 hours after first feeling ill; however these findings were only seen among those who had never used antibiotics before so further research needs to take place before we make definitive claims about its effectiveness!
Mistral7B:	10% more likely to recover from cold within one week than those who took placebo. Zinc lozenges may reduce duration by about half day compared with placebos. However, it has been shown that taking too high dosage of zinc could cause nausea, vomiting, loss of taste and smell, headaches etc.
GPT4 Context:	Research on zinc’s effectiveness for colds has shown mixed results. Some studies suggest that zinc lozenges or syrup can reduce the length of a cold by one day, especially if taken within 24 hours of the first signs and symptoms. However, the research is not conclusive enough to recommend zinc as a standard treatment for colds.
GPT4:	Zinc may help to shorten the duration of a cold if taken within 24 hours of the first symptoms. However, it’s not a cure and its effectiveness can vary between individuals. Always consult with a healthcare provider before starting any new supplement regimen.

Table 5.4: Pairwise preferences for the test set evaluation. Each row shows the amount of times the system in the row was preferred over the system in the column. Annotations with "No Preference" are not included in this table. The table is ordered to maximize the row sums.

Preferred	Losing System				
	GPT4	GPT4 Context	Mistral7B Context	Reference Answer	Mistral7B
GPT4	-	7	10	17	14
GPT4 Context	6	-	12	16	12
Mistral7B Context	3	3	-	10	9
Reference Answer	1	2	11	-	10
Mistral7B	1	4	7	10	-

Table 5.5: Amount of times a system was tied with another system by the experts in the manual judgements.

	Ties				
	GPT4	GPT4 Context	Mistral7B Context	Reference Answer	Mistral7B
GPT4	-	10	5	4	7
GPT4 Context	10	-	7	5	9
Mistral7B Context	5	7	-	3	6
Reference Answer	4	5	3	-	4
Mistral7B	7	9	6	4	-

To get a better understanding of the annotators’ overall preferences we aggregate the pairwise preference metrics. The Borda Count (BC) assigns the differing systems points based on the amount of times they were preferred over another system. Ties are resolved by giving both systems half points [Emerson, 2013]. The Kemeny-Young method (KY) generates a ranking that maximizes the likelihood that a higher ranked system is preferred over a lower ranked system in a direct comparison [Kemeny, 1959]. While the Borda Count takes ties into account, they are not considered in the Kemeny-Young ranking. Additionally we report the overall win rate for each system across all pairwise preference pairs. The results are shown in Table 5.6. The aggregated preferences confirm our prior impressions. Both GPT4-based systems outperform the Mistral-based systems. The system using the GPT4 model without retrieval is the preferred system. MashQA’s reference answers are only preferable to

Table 5.6: Aggregated preference annotations for the test set evaluation. **WR** is the percentage of wins across all pairwise comparisons. **WR No Ties** is the win rate when ignoring ties. **BC** is the Borda Count. **KY** is the Kemeny-Young-Rank. We report the scores separately for the evaluation excluding the reference answer. For all metrics except the Kemeny-Young-Rank, higher scores indicate a better performance.

System	WR	WR No Ties	BC	KY
Including Reference Answer				
GPT4	0.53	0.81	63.5	1
GPT4 Context	0.52	0.74	59.0	2
Mistral7B Context	0.29	0.38	35.5	3
Reference Answer	0.26	0.31	32.0	4
Mistral7B	0.24	0.33	35.0	5
Excluding Reference Answer				
GPT4	0.46	0.76	44.0	1
GPT4 Context	0.46	0.68	41.0	2
Mistral7B Context	0.24	0.34	24.0	3
Mistral7B	0.17	0.26	23.0	4

the Mistral-based system without retrieval. While the retrieval has no positive impact for GPT4, it has a positive impact for the Mistral model. The ranking of the generation systems stays the same even when omitting any comparisons to the reference answers.

To study the influence of the used context, we examine how the generated answers perform against the ground truth in relation to the manual relevance judgement for the provided context. Given we only use a single passage as context, we can group each generated answer into three categories: The context is relevant to the question, the context is irrelevant to the question, and the context is partially relevant to the question. Given our system does not retrieve any passages with incorrect information for the test topics, we perform no evaluation with regards to the correctness of the context.

Table 5.7 shows the win rate of the generated answers over the ground truth based on the relevance of the context. As observed before, the generated answers are commonly preferred over the ground truth no matter the relevance of the context. Similar to what we already observed in the automatic evaluation, relevant contexts have a positive impact on the quality of the generated answers. This is the case for both partially and fully relevant context. The win rate is highest for partially relevant contexts. However, given the low count and

Table 5.7: Wins of generated answers over the ground truth based on the relevance of the context. For this we investigate all models together. Ties are not included in this table. For the binarized relevance we count all partially relevant and relevant contexts as relevant.

Context Relevance	Win Rate Generated	Count	σ^2
Not Relevant	0.68	9	0.29
Partially Relevant	0.81	8	0.22
Relevant	0.73	25	0.29
<i>Relevant (Binarized)</i>	0.75	33	0.28

high variance, we fail to find any significant differences between the different relevance classes.

Overall, the manual evaluation of the generation process partially confirms our prior observations. The large GPT4 model outperforms the smaller Mistral model - no matter if retrieval is used or not. While the purely generative GPT4-based system still ranks highest, its retrieval-augmented counterpart has very similar scores. For Mistral7B we even show a positive impact of the retrieval process.

Combining the Metrics

Overall, the automatic and manual evaluation yield similar results. This is interesting because the automatic evaluation metrics rely heavily on the reference answer, and we have shown that the generated answers are often preferred over the reference answer. To validate our results regarding the automatic evaluation metrics, we examine how our manual preference judgements and the automatic evaluation metrics align. Since we trust our manual preference annotations more than the automatic metrics, we assess the automatic metrics according to how closely they match the manual annotation.

For this, we take another look at the pairs annotated by medical experts. We only consider those pairs where two generated answers are compared. Additionally, we drop all pairs where the annotators chose "No Preference". For each pair, we then determine the difference in the aggregated automatic evaluation metrics. We get a value between -1 (the first answer has a score of 0 and the other answer has a score of 1) and 1 (the opposite is true). For a good metric, we would expect (large) positive values to be indicative of a preference for the first answer, and (large) negative values to be indicative of a preference for the second answer. We can then vary a threshold parameter

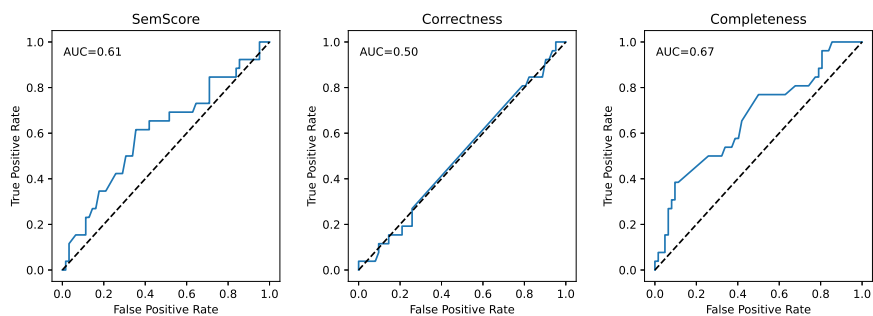


Figure 5.5: ROC curves for the preference prediction using the automatic evaluation metrics. The dashed line shows the performance of a random classifier.

