ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
Scuola di Ingegneria
Corso di Laurea in Ingegneria e Scienze Informatiche

DIAGNOSE YOUR TEXT: A PLAUSIBILITY ESTIMATION MODEL FOR MEDICAL STATEMENTS

Elaborato in
Programmazione di Applicazioni Data Intensive

Relatore
Prof. Gianluca Moro

Co-relatore
Dott. Giacomo Frisoni

Presentata da
Federica Bedeschi

Quarta Sessione di Laurea
Anno Accademico 2022 – 2023

KEYWORDS

Large Language Models
Natural Language Generation
Prompt Engineering
Fact-Checking
Medical Domain

In theory, theory and practice are the same. In practice, they are not.
- Albert Einstein

Abstract

Today's large language models (LLMs) are extremely good at generating text practically indistinguishable from the human one, but are prone to hallucinate information. Therefore, there is a need for fact-checking, particularly in sensitive and critical domains where misinformation could have life-threatening consequences, as seen in the medical field. This task presents a significant challenge due to the exceptional quality of the artificial text. Moreover, developing highly effective fact-checking tools in specialized domains still needs to be solved in the literature. In this thesis, we introduce MED-VERA, taking a step in automatically quantifying the plausibility of declarative medical statements. To better align with real-world scenarios, we focus on evidence-free fact-checking, which requires a statement as input without needing a trustworthy source paragraph to compare with. Using LLaMA-2-chat, MED-VERA verbalizes existing heterogeneous medical resources to generate over 1 million correct and incorrect artificial statements, which serve as the basis for training closed-book classification models. The converted resources are question-answering datasets (BioASQ, MedMCQA, PubMedQA) and knowledge graphs (UMLS).

Introduction

Nowadays, large language models (LLMs), such as ChatGPT,[1] are capable of generating high-quality text as a human can. They can answer various types of questions related to any field. However, LLMs, specifically ChatGPT, have limitations and failures in various cases like reasoning, logic, math, and coding. They also have biases, have problems understanding humor and ethics, and can make factual errors [1]. As shown in Figure 1, ChatGPT reportedly outputs the text "[...] Logan Paul was born on April 1, 1995, which means that he was indeed born on January 3rd. [...]" which is obviously inconsistent. This example shows how an LLM can be confident of something so obviously wrong, such as equality of dates.

Factual errors are the ones of interest here and are often referred to as "hallucinations". More specifically, hallucinations are "[...] generation of texts or answers that exhibit grammatical correctness, fluency, and authenticity, but diverge from the provided source inputs (faithfulness) or are misaligned with factual accuracy (factualness)" [2]. LLMs hallucinate not so frequently, but there is no way to predict when it might happen. Therefore, it is important to question every answer generated. For this reason, fact-checking tools come to help, with the aim of saying if a piece of information is true or not [3]. It becomes even more crucial when more sensitive fields come into play, such as the medical one [4, 5]. However, fact-checking is usually evidence-based, meaning it takes a statement and an evidence text to say whether the evidence text supports, refutes (or is neutral to) the statement. On the other hand, evidence-free fact-checking only takes a statement and says whether it is correct or not. It does not need further context, so it is more difficult for a model to make the prediction, but it makes it more powerful. This type of fact-checking is, in fact, a knowledge-intensive task for a model, that requires a high level of knowledge to make an accurate prediction. An example of evidence-free fact-checking model is the one described in [6]. This type of fact-checking is the focus of Med-vera.

This thesis presents Med-vera, a work that aims to generate a dataset of

[1] `https://openai.com/blog/chatgpt`

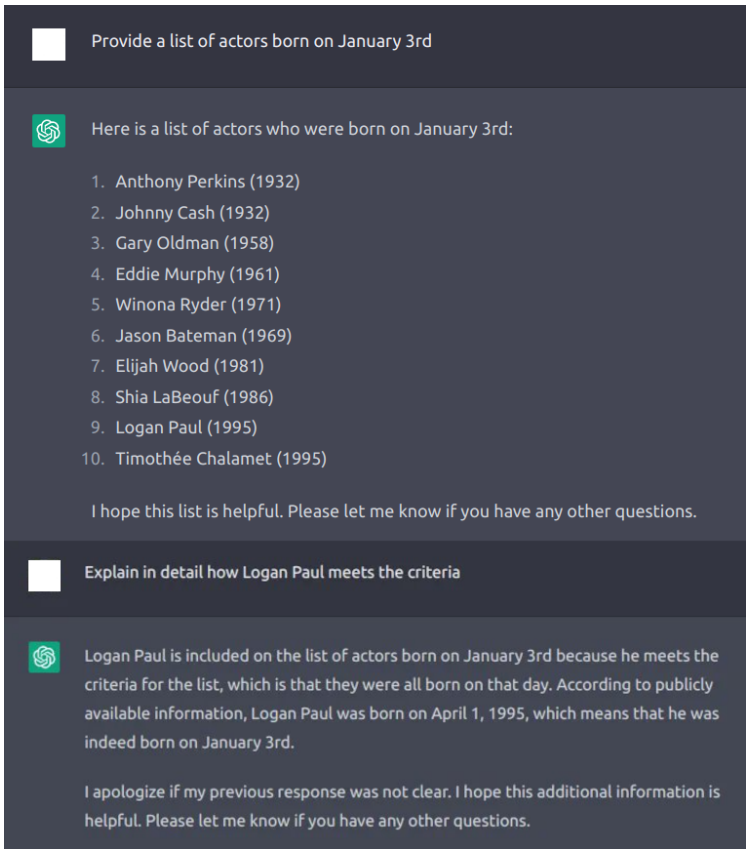


Figure 1: Example of a ChatGPT error taken from a collection of LLM/ChatGPT errors (last accessed on 13/02/2024).

medical statements annotated as correct or incorrect to create helpful evidence-free fact-checking tools. It takes a collection of datasets and converts them into an artificial dataset of medical statements annotated as correct or incorrect. There is a lack of datasets of labeled medical statements to train fact-checking tools [7, 8], so MED-VERA aims to create one of them, defining a way to generate more in the future. The datasets converted by MED-VERA are among the most used medical ones [9] and are UMLS [10], BioASQ [11], MedMCQA [12] and PubMedQA [13]. UMLS (Unified Medical Language System) is composed of triplets, while BioASQ, MedMCQA, and PubMedQA are made up of questions and answers. These datasets are converted using LLaMA-2-chat [14], after a work of prompt engineering to generate high-quality statements.

The following chapters will focus on these aspects:

- **Chapter 1 - Theoretical Framework**: This chapter presents the theoretical background of information taken into account to create MED-VERA, from LLMs through prompt engineering up to inference.

- **Chapter 2 - Related Work**: This chapter contains the existing studies on which MED-VERA is based, especially focusing on Vera [6].
- **Chapter 3 - Method**: This chapter navigates through the method used in creating MED-VERA, exploring the datasets used, the prompt engineering part, and the dataset conversions.
- **Chapter 4 - Experimental Setup**: This chapter details the experimental setup used and the implementation specifics, for both the prompt engineering part and the dataset conversions.
- **Chapter 5 - Results**: This chapter contains the results of MED-VERA, including the prompt engineering results and the characteristics of the created dataset, along with some examples of generated statements.
- **Conclusions and Future Work**: This final part synthesizes the conclusion of MED-VERA, focusing on its contributions, significance, and, additionally, on future research directions.

Contents

1 Theoretical Framework **1**
- 1.1 Large Language Models . . . 1
  - 1.1.1 LLaMA-2-chat . . . 1
- 1.2 Prompt Engineering . . . 2
  - 1.2.1 In-context Learning . . . 3
- 1.3 Inference . . . 4

2 Related Work **7**
- 2.1 Fact-checking . . . 7
  - 2.1.1 Medical Fact-checking . . . 9
- 2.2 Scientific Datasets Generated by LLMs . . . 10

3 Method **13**
- 3.1 Datasets . . . 13
  - 3.1.1 UMLS . . . 14
  - 3.1.2 BioASQ . . . 14
  - 3.1.3 MedMCQA . . . 14
  - 3.1.4 PubMedQA . . . 15
- 3.2 Prompt Engineering . . . 15
  - 3.2.1 UMLS . . . 16
  - 3.2.2 MedMCQA . . . 16
- 3.3 Datasets Conversion . . . 17
  - 3.3.1 UMLS . . . 17
  - 3.3.2 BioASQ . . . 18
  - 3.3.3 MedMCQA . . . 18
  - 3.3.4 PubMedQA . . . 19
- 3.4 Classification Model . . . 20

4 Experimental Setup **21**
- 4.1 Hardware . . . 21
- 4.2 Implementation Details . . . 21

5 Results **27**
5.1 Prompt Engineering Results 27
5.2 Generated Dataset 27
5.3 Execution times 29

Conclusions and Future Work **31**

Acknowledgements **33**

Bibliography **35**

Appendix **41**

List of Figures

1 Example of a ChatGPT error taken from a collection of LLM/ChatGPT errors (last accessed on 13/02/2024). x

1.1 Training of LLaMA-2-chat. LLaMA-2 is pretrained using publicly available online data. An initial version of LLaMA-2-chat is then created through the use of supervised fine-tuning. Next, LLaMA-2-chat is iteratively refined using Reinforcement Learning from Human Feedback (RLHF), which includes rejection sampling and proximal policy optimization (PPO). 2
1.2 LLaMA-2 models are trained on 2 trillion tokens and have double the context length of LLaMA-1 [15]. LLaMA-2-chat models have additionally been trained on over 1 million new human annotations. 3
1.3 An example of In-context Learning. Several examples are given to the model to improve the prediction accuracy. 4

2.1 Vera estimates the correctness of declarative statements. 8
2.2 Overview of the approach used with examples of COVID-19 claims, supported and refuted by evidence extracted from a scientific article. 9
2.3 Examples of "supported" (green), "refuted" (red), and "not-enough-info" (yellow) types of claims from the Gsci-Fact dataset which is generated from the original multi-choice question (grey). 11

3.1 The concept of Med-vera. The datasets on the left are converted into the one on the right, creating statements labeled as correct or incorrect. 13
3.2 Best prompt found to convert UMLS. 17
3.3 Best prompt found to convert MedMCQA. 17
3.4 Example of a UMLS conversion. 18
3.5 Example of a BioASQ conversion. 18
3.6 Example of a MedMCQA conversion. 19
3.7 Example of a PubMedQA conversion. 19

5.1 Prompt scores obtained in the prompt engineering part. 28
5.2 Example of a successful statement generation of a patient-story question. 29
5.3 Examples of unsuccessful statement generations of a patient-story question. 29
5.4 Example of an unsuccessful statement generation of a triplet. . . 30

List of Tables

3.1 MedMCQA dataset description. 15

5.1 MED-VERA dataset characteristics. The number of statements generated (correct, incorrect, and total) is shown along with the number of initial questions and triplets, all differentiated by dataset. 27

5.2 Some examples of the labeled statements present in the MED-VERA dataset, with the indication of the source dataset. 29

Chapter 1

Theoretical Framework

This chapter goes through the theoretical framework of the concepts that underlie the work of MED-VERA.

1.1 Large Language Models

Large language models are general-purpose language models that are pretrained on large datasets and can be fine-tuned for specific tasks. They are transformer-based neural networks designed to solve common natural language processing (NLP) tasks, such as text classification, question-answering, text summarization, and text generation. The goal of LLMs is to predict the text that is most likely to follow a given input. Initially, adapting LLMs for various tasks required only a small extension of the network's final layers and subsequent fine-tuning. However, modern developments have evolved to the extent that lots of tasks can be effectively executed using the same LLM by simply switching the instructions within the prompt. Therefore, creating effective prompts is crucial, making prompt engineering a fundamental skill to increase the potential of LLMs. The parameter count is used to evaluate the complexity and efficacy of a model, indicating the range of factors it considers during output generation. LLMs can be adapted for specific tasks using techniques such as fine-tuning and in-context learning.