Table 5.8: Pairwise correlation between aggregated evaluation metrics for the test set. Correlation is measured using Kendall’s τ . We report preference metrics as measured when not including the ground truth. Sem is short for SemScore, Corr for Correctness, and Comp for Completeness.

	Preference Metrics				Automatic Metrics		
	WR	WR No Ties	BC	KY	Sem.	Corr.	Comp.
WR	-	1.00	1.00	1.00	0.67	-0.67	0.67
WR No Ties	1.00	-	1.00	1.00	0.67	-0.67	0.67
BC	1.00	1.00	-	1.00	0.67	-0.67	0.67
KY	1.00	1.00	1.00	-	0.67	-0.67	0.67
SemScore	0.67	0.67	0.67	0.67	-	-0.33	1.00
Correctness	-0.67	-0.67	-0.67	-0.67	-0.33	-	-0.33
Completeness	0.67	0.67	0.67	0.67	1.00	-0.33	-

to determine how well the automatic evaluation metrics predict the manual preference. We determine the receiver operating characteristic curve (ROC) for predicting the preference using the difference in scores. The results can be seen in Figure 5.5. For the Correctness metric, the area under the curve (AUC) is 0.5, indicating that it is not possible to predict manual preference using this metric alone. The AUC scores for the Completeness metric (0.67) and SemScore (0.61) are close to another. However, our overall impression is that the automatic evaluation metrics alone are not perfect predictors of answer quality.

The pairwise rank correlation between the aggregated preference metrics and the automatic evaluation metrics can be seen in Table 5.8. SemScore and Completeness show a good correlation of $\tau = 0.67$ to each of our preference

Table 5.9: Automatic evaluation metrics for our generative models when using the context provided by MashQA compared to our prior runs.

System	SemScore	Completeness	Correctness
GPT4 MashQA	0.80	0.91	0.14
GPT4 Retrieved	0.73	0.86	0.12
GPT4	0.74	0.89	0.12

metrics. The correctness metric has a negative correlation of $\tau = -0.67$ to the preference metrics. This is in line with our prior observations that the correctness metric is unable to predict the expert preferences. Even though the other automatic metrics have the weakness of using a potentially flawed reference answer, they still show a good correlation with the expert preferences. While our results underline the importance of manual evaluation, they also show that automatic evaluation metrics can be used to get a rough estimate of the quality of the generated answers. Furthermore, these findings support the results of our grid search, that mainly relied on the SemScore metric.

5.2.4 Ablation: Generation using MashQA’s context

We have so far largely ignored one feature of MashQA - the dataset we base our work on. Given its original use to evaluate the performance of extractive QA models, it supplies a context document for each question. In our work so far we have opted to use a retrieval system to find relevant context for our questions. In this section we evaluate the answer quality of our generative model when using the context provided by MashQA.

For this evaluation we opt to use the best generative configuration as determined in the previous section. We thus generate our answers using the GPT4 model with OpenAI’s inference API. Zhu et al. [2020] report their results for the dataset using a sentence-level F_1 -score as well as the percentage of exact matches (EM) between extracted answer and the ground truth. MultiCo, the extractive approach introduced by Zhu et al. [2020] is the only system evaluated on the dataset. It reaches an F_1 -score of 0.65 and an exact match rate of 0.29. We generate new answers using the provided context and calculate both the sentence-level, as well as our automatic evaluation metrics.

When evaluating our generation results using the F_1 and EM-metrics, we get scores of 0 for all tested configurations. This is somewhat expected, given the pitfalls of evaluating generative models as discussed in the literature review. This makes a comparison of our approach to that of Zhu et al. [2020] difficult. Because there are no published answers created by MultiCo, we are unable to

evaluate this approach using our metrics. Table 5.9 shows how the automated metrics are changed when using MashQA’s context instead of the retrieved context or none at all. For all other metrics, the usage of the provided context results in an improvement. These differences are not significant, however this might be due to the small sample size of only 27 topics.

These findings underline our prior observations. Given the reference answers are extracted spans of the given context documents, the provided context matches the reference answers perfectly. This results in generation results that are very similar to the reference answers, and thus have high automatic evaluation scores.

5.3 Summary

In this chapter we described our construction of the generative step for our RAG system. We treated the language model as a black box, focusing solely on optimizing all other aspects such as system prompt, context preprocessing and few-shot examples. This allowed us to easily switch between different LLMs and evaluate the retrieval effectiveness of our system using different models. We performed both an automatic and a manual evaluation of the generated answers. We find that a system using OpenAI’s GPT4 model without any retrieval component outperforms all tested RAG configurations. Both the automatic and manual evaluation yield similar rankings, with the exception that the manual evaluation indicates a positive impact of retrieval for the Mistral7B model. We also find, that relevant context snippets can lead to better answers, as measured both by manual and automatic evaluation. This is especially true if the retrieved snippets align well with the expected answer.

Furthermore we find that both the SemScore and Completeness metric can be used to evaluate the quality of the generated answers. However, we still note that the metrics’ dependence on potentially imperfect reference answers can lead to misleading results.

Chapter 6

Conclusion

In this thesis, we worked on answering non-factual questions for health-related consumer questions. We introduced a retrieval-augmented generation (RAG) system for this task. In this chapter, we summarize our findings, possible pitfalls, and the potentials for future work.

6.1 Findings and Contributions

Our work in this thesis was heavily oriented on issues specific to the task of (health-related) question answering. In our literature review, we found that there is still a lack of datasets available for the task of non-factual health-related question answering [Mutabazi et al., 2021]. Furthermore, the question is how to best preprocess questions for the retrieval step in QA systems is still an open issue. Lastly, generative approaches for QA come with the issue of automatically evaluating the generated answers. We addressed these issues in our work and made the following contributions.

Dataset: We built upon the existing MashQA [Zhu et al., 2020] dataset to create a new dataset called MashTREC for health-related consumer questions. This dataset combines the original MashQA dataset with narratives taken from prior retrieval tasks. This enables us to better evaluate retrieval systems for non-factual health-related consumer questions. In total our dataset consists of 50 training topics, and 27 test topics with additional narratives.

Preprocessing: We experimented with two distinct approaches to turn natural language questions into keyword queries. Our first approach works purely extractive and uses a term classification model to extract keywords from the question. We train a BERT-based model that outperforms the current state-of-the-art on the Essential Tasks dataset. Our second approach is a

generative model that uses a T5-based sequence-to-sequence model to generate keyword queries from questions. This enables us to introduce new - potentially more complex - keywords into the query. Furthermore, we create a dataset that combines consumer questions with keyword queries in a semi-supervised manner. However, our evaluations show that both of our preprocessing approaches are unable to significantly improve the retrieval effectiveness. This is the case for the yes-no questions taken from the TREC Health Misinformation Track, as well as for the non-factual questions from MashTREC.

Retrieval: We conducted a grid search, combining existing (re-)rankers to optimize the retrieval effectiveness for non-factual questions. For this, we used a manual annotation process that produced 3,439 judgements for the training set and 135 judgements for the test set. These judgements include annotations for both the relevance as well as factual correctness of the retrieved snippets. We produced a baseline score for the retrieval on our newly introduced MashTREC dataset and performed an evaluation on the TREC Health Misinformation Track dataset.