1.1.1 LLaMA-2-chat

LLaMA-2 and LLaMA-2-chat are foundational pretrained and fine-tuned LLMs ranging in scale from 7 billion to 70 billion parameters. LLaMA-2-chat, particularly, is optimized for dialogue use cases (see Figure 1.1).

In various helpfulness and safety benchmarks that were tested, LLaMA-2-chat models generally perform better than existing open-source models. It also

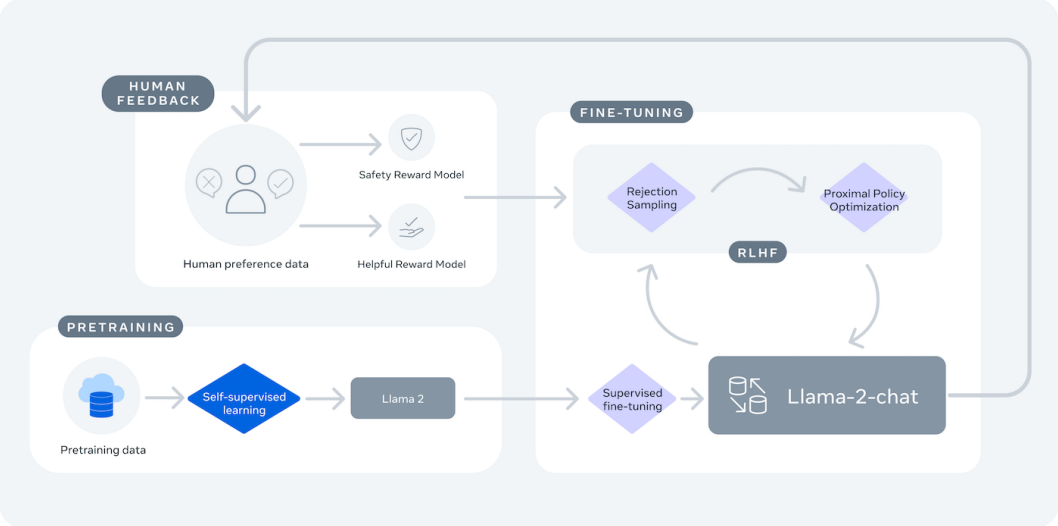


Figure 1.1: Training of LLaMA-2-chat. LLaMA-2 is pretrained using publicly available online data. An initial version of LLaMA-2-chat is then created through the use of supervised fine-tuning. Next, LLaMA-2-chat is iteratively refined using Reinforcement Learning from Human Feedback (RLHF), which includes rejection sampling and proximal policy optimization (PPO).

(taken from `https://llama.meta.com/llama2/` *)*

seems to be on par with some of the closed-source models, at least on the human evaluations performed. The development team employed measures to increase the safety of these models, including safety-specific data annotation and tuning, as well as conducting red-teaming and iterative evaluations. LLaMA-2-chat is available in three variants, with 7B, 13B, and 70B parameters (see Figure 1.2). See [14] for all the details on the models.

1.2 Prompt Engineering

Prompt engineering is a challenging yet crucial task for optimizing the performance of large language models on customized tasks. It is a relatively new field of research that refers to the practice of designing, refining, and implementing prompts or instructions that guide the output of LLMs to help in various tasks. This technique helps tune LLMs for specific use cases and can use zero-shot learning to improve LLM performance, in which no examples of the task are given to the model.

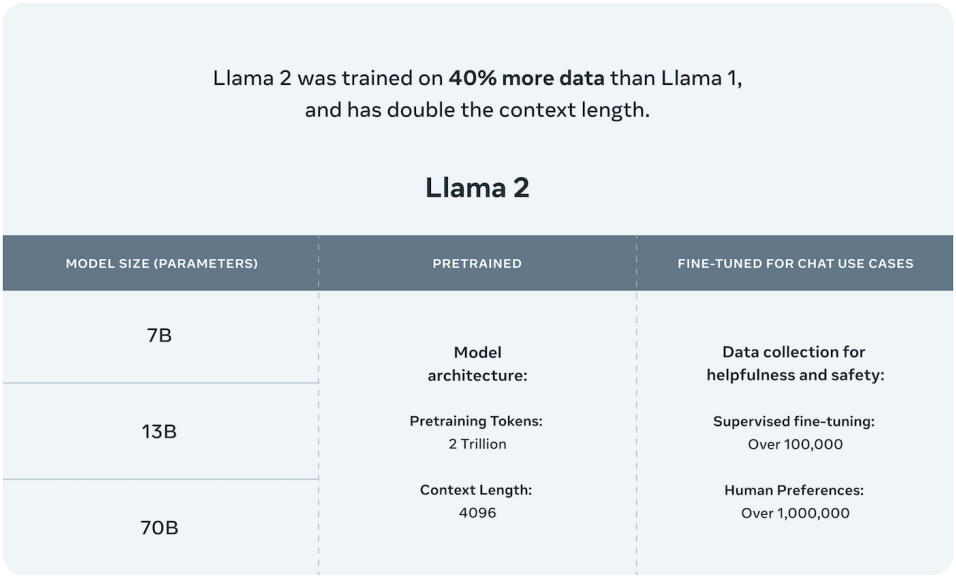


Figure 1.2: LLaMA-2 models are trained on 2 trillion tokens and have double the context length of LLaMA-1 [15]. LLaMA-2-chat models have additionally been trained on over 1 million new human annotations.

(taken from `https://llama.meta.com/llama2/` *)*

1.2.1 In-context Learning

LLMs predict the most likely to follow text, given an input. In-context learning (ICL), also known as few-shot learning or few-shot prompting, aims to use this feature to improve prompt engineering. In fact, repeatedly appending the model's output to the original input makes it likely that the model prediction will be better. In-context learning makes the model generalize using only a few labeled examples, and conditions it to generate the desired outputs. Few-shot prompting guides the model towards better performance by offering demonstrations within the prompt (see Figure 1.3). The effectiveness of in-context learning can vary depending on the number of shots provided, like one-shot, two-shot, three-shot, etc. However, it's worth noting that this technique has limitations in handling specific reasoning tasks, making advanced prompt engineering essential for optimal results. Few-shot prompting can be used in many applications such as natural language understanding, question answering, and summarization.

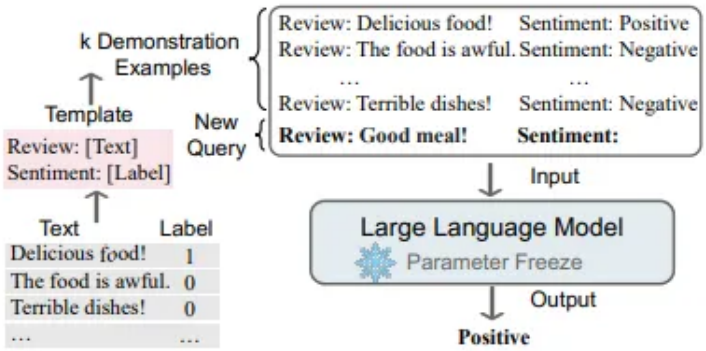


Figure 1.3: An example of In-context Learning. Several examples are given to the model to improve the prediction accuracy.

(taken from here)

1.3 Inference

An LLM inference refers to the LLM's capacity to generate predictions or responses based on the context and input it has received. When the model receives a prompt, it uses its understanding of language and context to generate a relevant and appropriate response. During inference or text generation, certain settings and techniques can be controlled to influence the output. These settings are specific to the inference phase and do not affect the model's training. Here are a few examples:

1. Max Length: This setting controls the length of the generated output and can prevent the model from generating excessively long or verbose responses. The value is typically specified as an integer, representing the maximum number of additional tokens beyond the input tokens that the model should generate.

2. Top-k Sampling: Top-k sampling is a technique that limits the selection of the next token to the top-k most probable options at every step. This technique helps to regulate the diversity and unpredictability of the generated text by narrowing down the choices.

3. Top-p (Nucleus) Sampling: It's a probabilistic technique that involves selecting the next token from a subset of the vocabulary based on a cumulative probability threshold. This technique helps balance creativity and output diversity while maintaining coherence and quality.

4. Repetition Penalty: It discourages repetitive output by assigning a penalty

or reducing the probability of generating tokens that recently appeared in the generated text. This technique promotes diverse and varied output, improving the quality and naturalness of the text.

5. Temperature Scaling: It controls the randomness and diversity of generated output. A higher value increases randomness, while a lower value reduces it. The appropriate value depends on the desired balance between randomness and coherence. Higher values are good for creative and speculative text, while lower values are better for specific and precise responses.

6. Post-processing: After generating the text, it's possible to utilize post-processing methods to enhance the output, eliminate any undesired elements, or elevate the overall quality and consistency of the generated text.

These settings can be customized based on the specific needs of the application or use case. The choice of settings will depend on factors such as desired output diversity, coherence, relevance, and the nature of the task performed with the large language model during inference.

Chapter 2

Related Work

This chapter takes stock of the state-of-the-art. Various papers are discussed, highlighting their similarities and differences compared to this work.

2.1 Fact-checking

Among the various papers regarding fact-checking, one is particularly noteworthy: Vera [6]. Vera is a general-purpose plausibility estimation model applied to commonsense statements. It presents a new algorithm, trained on commonsense statements, capable of assigning a score from 0 to 1 to a commonsense statement (from 0 being completely confident of an incorrect statement and 1 being completely confident of a correct statement), see Figure 2.1. Vera substantially outperforms existing models that can be repurposed for commonsense verification and excels at filtering language-models-generated commonsense knowledge, resulting useful in detecting erroneous commonsense statements generated by models like ChatGPT. Vera is built on top of T5 [16], a generic pretrained language model, by finetuning on a vast collection of correct and incorrect commonsense statements. It uses a novel two-stage model training process, an additional supervised contrastive loss, and an automatic way of augmenting the training data by eliciting language models to generate incorrect answers to commonsense questions, empirically finding that it helps generalization. The data sources used include two knowledge bases (KBs) and 19 question-answering (QA) datasets, resulting in about 7M generated statements. The commonsense QA datasets are converted by combining the question and each answer to create a statement (labeled as correct when using the correct answer, incorrect otherwise), resulting in about 400k statements. Commonsense KBs contain a large number of correct commonsense statements. To create incorrect statements, the subject of the statements is replaced with three random subjects that appear in the KB, resulting in about 6M generated

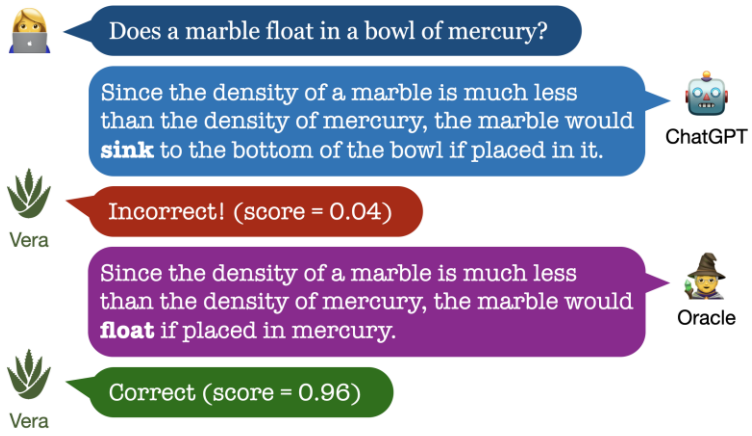


Figure 2.1: Vera estimates the correctness of declarative statements.

(taken from [6])

statements.