Generation: Treating the language model as a black box, we optimized the prompts for a retrieval-augmented generation system. Our selected prompt only features a single snippet as context and uses a summarizer to remove noise from the retrieved sources. Furthermore, only a single one-shot example seems to be necessary to guide the model towards generating good answers.

We evaluate our RAG system using both Mistral7B and GPT4 as the generative model. Both the automatic metrics, as well as the human evaluation show that GPT4-based systems outperform the systems using the smaller language model. We find that the retrieval step does not improve the answer quality of GPT4. For Mistral7B, the manual evaluation shows that the retrieval step can improve the answer quality. However, the automatic metrics do not fully reflect this improvement.

We find evidence that the retrieval step can help to better align the generated answers with the reference answers if the retrieved context matches the reference answer. In the context of non-factual QA, this comes with the issue that there are usually multiple correct answers to a question and a generated answer can well be of good quality even if it does not align with the reference answer. This is illustrated by the fact that medical experts commonly preferred generated answers over the reference answers.

Evaluation: Because we conducted both an automatic as well as a manual evaluation for the generation results on our test topics, we were able to compare the results of both evaluations. Treating the expert preferences as the ground truth, we found that the automatic evaluation metrics were only partially able

to reflect the quality of the generated answers. The Correctness metric by RAGAs was not able to reflect the quality of the generated answers as indicated by the expert evaluation. While far from perfect predictors, both SemScore as well as the Completeness metric are usable to estimate the overall quality of the generated answers.

6.2 Pitfalls

Our work is not without its pitfalls. We source our questions from MashQA, which consists of questions scraped from WebMD. It is likely that both of the evaluated language models have seen our test topics during training. Furthermore, due to our usage of Mono- and DuoT5, there is the possibility of train-test-leakage in our retrieval step [Fröbe et al., 2022]. To mitigate this issue, we should use an additional test dataset that is guaranteed to consist of new and unseen consumer question and answer pairs.

During the evaluation of our retrieval system, we found that the retrieval effectiveness on the test dataset significantly increases compared to the training dataset. While this is likely due to the fact that the test dataset slightly differs from the training dataset, it could also be an indicator of mistakes in our annotations or the calculation of the evaluation metrics. While we found no indication of this, it is still a possibility that the baseline score on MashTREC is not fully reliable. The last point to mention regards the evaluation of our RAG system. While we found some evidence that retrieving relevant snippets can improve the answer quality, we failed to make many statistically significant observations. This is likely due to the relatively small size of our test set that consists of only 27 topics.

Furthermore, we did not take possible biases in the language models we used into account. GPT4 has shown to exhibit both racial and gender biases in generated text on medical topics [Zack et al., 2024]. For further work, it would be advisable to evaluate the biases in the language models we use and potentially mitigate them.

6.3 Future Work

Our work introduced multiple new datasets that could be used for future research. Especially the evaluation of our retrieval system on the TREC Health Misinformation Track dataset shows that there is still a lot of potential for improvement. One possible future direction could be to use the judgements from our manual annotation process to train a learning-to-rank model. This could potentially improve the retrieval performance for health-related questions.

We treated the language model that we used to generate answers as a black box. No additional fine-tuning was performed on the model. In particular, this also means that the generative step is not specific to health-related issues. One possible future direction could be to evaluate how our system performs on domains other than the medical domain. Furthermore, we could also go into the opposite direction and fine-tune the language model on health-related data. Our best-performing systems use the large-scale GPT4-model. While the specific numbers are not known, we know that both the training as well as inference of models of these sizes come with a significant carbon footprint [Chien et al., 2023]. Fine-tuning a smaller model for the specific task of health-related RAG could potentially reduce the carbon footprint of our system while still maintaining a high performance [Fu et al., 2024].

Bibliography

- Asma Ben Abacha and Dina Demner-Fushman. A Question-Entailment Approach to Question Answering. *BMC Bioinform.*, 20(1):511:1–511:23, 2019. doi: 10.1186/S12859-019-3119-4.
- Asma Ben Abacha, Eugene Agichtein, Yuval Pinter, and Dina Demner-Fushman. Overview of the Medical Question Answering Task at TREC 2017 LiveQA. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, volume 500–324 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2017.
- Asma Ben Abacha, Yassine Mrabet, Mark Sharp, Travis R. Goodwin, Sonya E. Shooshan, and Dina Demner-Fushman. Bridging the Gap Between Consumers’ Medication Questions and Trusted Answers. In Lucila Ohno-Machado and Brigitte Séroussi, editors, *MEDINFO 2019: Health and Wellbeing e-Networks for All - Proceedings of the 17th World Congress on Medical and Health Informatics, Lyon, France, 25-30 August 2019*, volume 264 of *Studies in Health Technology and Informatics*, pages 25–29. IOS Press, 2019a. doi: 10.3233/SHTI190176.
- Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. Overview of the MediQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 370–379. Association for Computational Linguistics, 2019b. doi: 10.18653/V1/W19-5039.
- Jafar A. Alzubi, Rachna Jain, Anubhav Singh, Pritee Parwekar, and Meenu Gupta. COBERT: COVID-19 Question Answering System Using BERT. *Arabian Journal for Science and Engineering*, June 2021. ISSN 2193-567X, 2191-4281. doi: 10.1007/s13369-021-05810-5.

- John W. Ayers, Adam Poliak, Mark Dredze, Eric C. Leas, Zechariah Zhu, Jessica B. Kelley, Dennis J. Faix, Aaron M. Goodman, Christopher A. Longhurst, Michael Hogarth, and Davey M. Smith. Comparing Physician and Artificial Intelligence Chatbot Responses to Patient Questions Posted to a Public Social Media Forum. *JAMA Internal Medicine*, 183(6):589, June 2023. ISSN 2168-6106. doi: 10.1001/jamainternmed.2023.1838.
- Ansar Aynetdinov and Alan Akbik. SemScore: Automated Evaluation of Instruction-Tuned LLMs Based on Semantic Textual Similarity. Technical Report arXiv:2401.17072, 2024.
- Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare R. Voss, editors, *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics, 2005.
- Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O’Reilly, Beijing, 2009. ISBN 978-0-596-51649-9. doi: <http://my.safaribooksonline.com/9780596516499>.
- Alexander Bondarenko, Maik Fröbe, Lukas Gienapp, Alexander Pugachev, Jan Heinrich Reimer, Ferdinand Schlatt, Ekaterina Artemova, Martin Potthast, Benno Stein, Pavel Braslavski, and Matthias Hagen. Webis at TREC 2022: Deep Learning and Health Misinformation. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, Online, November 15-19, 2022*, volume 500–338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022.
- Célia Boyer, Mark Selby, Jean-Raoul Scherrer, and Ron D. Appel. The Health on the Net Code of Conduct for Medical and Health Websites. 28(5):603–610, 1998. doi: 10.1016/S0010-4825(98)00037-7.
- Manvi Breja and Sanjay Kumar Jain. A Survey on Non-Factoid Question Answering Systems. *International Journal of Computers and Applications*, 44(9):830–837, 2022. doi: 10.1080/1206212X.2021.1949117.
- 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. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*, 2020.
- Berkant Barla Cambazoglu, Valeria Bolotova-Baranova, Falk Scholer, Mark Sanderson, Leila Tavakoli, and W. Bruce Croft. Quantifying Human-Perceived Answer Utility in Non-Factoid Question Answering. In Falk Scholer, Paul Thomas, David Elsweiler, Hideo Joho, Noriko Kando, and Catherine Smith, editors, *CHIIR ’21: ACM SIGIR Conference on Human Information Interaction and Retrieval, Canberra, ACT, Australia, March 14-19, 2021*, pages 75–84. ACM, 2021. doi: 10.1145/3406522.3446028.
- Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Evaluating Question Answering Evaluation. In Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen, editors, *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 119–124. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-5817.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking Large Language Models in Retrieval-Augmented Generation. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17754–17762. AAAI Press, 2024. doi: 10.1609/AAAI.V38I16.29728.
- Andrew A. Chien, Liuzixuan Lin, Hai Nguyen, Varsha Rao, Tristan Sharma, and Rajini Wijayawardana. Reducing the carbon impact of generative AI inference