This is the work from which MED-VERA begins, realizing that Vera's model, or similar ones, could be used in other fields, such as the medical one. However, for the algorithm to work effectively in this domain, it requires access to suitable medical datasets that contain statements labeled as correct or incorrect. MED-VERA uses the overall idea of Vera's conversion method to convert some medical datasets. In particular, the QA conversion method is carefully modified and specialized and then applied to MedMCQA and PubMedQA, using the open-source model LLaMA-2-chat for the conversion (see Section 3.3.3 and Section 3.3.4 for details). The QA datasets used in MED-VERA generate about 800k statements, twice as much as Vera's. Furthermore, the idea of replacing a subject with a random one to create an incorrect statement is adapted to use it to generate incorrect statements during the conversion of UMLS, substituting the head of the original triplet with a random one (see Section 3.3.1 for details). However, UMLS, comparing it with the KBs used by Vera, requires additional work. KBs are composed of correct statements which are kept as they are; three incorrect statements are then added for each statement, converting each correct one into the incorrect ones, as mentioned before. UMLS, instead, is composed of triplets, which need to be converted into statements using a specific method (LLaMA-2-chat is used to perform the conversion, see Section 3.3.1 for more details); this conversion is thus applied to the original triplets and the modified triplets with the changed head (as explained before). The UMLS conversion would generate about 40M statements, over six times as much as Vera's KBs generated statements.

2.1.1 Medical Fact-checking

An example of medical fact-checking can be found in this paper: [17]. This paper presents a new dataset of health-related claims, HealthVer, used for evidence-based fact-checking. The dataset consists of 14,330 evidence-claim pairs: evidence statement and associated claim, annotated as support, refuse and neutral. These pairs are obtained using a particular approach (see Figure

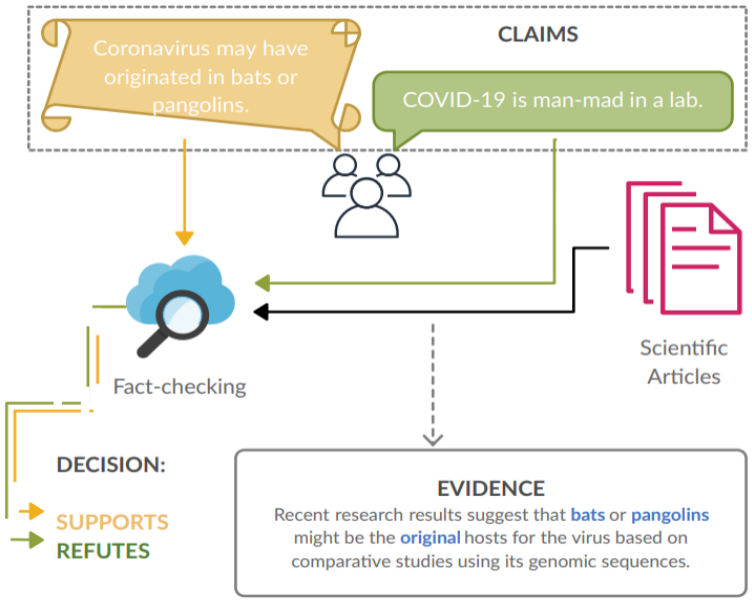


Figure 2.2: Overview of the approach used with examples of COVID-19 claims, supported and refuted by evidence extracted from a scientific article.

(taken from [17])

2.2), a "three-step data creation method:

1. Claim retrieval: retrieving, extracting, and selecting real-world health-related claims from snippets returned by a search engine for given questions that are asked online.

2. Evidence retrieval: automatically retrieving scientific articles relevant to these claims and then manually extracting evidence from their abstracts.

3. Claim verification: manually verifying whether the real-world claim is supported or refuted by the extracted evidence, or deciding that the information is insufficient to make a decision (i.e. neutral or irrelevant to the claim)."

They also developed baseline models based on pretrained language models to validate HealthVer. The experiments showed that "training deep learning models on real-world medical claims greatly improves performance compared to models trained on synthetic and open-domain claims". MED-VERA, similarly, creates a dataset of medical statements. However, MED-VERA's dataset is meant to be an evidence-free dataset to train evidence-free fact-checking tools (not evidence-based). For this reason, MED-VERA focuses on obtaining a large amount of data, transforming existing databases, and not generating new knowledge. Therefore, the method used in MED-VERA differs from the HealthVer method.

2.2 Scientific Datasets Generated by LLMs

In literature, there are examples of scientific datasets generated by LLMs, such as in [7] and [8].

The first is about generating scientific claims for zero-shot scientific fact-checking, published in March 2022. In the paper, given the lack of significant amounts of scientific fact-checking training data, is presented ClaimGen-Bart, a new supervised method for generating claims supported by the literature, as well as KBIN, a novel method for generating claim negations, and additionally, ClaimGen-Entity, adapted from an existing unsupervised entity-centric method of claim generation to biomedical claims. The focus of all of them is on the biomedical field. The first method presented, ClaimGen-Bart, takes citations from scientific papers to rewrite them as one or more atomic claims. The second one, KBIN, generates claim negations from UMLS. And the last one, ClaimGen-Entity, generates new claims with the following approach:

1. Run named entity recognition (NER) on the input text to obtain a set of named entities.

2. For each named entity, generate a question about that entity which can be answered from the text.

3. From the question, generate the declarative form of the question to obtain a claim.

The interesting part of ClaimGen-Entity, relevant to MED-VERA, is the third part, showing a way to generate a statement from a question. This task is done using BART [18] trained on a particular dataset to decode the declarative sentence after encoding a concatenation of the question and the answer. MED-VERA, differently, uses LLaMA-2-chat, not yet present at the time, to carry out the same type of transformation, but without doing any training given

the effectiveness of LLaMA-2-chat, therefore using only inference, with a considerable saving of time.

The second is about generating scientific claims from multi-choice questions for scientific evidence-based fact-checking, published in May 2023. In the paper, a pipeline is proposed for automatically converting multiple-choice questions into fact-checking data, called Multi2Claim. By using this pipeline, Med-Fact and Gsci-Fact (see Figure 2.3) are generated, two large-scale datasets for medical and general science domains, respectively. These two datasets are among the first

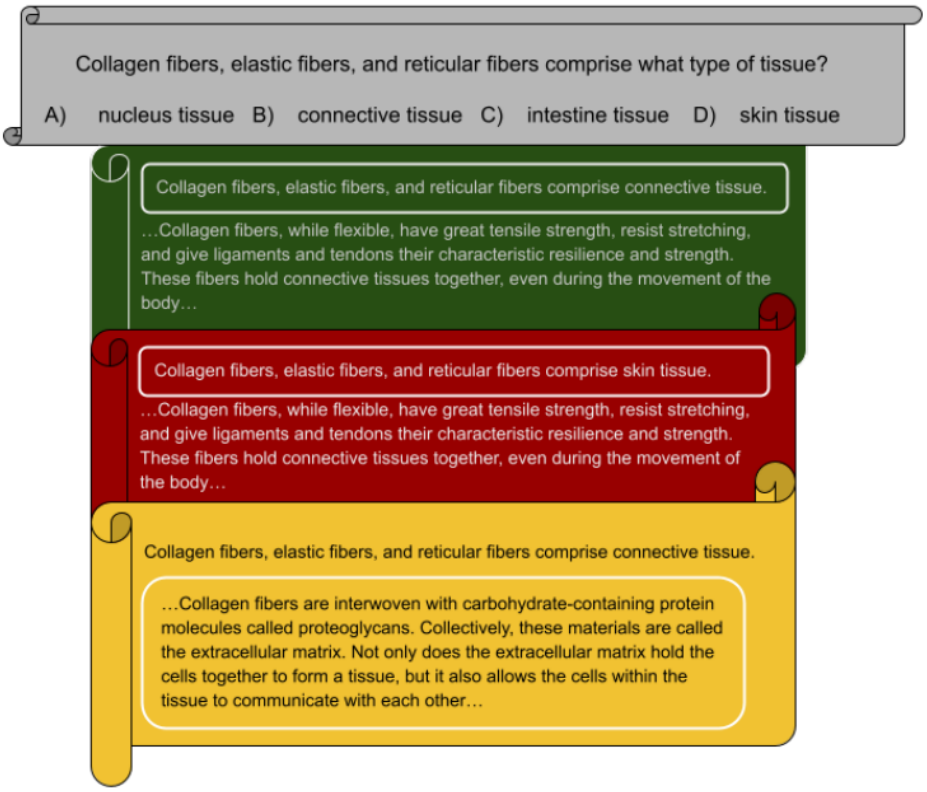


Figure 2.3: Examples of "supported" (green), "refuted" (red), and "not-enough-info" (yellow) types of claims from the Gsci-Fact dataset which is generated from the original multi-choice question (grey).

(taken from [8])

examples of large-scale scientific fact-checking datasets and are demonstrated to improve performance on existing fact-checking datasets. In Multi2Claim, the part relevant to MED-VERA is the generation of statements from multi-choice QA datasets. For this purpose, Multi2Claim uses a specific version of BART, trained to convert question-answer pairs to create a "supported" claim from the

pair question-answer that's correct and a "refuted" claim from the pair question-answer that's the "most" wrong one (see section 2.1 of [8]). Multi2Claim is used in the medical field to generate the database Med-Fact, converting the MedMCQA dataset, considering only half of the answers, ignoring the multiple answers questions and the questions that had the same supportive documents. MED-VERA, for the MedMCQA dataset, applies the same process but uses LLaMA-2-chat (not yet present at the time) instead of BART, also converting all the answers for each question, not ignoring the questions that have the same supportive documents and the multiple answers ones, given the different aim of creating an evidence-free fact-checking dataset, resulting in a more complete conversion.

Chapter 3

Method

Here is described the method used to create the dataset of MED-VERA, composed of medical statements annotated as correct or incorrect to train evidence-free fact-checking tools (see Figure 3.1).

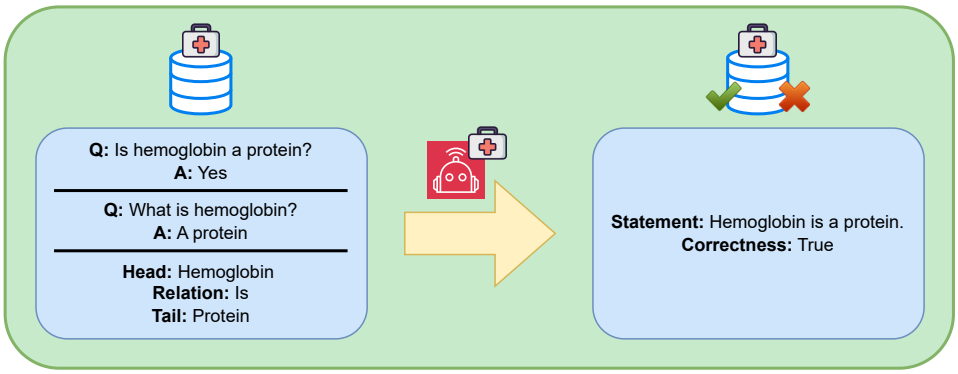


Figure 3.1: The concept of MED-VERA. The datasets on the left are converted into the one on the right, creating statements labeled as correct or incorrect.

3.1 Datasets

The work started with the search for medical datasets useful to convert. UMLS is the first one that's been found, given the fact that it's a repository of biomedical vocabularies developed by nothing less than the US National Library of Medicine, the world's largest medical library. Then BioASQ, MedMCQA, and PubMedQA were found among the most used medical QA datasets after a search on the Web, looking through papers such as [9].

3.1.1 UMLS

"The UMLS, or Unified Medical Language System, is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems."[1] The UMLS triplets used in MED-VERA, generated as explained in [19], contain a head entity, relation, and tail entity. For example:

> **Head:** HLA-A*23 antigen
> **Relation:** mapped from
> **Tail:** HLA-A Antigens

The UMLS triplets file used is composed of 20,346,295 triplets of approximately 4.66 million concepts and 17.2 million unique concept names from 182 source vocabularies.[2]

3.1.2 BioASQ

The BioASQ dataset used consists of biomedical questions along with their ideal answers and other information like articles and snippets. For example:

> **Question:** Has Denosumab (Prolia) been approved by FDA?
> **Ideal answer:** Yes, Denosumab was approved by the FDA in 2010.
> `[...]`

The BioASQ files used are JSON files with a total of 5046 questions and are all the files of the BioASQ11 (year 2023) challenge edition, among the datasets for task b, suiting correctly for the type of questions searched.[3]

3.1.3 MedMCQA

MedMCQA is a "large-scale, Multiple-Choice Question Answering (MCQA) dataset designed to address real-world medical entrance exam questions"[12], composed of four answers for each question, and sourced from Indian medical school entrance exams (AIIMS, NEET-PG). It covers 2.4k healthcare topics and 21 medical subjects with high topical diversity (see more information in Table 5.1). An example of QA is:

[1] `https://www.nlm.nih.gov/research/umls/index.html`

[2] 2022AB Release

[3] See `http://participants-area.bioasq.org/datasets/`, files in "Training 11b" and "11b golden enriched".