- (today and in 2035). In George Porter, Tom Anderson, Andrew A. Chien, Tamar Eilam, Colleen Josephson, and Jonggyu Park, editors, *Proceedings of the 2nd Workshop on Sustainable Computer Systems, HotCarbon 2023, Boston, MA, USA, 9 July 2023*, pages 11:1–11:7. ACM, 2023. doi: 10.1145/3604930.3605705.
- Joanna Tw Chu, Man Ping Wang, Chen Shen, Kasisomayajula Viswanath, Tai Hing Lam, and Sophia Siu Chee Chan. How, When and Why People Seek Health Information Online: Qualitative Study in Hong Kong. *Interactive Journal of Medical Research*, 6(2):e24, December 2017. ISSN 1929-073X. doi: 10.2196/ijmr.7000.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1300.
- Charles Clarke, Maria Maistro, and Mark Smucker. Overview of the TREC 2021 Health Misinformation Track. In *TREC*, 2021.
- Charles L. A. Clarke, Saira Rizvi, Mark D. Smucker, Maria Maistro, and Guido Zuccon. Overview of the TREC 2020 Health Misinformation Track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020a.
- Charles L. A. Clarke, Alexandra Vtyurina, and Mark D. Smucker. Offline Evaluation Without Gain. In Krisztian Balog, Vinay Setty, Christina Lioma, Yiqun Liu, Min Zhang, and Klaus Berberich, editors, *ICTIR '20: The 2020 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Norway, September 14-17, 2020*, pages 185–192. ACM, 2020b. doi: 10.1145/3409256.3409816.
- Charles L. A. Clarke, Maria Maistro, Mahsa Seifkar, and Mark D. Smucker. Overview of the TREC 2022 Health Misinformation Track (Notebook). Notebook, 2022.

- Daniel Cohen, Liu Yang, and W. Bruce Croft. WikiPassageQA: A Benchmark Collection for Research on Non-Factoid Answer Passage Retrieval. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1165–1168. ACM, 2018. doi: 10.1145/3209978.3210118.
- Eduardo Gabriel Cortes, Vinicius Woloszyn, Dante Barone, Sebastian Möller, and Renata Vieira. A Systematic Review of Question Answering Systems for Non-Factoid Questions. *Journal of Intelligent Information Systems*, 58(3):453–480, 2022. doi: 10.1007/S10844-021-00655-8.
- Dina Demner-Fushman, Yassine Mrabet, and Asma Ben Abacha. Consumer Health Information and Question Answering: Helping Consumers Find Answers to Their Health-Related Information Needs. *J. Am. Medical Informatics Assoc.*, 27(2):194–201, 2020. doi: 10.1093/JAMIA/OCZ152.
- 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423.
- Peter Emerson. The Original Borda Count and Partial Voting. *Social Choice and Welfare*, 40(2):353–358, 2013. doi: 10.1007/S00355-011-0603-9.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated Evaluation of Retrieval Augmented Generation. Technical Report arXiv:2309.15217, 2023.
- Eurostat. EU Citizens: Over Half Seek Health Information Online. <https://ec.europa.eu/eurostat/web/products-eurostat-news/-/edn-20220406-1>, 2022.
- Maik Fröbe, Christopher Akiki, Martin Potthast, and Matthias Hagen. How train-test leakage affects zero-shot retrieval. In Diego Arroyuelo and Barbara Poblete, editors, *String Processing and Information Retrieval - 29th International Symposium, SPIRE 2022, Concepción, Chile, November 8-10, 2022, Proceedings*, volume 13617 of *Lecture Notes in Computer Science*, pages 147–161. Springer, 2022. doi: 10.1007/978-3-031-20643-6_11.

- Xue-Yong Fu, Md. Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Bhushan TN. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization? Technical report, 2024.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey. Technical Report arXiv:2312.10997, 2023.
- Zhen Guo, Peiqi Wang, Yanwei Wang, and Shangdi Yu. Improving Small Language Models on PubMedQA via Generative Data Augmentation. Technical Report arXiv:2305.07804, arXiv, 2023.
- S. Harvey, A. Memon, R. Khan, and F. Yasin. Parent’s Use of the Internet in the Search for Healthcare Information and Subsequent Impact on the Doctor–Patient Relationship. *Irish Journal of Medical Science (1971 -)*, 186(4):821–826, November 2017. ISSN 0021-1265, 1863-4362. doi: 10.1007/s11845-017-1555-6.
- Hana L Haver, Emily Ambinder, Manisha Bahl, Eniola T Oluyemi, Jean Jeudy, and Paul H Yi. Appropriateness of Breast Cancer Prevention and Screening Recommendations Provided by Chatgpt. *Radiology*, 307(4):e230424, 2023.
- Sebastian Hofstätter, Sophia Althammer, Mete Sertkan, and Allan Hanbury. Establishing Strong Baselines For TripClick Health Retrieval. In Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørkvåg, and Vinay Setty, editors, *Advances in Information Retrieval*, volume 13186, pages 144–152. Springer International Publishing, Cham, 2022. ISBN 978-3-030-99738-0 978-3-030-99739-7. doi: 10.1007/978-3-030-99739-7_17.
- Andrea Iovine, Anjie Fang, Besnik Fetahu, Jie Zhao, Oleg Rokhlenko, and Shervin Malmasi. CycleKQR: Unsupervised Bidirectional Keyword-Question Rewriting. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11875–11886, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.814.
- Gautier Izacard and Edouard Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for*

- Computational Linguistics: Main Volume*, pages 874–880, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74.
- Kalervo Järvelin and Jaana Kekäläinen. IR Evaluation Methods for Retrieving Highly Relevant Documents. In Emmanuel J. Yannakoudakis, Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors, *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 24-28, 2000, Athens, Greece*, pages 41–48. ACM, 2000. doi: 10.1145/345508.345545.
- Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. Technical Report arXiv:2310.06825, 2023a.
- 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, May 2023b.
- Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Huaiyuan Ying, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. Biomedical Question Answering: A Survey of Approaches and Challenges. *ACM Comput. Surv.*, 55(2):35:1–35:36, 2023. doi: 10.1145/3490238.
- Shaheen Kanthawala, Amber Vermeesch, Barbara Given, and Jina Huh. Answers to Health Questions: Internet Search Results Versus Online Health Community Responses. *Journal of Medical Internet Research*, 18(4):e95, April 2016. ISSN 1438-8871. doi: 10.2196/jmir.5369.
- 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, September 2020.
- John G. Kemeny. Mathematics Without Numbers. *Daedalus*, 88(4):577–591, 1959. ISSN 00115266.

- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. Learning What Is Essential in Questions. In Roger Levy and Lucia Specia, editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 80–89, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1010.
- Khresmoi. Khresmoi Manually Annotated Reference Corpus, November 2016.
- Ryul Kim, Han-Joon Kim, and Beomseok Jeon. The Good, the Bad, and the Ugly of Medical Information on the Internet: Medical Information on The Internet. *Movement Disorders*, 33(5):754–757, May 2018. ISSN 08853185. doi: 10.1002/mds.27324.
- Tiffany H. Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, and Victor Tseng. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLOS Digital Health*, 2(2):e0000198, February 2023. ISSN 2767-3170. doi: 10.1371/journal.pdig.0000198.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. Association for Computational Linguistics, 2017. doi: 10.18653/V1/D17-1082.
- J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics. Journal of the International Biometric Society*, 33(1):159–174, 1977. ISSN 0006341X, 15410420.
- Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. Internet-Augmented Language Models Through Few-Shot Prompting for Open-Domain Question Answering. Technical Report arXiv:2203.05115, 2022.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinform.*, 36(4):1234–1240, 2020. doi: 10.1093/BIOINFORMATICS/BTZ682.
- Patrick S. H. 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. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*, 2020.
- Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. Technical Report arXiv:2303.14070, arXiv, 2023.
- Chin-Yew Lin and Franz Josef Och. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain, July 2004. doi: 10.3115/1218955.1219032.
- Jimmy Lin. The Neural Hype and Comparisons Against Weak Baselines. *SIGIR Forum*, 52(2):40–51, 2018. doi: 10.1145/3308774.3308781.
- Feifan Liu, Lamont D. Antieau, and Hong Yu. Toward Automated Consumer Question Answering: Automatically Separating Consumer Questions from Professional Questions in the Healthcare Domain. *Journal of Biomedical Informatics*, 44(6):1032–1038, December 2011. ISSN 15320464. doi: 10.1016/j.jbi.2011.08.008.
- Alejandro Lozano, Scott L. Fleming, Chia-Chun Chiang, and Nigam Shah. Clinfo.ai: An Open-Source Retrieval-Augmented Large Language Model System for Answering Medical Questions Using Scientific Literature. Technical Report arXiv:2310.16146, 2023.
- Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. Simplified Data Wrangling with Irdatasets. In *SIGIR*, 2021.
- Craig Macdonald and Nicola Tonellotto. Declarative Experimentation Ininformation Retrieval Using Pyterrier. In *Proceedings of ICTIR 2020*, 2020.
- N. Miller, E. M. Lacroix, and J. E. Backus. MEDLINEplus: Building and Maintaining the National Library of Medicine's Consumer Health Web Service. *Bulletin of the Medical Library Association*, 88(1):11–17, January 2000. ISSN 0025-7338.

- Ryan Musa, Xiaoyan Wang, Achille Fokoue, Nicholas Mattei, Maria Chang, Pavan Kapanipathi, Bassem Makni, Kartik Talamadupula, and Michael Witbrock. Answering Science Exam Questions Using Query Reformulation with Background Knowledge. In *Automated Knowledge Base Construction (AKBC)*, 2018.
- Emmanuel Mutabazi, Jianjun Ni, Guangyi Tang, and Weidong Cao. A Review on Medical Textual Question Answering Systems Based on Deep Learning Approaches. *Applied Sciences*, 11(12):5456, June 2021. ISSN 2076-3417. doi: 10.3390/app11125456.
- Khalid Nassiri and Moulay Akhloufi. Transformer Models Used for Text-Based Question Answering Systems. *Applied Intelligence*, 53(9):10602–10635, 2023. doi: 10.1007/S10489-022-04052-8.
- Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, and Georgios Paliouras. Overview of BioAsq Tasks 11b and Synergy11 in CLEF 2023. In Mohammad Aliannejadi, Guglielmo Faggioli, Nicola Ferro, and Michalis Vlachos, editors, *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023)*, Thessaloniki, Greece, September 18th to 21st, 2023, volume 3497 of *CEUR Workshop Proceedings*, pages 19–26. CEUR-WS.org, 2023.
- Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. Learning to Attend on Essential Terms: An Enhanced Retriever-Reader Model for Open-Domain Question Answering. 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 335–344, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1030.
- Arnold Overwijk, Chenyan Xiong, and Jamie Callan. ClueWeb22: 10 Billion Web Documents with Rich Information. In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3360–3362. ACM, 2022. doi: 10.1145/3477495.3536321.
- João R. M. Palotti, Guido Zuccon, Lorraine Goeuriot, Liadh Kelly, Allan Hanbury, Gareth J. F. Jones, Mihai Lupu, and Pavel Pecina. CLEF eHealth Evaluation Lab 2015, Task 2: Retrieving Information About Medical Symptoms. In Linda Cappellato, Nicola Ferro, Gareth J. F. Jones, and Eric

- SanJuan, editors, *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation Forum, Toulouse, France, September 8-11, 2015*, volume 1391 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135.
- Ronak Pradeep, Rodrigo Frassetto Nogueira, and Jimmy Lin. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. Technical Report arXiv:2101.05667, 2021.
- Alexander Pugachev, Ekaterina Artemova, Alexander Bondarenko, and Pavel Braslavski. Consumer Health Question Answering Using Off-the-Shelf Components. In Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo, editors, *Advances in Information Retrieval*, volume 13981, pages 571–579. Springer Nature Switzerland, Cham, 2023. ISBN 978-3-031-28237-9 978-3-031-28238-6. doi: 10.1007/978-3-031-28238-6_48.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1264.
- Arya Rao, Michael Pang, John Kim, Meghana Kamineni, Winston Lie, Anoop K. Prasad, Adam Landman, Keith J Dreyer, and Marc D. Succi. Assessing the Utility of ChatGPT Throughout the Entire Clinical Workflow. Preprint, Health Informatics, 2023.
- Navid Rekasaz, Oleg Lesota, Markus Schedl, Jon Brassey, and Carsten Eickhoff. TripClick: The Log Files of a Large Health Web Search Engine. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2507–2513, 2021. doi: 10.1145/3404835.3463242.