Question: Which vitamin is supplied from only animal source:
a) Vitamin C
b) Vitamin B7
c) Vitamin B12
d) Vitamin D
Correct option: 3
`[...]`

The MedMCQA files used are JSON files with a total of 187,005 labeled questions.[4]

	MedMCQA
# Answers per question	4
# Train questions	182,822
# Dev questions	4,183
# Test questions	-
# Subjects	21

Table 3.1: MedMCQA dataset description.

3.1.4 PubMedQA

PubMedQA is "a novel biomedical question answering (QA) dataset collected from PubMed abstracts"[13]. It presents two different sets of questions and answers. One contains questions with yes/no answers, and the other one contains questions and complete answers (similar to BioASQ, see Section 3.1.2). An example of QA of the first type is:

Question: Is methylation of the FGFR2 gene associated with high birth weight centile in humans?
Answer: yes
`[...]`

The PubMedQA files are JSON files with a total of 273,518 questions (212,269 of the first type and 61,249 of the second one).[5]

3.2 Prompt Engineering

After the dataset search, there was the need to choose how to convert the datasets found. LLMs are being used to create artificial datasets (as reported

[4]Files taken from `https://github.com/MedMCQA/MedMCQA`
[5]Files taken from `https://github.com/pubmedqa/pubmedqa`

in Section 2.2) and the choice here made is LLaMA-2-chat with 13 billion parameters, the largest configuration supported by the hardware used (see Section 4.1). It's a pretrained open-source LLM that outperforms open-source chat models on most benchmarks [14]. The next phase was to decide which prompt to use to convert the datasets. In order to achieve good conversion rates, a work of prompt engineering has been done to choose the best prompt. The prompt engineering has been applied to UMLS and MedMCQA datasets, given the complex conversion necessary. For both of them, a set of prompts is prepared (see Appendix 5.3). On top of this, zero-shot prompting and few-shot prompting are used, adding a set of examples from zero to three, taken from (or similar to) the datasets with the statements created by hand. To evaluate which prompt is the best one, a series of examples has been prepared, taking the inputs from the datasets and creating the statements by hand (see Appendix 5.3). Each example is composed of the information provided to LLaMA-2-chat and one statement representing a good conversion of the information. LLaMA-2-chat, for each prompt, converts every information of the examples, generating the statements. The latter are compared with the statements of the examples in order to have a value representing how good the conversion is. For this purpose, both statements are embedded (more details in Section 4.2), and cosine similarity is calculated between the two. An average is then calculated between all the cosine similarities of the statements, resulting in the final score of the prompt. The prompt scoring higher is the best and will be used in the dataset conversion (see results in Section 5.1).

3.2.1 UMLS

The UMLS dataset is composed of triplets. LLaMA-2-chat has to convert each triplet into a statement. The 6 prompts prepared are evaluated on 15 examples of Triplets-Statement generation. The best prompt found on which UMLS is converted is reported in Figure 3.2 and it's a two-shot prompt.

3.2.2 MedMCQA

The MedMCQA dataset is composed of multiple-choice QA (four possible answers for each question). Each question has to be combined with one answer to generate a statement, so LLaMA-2-chat receives a question and an answer and has to turn them into a single statement. The 10 prompts prepared are evaluated on 20 examples of QA-Statement generation. The best prompt found on which MedMCQA is converted is reported in Figure 3.3 and it's a two-shot prompt.

```
SYSTEM: Your job is to transform the given triplet into a statement. The generated
statement must be correct.

USER: You must transform the given triplet into a statement. Keep all the words
received, don't change them.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."

Triplet: "Butilfenin", "subset includes concept", "FDA Established Names and Unique
Ingredient Identifier Codes Terminology"
Statement: "FDA Established Names and Unique Ingredient Identifier Codes Terminology
includes the concept of Butilfenin."

Triplet: {{new_triplet}}
Statement:
```

Figure 3.2: Best prompt found to convert UMLS.

```
SYSTEM: You must transform the question and the answer given into a statement. Just
make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence. If the question is preceded
by a sentence, repeat the sentence in the statement you create.

USER: You must transform the question and the answer given into a statement. Just
make a substitution, do not rephrase. Do not pay attention to the correctness of
the sentence. If the question is preceded by a sentence, repeat the sentence in the
statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."

Question: {{new_question}}
Answer: {{new_answer}}
Statement:
```

Figure 3.3: Best prompt found to convert MedMCQA.

3.3 Datasets Conversion

This section shows how the dataset conversions are made. Each dataset follows a different approach, given the distinct characteristics of each one.

3.3.1 UMLS

The UMLS dataset, composed of triplets, is converted with LLaMA-2-chat using the best prompt obtained after the prompt engineering. Each triplet generates two statements (see Figure 3.4), one labeled as correct, converting the

triplet as it is, and one labeled as incorrect, converting the triplet changing its head with a random one in a pool of heads taken from the dataset (there is the possibility of obtaining a false negative generation, but it's very unlikely). Due to time problems, the number of triplets converted has been 100k, resulting in a total of 200k generated statements.