- Adam Roberts, Colin Raffel, and Noam Shazeer. How Much Knowledge Can You Pack into the Parameters of a Language Model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.437.
- Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, Steven Bedrick, and William R. Hersh. Overview of the TREC 2022 Clinical Trials Track. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, Online, November 15-19, 2022*, volume 500–338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022.
- Anna Rogers, Matt Gardner, and Isabelle Augenstein. QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension. *ACM Comput. Surv.*, 55(10):197:1–197:45, 2023. doi: 10.1145/3560260.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-Augmented Black-Box Language Models. Technical Report arXiv:2301.12652, arXiv, 2023.
- Shrutikirti Singh and Seba Susan. Healthcare Question–Answering System: Trends and Perspectives. In Sarika Jain, Sven Groppe, and Nandana Mihindukulasooriya, editors, *Proceedings of the International Health Informatics Conference*, volume 990, pages 239–249. Springer Nature Singapore, Singapore, 2023. ISBN 978-981-19908-9-2 978-981-19909-0-8. doi: 10.1007/978-981-19-9090-8_21.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera Y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large Language Models Encode Clinical Knowledge. *Nature*, July 2023a. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-023-06291-2.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike

- Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. Towards Expert-Level Medical Question Answering with Large Language Models. Technical Report arXiv:2305.09617, arXiv, 2023b.
- Ian Soboroff. Overview of TREC 2021. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021, Online, November 15-19, 2021*, volume 500–335 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2021.
- Hanna Suominen, Lorraine Goeuriot, Liadh Kelly, Laura Alonso Alemany, Elias Bassani, Nicola Brew-Sam, Viviana Cotik, Darío Filippo, Gabriela González Sáez, Franco Luque, Philippe Mulhem, Gabriella Pasi, Roland Roller, Sandaru Seneviratne, Rishabh Upadhyay, Jorge Vivaldi, Marco Viviani, and Chenchen Xu. Overview of the CLEF eHealth Evaluation Lab 2021. In K. Selçuk Candan, Bogdan Ionescu, Lorraine Goeuriot, Birger Larsen, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21-24, 2021, Proceedings*, volume 12880 of *Lecture Notes in Computer Science*, pages 308–323. Springer, 2021. doi: 10.1007/978-3-030-85251-1_21.
- Lynda Tamine and Lorraine Goeuriot. Semantic Information Retrieval on Medical Texts: Research Challenges, Survey, and Open Issues. *ACM Comput. Surv.*, 54(7):146:1–146:38, 2022. doi: 10.1145/3462476.
- Arun James Thirunavukarasu, Refaat Hassan, Shathar Mahmood, Rohan Sanghera, Kara Barzangi, Mohammed El Mukashfi, and Sachin Shah. Trialling a Large Language Model (ChatGPT) in General Practice With the Applied Knowledge Test: Observational Study Demonstrating Opportunities and Limitations in Primary Care. *JMIR Medical Education*, 9:e46599, April 2023a. ISSN 2369-3762. doi: 10.2196/46599.
- Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large Language Models in Medicine. *Nature Medicine*, 29(8):1930–1940, August 2023b. ISSN 1078-8956, 1546-170X. doi: 10.1038/s41591-023-02448-8.
- Rishabh Upadhyay, Gabriella Pasi, and Marco Viviani. An Unsupervised Approach to Genuine Health Information Retrieval Based on Scientific

- Evidence. In Richard Chbeir, Helen Huang, Fabrizio Silvestri, Yannis Manolopoulos, and Yanchun Zhang, editors, *Web Information Systems Engineering – WISE 2022*, volume 13724, pages 119–135. Springer International Publishing, Cham, 2022. ISBN 978-3-031-20890-4 978-3-031-20891-1. doi: 10.1007/978-3-031-20891-1_10.
- Liesbet Van Bulck and Philip Moons. What If Your Patient Switches from Dr. Google to Dr. Chatgpt? A Vignette-Based Survey of the Trustworthiness, Value, and Danger of Chatgpt-Generated Responses to Health Questions. *European Journal of Cardiovascular Nursing*, page zvad038, April 2023. ISSN 1474-5151, 1873-1953. doi: 10.1093/eurjcn/zvad038.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.
- Harriet Louise Walker, Shahi Ghani, Christoph Kuemmerli, Christian Andreas Nebiker, Beat Peter Müller, Dimitri Aristotle Raptis, and Sebastian Manuel Staubli. Reliability of Medical Information Provided by ChatGPT: Assessment Against Clinical Guidelines and Patient Information Quality Instrument. *Journal of Medical Internet Research*, 25:e47479, June 2023. ISSN 1438-8871. doi: 10.2196/47479.
- Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models Are Not Fair Evaluators. Technical Report arXiv: 2305.17926, 2023.
- William Webber, Alistair Moffat, and Justin Zobel. A Similarity Measure for Indefinite Rankings. *ACM Trans. Inf. Syst.*, 28(4):20:1–20:38, 2010. doi: 10.1145/1852102.1852106.
- Anuradha Welivita and Pearl Pu. A Survey of Consumer Health Question Answering Systems. *AI Mag.*, 44(4):482–507, 2023. doi: 10.1002/AAAI.12140.
- Ryen W. White, Matthew Richardson, and Wen-tau Yih. Questions vs. Queries in Informational Search Tasks. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 135–136. ACM, 2015. doi: 10.1145/2740908.2742769.

- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. PMC-LLAMA: Towards Building Open-Source Language Models for Medicine. Technical Report arXiv:2304.14454, 2023.
- Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking Retrieval-Augmented Generation for Medicine. Technical Report arXiv:2402.13178, 2024.
- 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. Technical Report arXiv:2007.00808, 2020.
- Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. A Critical Evaluation of Evaluations for Long-form Question Answering. Technical Report arXiv:2305.18201, arXiv, 2023a.
- Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval Meets Long Context Large Language Models. Technical Report arXiv:2310.03025, 2023b.
- Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. Prca: Fitting Black-Box Large Language Models for Retrieval Question Answering Via Pluggable Reward-Driven Contextual Adapter. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5364–5375. Association for Computational Linguistics, 2023.
- Travis Zack, Eric Lehman, Mirac Suzgun, Jorge A. Rodriguez, Leo Anthony Celi, Judy Gichoya, Dan Jurafsky, Peter Szolovits, David W. Bates, Raja-Elie E. Abdunour, Atul J. Butte, and Emily Alsentzer. Assessing the Potential of

GPT-4 to Perpetuate Racial and Gender Biases in Health Care: A Model Evaluation Study. *The Lancet. Digital health*, 6(1):e12–e22, January 2024. ISSN 2589-7500. doi: 10.1016/S2589-7500(23)00225-X.

Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R. Dalal, Jennifer L. Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, Jack Boyd, Kathleen Boyd, Karen Hirsch, Curt Langlotz, Rita Lee, Joanna Melia, Joanna Nelson, Karim Sallam, Stacey Tullis, Melissa Ann Vogelsong, John Patrick Cunningham, and William Hiesinger. Almanac — Retrieval-Augmented Language Models for Clinical Medicine. *NEJM AI*, 1(2):AIoa2300068, 2024. doi: 10.1056/AIoa2300068.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating Text Generation with Bert. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and Reading: A Comprehensive Survey on Open-Domain Question Answering. Technical Report arXiv:2101.00774, 2021.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. Question Answering with Long Multiple-Span Answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.342.

Appendices

Appendix A

Distribution of Categories for MashTREC Questions

Category	Test	Training
Migraines Headaches	3	2
Beauty	3	0
Cold And Flu	2	4
Skin Problems And Treatments	2	3
Diabetes	2	2
A To Z Guides	2	2
Digestive Disorders	2	0
Multiple Sclerosis	2	0
Add Adhd	1	3
Cancer	1	3
Pain Management	1	2
Heart Disease	1	2
Ibd Crohns Disease	1	1
Mental Health	1	0
Fitness Exercise	1	0
Alzheimers	1	0
Hiv Aids	1	0
Oral Health	0	6
Lung	0	3
Brain	0	2
Rheumatoid Arthritis	0	2
Breast Cancer	0	2
Allergies	0	1
Back Pain	0	1
Palliative Care	0	1
Sexual Conditions	0	1
Healthy Aging	0	1
Prostate Cancer	0	1
Eye Health	0	1
Urinary Incontinence Oab	0	1
Ibs	0	1
Heartburn Gerd	0	1
Schizophrenia	0	1
Total	27	50

Appendix B

Guidelines for Annotation of Query-Question Matches

In this annotation tasks, you compare health-related questions and search queries that were submitted to a medical search engine. For each pair, the following question should be answered:

Does the search query encode the information need (knowledge gap) that was formulated in the question?

Think of the following exemplary scenario: A friend comes up to you and asks for medical advice. Since you don't know the answer to their question, you want to use a search engine to find relevant web documents that can help you answer the question. Is the presented query a good match to find answers to the question?

Please annotate each question-query pair with one of the two labels:

1. **match:**

The query and the question encode the same information need. (The query is suitable to search for documents that answer the question.)

Example: The question “Can you get sick from not sleeping enough?” and the query “sleep deprivation and disease” both ask for medical conditions that could be caused by not sleeping enough (i.e., sleep deprivation).

Question-query pairs still match if they use different vocabulary (e.g., synonyms).

2. **no match:**

The query and question do *not* encode the same information need. (The query is *not* suitable to search for documents that answer the question.) Even though the question and query might address similar topics, they

might focus on different directions.

Example: The question “What causes gastritis?” and the query “chronic constipation” are both related to the digestive system, but question focuses on causes and the query focuses on symptoms.

Question-query pairs do *not* match if:

- they focus on different sub-topics (e.g., cause vs. symptom, symptom vs. treatment)
- the question is more specific than the query
- the query is more specific than the question
- the query already contains (part of) the answer to the question

Note that you should *not* decide based on whether you would use the exact same query (e.g., because of prior knowledge or because you prefer a different style of queries) or not. If either the query or the question is not understandable with general knowledge (looking up definitions or synonyms is allowed), it should be labeled as “no match”.

Appendix C

Guidelines for Relevance Judgements for Training Set

The task is twofold: First you are asked to identify whether a text passage is relevant to the given health-related question. If the passage is relevant, also indicate whether the information it contains is correct or incorrect with respect to the reference answer given to the question.

Relevance

- **Relevant:**

The text passage either directly answers the question or provides enough information to determine an answer. A relevant passage must address all parts of the question.

Example: For the question “When do I need a tooth extraction for an infection?”, if the passage talks about conditions in which case tooth extraction would be required after a tooth infection, it is relevant.

- **Partially Relevant:**

The text passage answers part of the question but would need to be combined with other information to get a complete answer.

Example: For the question “When do I need a tooth extraction for an infection?”, if the passage talks about reasons for tooth extraction, but not specifically about infection as a reason, it is only partially relevant.

- **Not relevant:**

The text passage either does not address the question, or fails to address all parts of a question. A passage is **not** relevant if it:

- is not English
- contains adult material
- is garbled, empty, unreadable or otherwise broken, or
- is not clear whether the given question is answered or not, due to not being understandable without medical expertise.

Example: For the question “When do I need a tooth extraction for an infection?”, if the text passage talks about infections caused by tooth extraction (different direction of causality) or if it only lists other treatment options for tooth infections, but not tooth extraction, then the passage is not relevant.

Important: For a relevant text passage, it does not matter whether you believe the information provided in that passage is correct or incorrect. Only judge whether a user would likely find the information relevant regardless of the passage’s correctness.

Correctness

Judge the correctness of passages that were deemed either *relevant* or *partially relevant*. If the provided reference answer is *not relevant* to the question, judge the correctness of the text passage as *unknown*. Correctness should be assessed by comparing the answer from the text passage to the given reference answer.

- **Correct:**

The text passage contains information that matches the given reference answer to the question. A passage should also be judged as correct if it only contains parts of the information given in the reference answer but does not contradict the reference answer.

Example: For the question “When do I need a tooth extraction for an infection?”, if both the reference answer and the text passage list loosening of the tooth as a reason for extraction, the passage should be judged as correct, even if the passage or the reference answer also list other reasons.

- **Incorrect:**

The text passage contains information that contradicts the reference answer.

Example: For the question “When do I need a tooth extraction for an infection?”, if the reference answer list loosening of the tooth due to an infection as a reason for extraction but the text passage states that

tooth extraction would *not* be necessary in that case, the passage should be judged as incorrect. Also, if the reference answer list loosening of the tooth due to an infection as the *only* reason for extraction but the text passage lists other reasons as well, the passage should be judged as incorrect.

- **Unknown:**

The passage is not relevant (see above) or the passage only contains information that is not mentioned in the reference answer (i.e., the correctness cannot be directly verified).

Example: For the question “When do I need a tooth extraction for an infection?”, if the reference answer list loosening of the tooth due to an infection as one common reason for extraction but the text passage only mentions another common reason for tooth extraction following an infection, the correctness cannot be assessed based on the reference answer, and hence, is judged as unknown.

Summary

Please read the question and its reference answer. Then rate the relevance of each text passage to the question. For relevant passages, also rate whether the information in the passage matches the given answer or not.

Appendix D

Guidelines for Relevance Judgements for Test Set

The task is twofold: First you are asked to identify whether a text passage is relevant to the given health-related question. If the passage is relevant, also indicate whether the information it contains is correct or incorrect with respect to the reference answer given to the question.

Relevance

- **Relevant:**

The text passage either directly answers the question or provides enough information to determine an answer. A relevant passage must address all parts of the question and match the narrative provided alongside the question.

Example: For the question “When do I need a tooth extraction for an infection?”, if the passage talks about conditions in which case tooth extraction would be required after a tooth infection, it is relevant.

- **Partially Relevant:**

The text passage answers part of the question but would need to be combined with other information to get a complete answer.

Example: For the question “When do I need a tooth extraction for an infection?”, if the passage talks about reasons for tooth extraction, but not specifically about infection as a reason, it is only partially relevant.

- **Not relevant:**

The text passage either does not address the question, or fails to address all parts of a question. A passage is **not** relevant if it:

- is not English
- contains adult material
- is garbled, empty, unreadable or otherwise broken, or
- is not clear whether the given question is answered or not, due to not being understandable without medical expertise.

Example: For the question “When do I need a tooth extraction for an infection?”, if the text passage talks about infections caused by tooth extraction (different direction of causality) or if it only lists other treatment options for tooth infections, but not tooth extraction, then the passage is not relevant.

Important: For a relevant text passage, it does not matter whether you believe the information provided in that passage is correct or incorrect. Only judge whether a user would likely find the information relevant regardless of the passage’s correctness.

Correctness

Judge the correctness of passages that were deemed either *relevant* or *partially relevant*. If the provided reference answer is *not relevant* to the question, judge the correctness of the text passage as *unknown*. Correctness should be assessed by comparing the answer from the text passage to the given reference answer.

- **Correct:**

The text passage contains information that matches the given reference answer to the question. A passage should also be judged as correct if it only contains parts of the information given in the reference answer but does not contradict the reference answer.

Example: For the question “When do I need a tooth extraction for an infection?”, if both the reference answer and the text passage list loosening of the tooth as a reason for extraction, the passage should be judged as correct, even if the passage or the reference answer also list other reasons.

- **Incorrect:**

The text passage contains information that contradicts the reference answer.

Example: For the question “When do I need a tooth extraction for an infection?”, if the reference answer list loosening of the tooth due to an infection as a reason for extraction but the text passage states that

tooth extraction would *not* be necessary in that case, the passage should be judged as incorrect. Also, if the reference answer list loosening of the tooth due to an infection as the *only* reason for extraction but the text passage lists other reasons as well, the passage should be judged as incorrect.

- **Unknown:**

The passage is not relevant (see above) or the passage only contains information that is not mentioned in the reference answer (i.e., the correctness cannot be directly verified).

Example: For the question “When do I need a tooth extraction for an infection?”, if the reference answer list loosening of the tooth due to an infection as one common reason for extraction but the text passage only mentions another common reason for tooth extraction following an infection, the correctness cannot be assessed based on the reference answer, and hence, is judged as unknown.

Summary

Please read the question and its reference answer. Then rate the relevance of each text passage to the question. For relevant passages, also rate whether the information in the passage matches the given answer or not.

Appendix E

Guidelines for Preference Judgements on Generated Answers

Your task is to compare two answers to a given health-related question. Please choose the answer that you prefer overall from a medical perspective. The following aspects can help you compare the two answers:

- Is each answer consistent with your medical knowledge and standard practice? - Does one answer cover more aspects of the question?
- Does one answer contain irrelevant content?
- Is the answer understandable to a lay person?
- Which answer is closer to the one you would give to your patients as a health professional?

If you like or dislike both answers equally, please select “Both Equal”. If you feel that you do not have the medical expertise to choose between the two answers given, please select “Unsure”.

Appendix F

System Prompts Used in the Grid Search

[INST]You are an accurate and reliable medical AI assistant that can answer questions with the help of external documents. It is very important that you limit the length of your answer to 3 sentences in language that is understandable for non-professionals. Please note that external documents may contain noisy or factually incorrect information. If the information in the document contains the correct answer, you will give an accurate answer. If the information in the document does not contain the answer, you will generate "I can not answer the question because of the insufficient information in documents." If there are inconsistencies with the facts in some of the documents, please generate the response "There are factual errors in the provided documents." and provide the correct answer. [/INST]

[INST]Answer a medical question truthfully using the documents provided below. [/INST]

[INST]Answer the question using the information from the given context. [/INST]

No system prompt

Appendix G

Prompt for the Generation of Long Form Answers

[INST]You are an accurate and reliable medical AI assistant that can answer questions with the help of external documents. It is very important that you limit the length of your answer to 3 sentences in language that is understandable for non-professionals. Please note that external documents may contain noisy or factually incorrect information. If the information in the document contains the correct answer, you will give an accurate answer. If the information in the document does not contain the answer, you will generate "I can not answer the question because of the insufficient information in documents." If there are inconsistencies with the facts in some of the documents please generate the response "There are factual errors in the provided documents." and provide the correct answer. [/INST]

Question: What are normal flu symptoms?

Document: Chest Pain, Breathing, High Fever, and More People typically recover after about a week or two without any lasting problems. But sometimes this illness can lead to serious complications that require emergency care. Every year more than 200,000 people in the U.S. wind up in the hospital because of the flu. Tens of thousands die. Infants, the elderly, and people with certain diseases or weakened immune systems are the most at risk. But a flu emergency can happen to anyone. So it's important to know the signs of trouble. Normal Flu Symptoms Different strains of the influenza virus cause the flu. You get it when you inhale the germ or pick it up on your hands and then

APPENDIX G. PROMPT FOR THE GENERATION OF LONG FORM ANSWERS

touch you eyes, nose, or mouth. Symptoms usually show up 1 to 4 days later. The flu can be hard to tell from a cold. But it usually comes on faster and is more severe.

Answer: Normal flu symptoms include:

High fever, Headache, Tiredness (can be extreme), Cough, Sore throat, Runny or stuffy nose, Body ach.e Although flu vaccines can prevent certain strains, there's no much you can do after you get sick.

Question: `#{question}`

Document: `#{summarized document}`

Answer:

Appendix H

Prompt for the Generation of Long Form Answers without Context

[INST]You are an accurate and reliable medical AI assistant that can answer questions. It is very important that you limit the length of your answer to 3 sentences in language that is understandable for non-professionals.[/INST]

Question: What are normal flu symptoms?

Answer: Normal flu symptoms include:

High fever, Headache, Tiredness (can be extreme), Cough, Sore throat, Runny or stuffy nose, Body ach.e Although flu vaccines can prevent certain strains, there's no much you can do after you get sick.

Question: \${question}

Answer:

Appendix I

Prompt for the Completeness Metric

You are assessing a submitted answer to a given question based on different criteria. You are also given an expert answer for reference. Here is the data:

[BEGIN DATA]

[Question]: \${question}

[Expert]: \${reference answer}

[Submission]: \${generated answer}

[Criteria]:

Does the answer address all aspects of the question? (0.0 means does not address at all, 1.0 means address all the aspects)

Does the answer omit any important content?

(0.0 means omits all the important content, 1.0 means contains all important aspects)

Does the answer contain any irrelevant

content (0.0 means all the content is irrelevant, 1.0 means all the content is relevant)

*** [END DATA]

Does the submission meet the criteria?

Decide on each of the 3 criteria and assess as a continuous

APPENDIX I. PROMPT FOR THE COMPLETENESS METRIC

score in the range from 0.0 to 1.0 with step 0.01
(examples: 0.01, 0.59, 0.87, and so on).
Higher is better. Return a list of 3 scores.

Declaration of Academic Integrity

1. I hereby confirm that this work — or in case of group work, the contribution for which I am responsible and which I have clearly identified as such — is my own work and that I have not used any sources or resources other than those referenced.

I take responsibility for the quality of this text and its content and have ensured that all information and arguments provided are substantiated with or supported by appropriate academic sources. I have clearly identified and fully referenced any material such as text passages, thoughts, concepts or graphics that I have directly or indirectly copied from the work of others or my own previous work. Except where stated otherwise by reference or acknowledgement, the work presented is my own in terms of copyright.

2. I understand that this declaration also applies to generative AI tools which cannot be cited (hereinafter referred to as ‘generative AI’).

I understand that the use of generative AI is not permitted unless the examiner has explicitly authorized its use (Declaration of Permitted Resources).

Where the use of generative AI was permitted, I confirm that I have only used it as a resource and that this work is largely my own original work. I take full responsibility for any AI-generated content I included in my work. Where the use of generative AI was permitted to compose this work, I have acknowledged its use in a separate appendix. This appendix includes information about which AI tool was used or a detailed description of how it was used in accordance with the requirements specified in the examiner’s Declaration of Permitted Resources.

I have read and understood the requirements contained therein and any use of generative AI in this work has been acknowledged accordingly (e.g. type, purpose and scope as well as specific instructions on how to acknowledge its use).

3. I also confirm that this work has not been previously submitted in an identical or similar form to any other examination authority in Germany or abroad, and that it has not been previously published in German or any other language.
4. I am aware that any failure to observe the aforementioned points may lead to the imposition of penalties in accordance with the relevant examination regulations. In particular, this may include that my work will be classified as deception and marked as failed. Repeated or severe attempts to deceive may also lead to a temporary or permanent exclusion from further assessments in my degree programme.

Leipzig, May 8, 2024

.....
Lukas Zeit-Altpeter