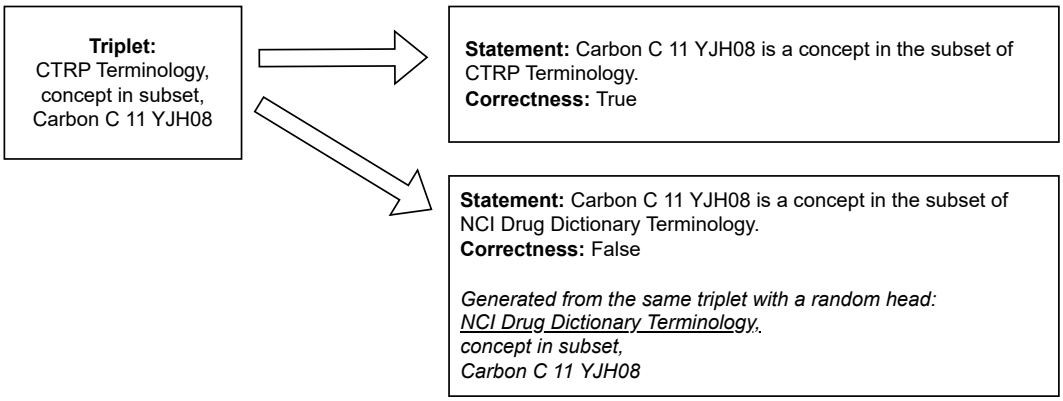


Figure 3.4: Example of a UMLS conversion.

3.3.2 BioASQ

The BioASQ dataset, composed of questions and complete answers, is converted by simply taking the complete answer as a statement labeled as correct, deleting in case a yes/no at the beginning if it's present (see Figure 3.5). The total of statements generated is 5046.

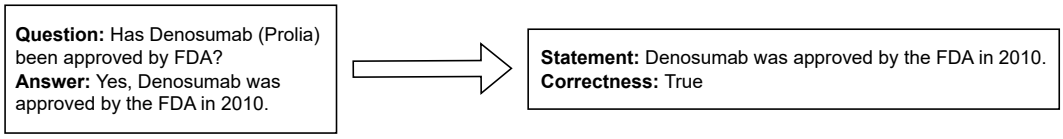


Figure 3.5: Example of a BioASQ conversion.

3.3.3 MedMCQA

The MedMCQA dataset, composed of multiple-choice QA of four answers to each question, is converted with LLaMA-2-chat using the best prompt obtained after the prompt engineering. Each single-answer question generates four statements, one labeled as correct, converting the question combined with the correct answer, and three labeled as incorrect, converting the question combined

with each of the three incorrect answers (see Figure 3.6). Each multiple-answer question generates one statement, labeled as correct, converting the question combined with the correct answer (choice made after encountering a problem in the dataset).[6] The total of statements generated is 557,748.

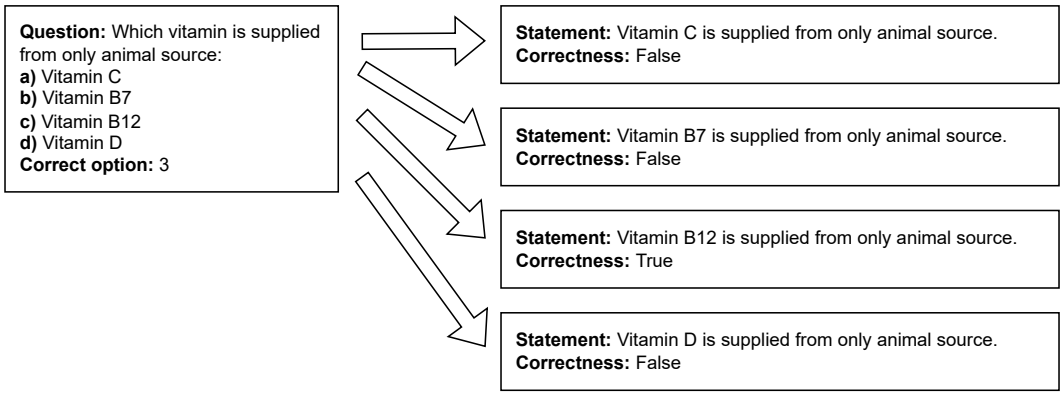


Figure 3.6: Example of a MedMCQA conversion.

3.3.4 PubMedQA

The PubMedQA dataset, divided into two parts, is treated differently for each one. As said in Section 3.1.4, the second part of the dataset is composed of questions and complete answers, the latter of which are directly taken as statements labeled as correct (similar to BioASQ, see Section 3.3.2). The first part instead, composed of questions and yes/no answers, is converted with LLaMA-2-chat using a simple prompt given the easy task. Each question and yes/no answer is converted into a statement. To obtain a 50/50 correct

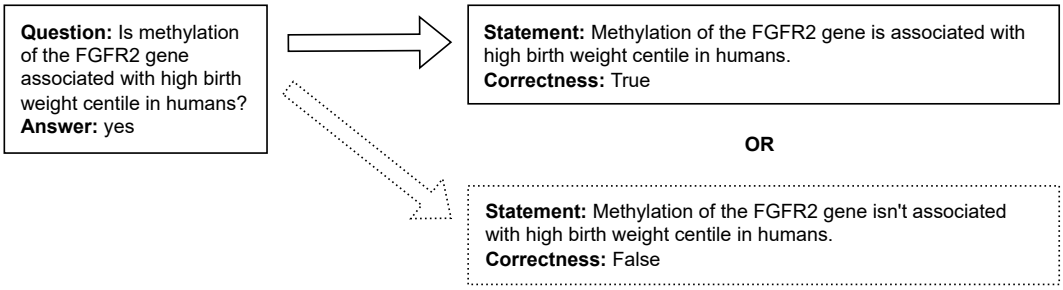


Figure 3.7: Example of a PubMedQA conversion.

and incorrect statements generation, the conversion is done as follows: half of

[6] See `https://github.com/medmcqa/medmcqa/issues/4` (last accessed on 27/02/2024)

the questions with yes answers and half of the questions with no answers are converted as they are, labeling them as correct, and the other two halves are converted as if the yes answers were no answers and vice versa, labeling them as incorrect (see Figure 3.7). The total of statements generated is 273,408.[7]

3.4 Classification Model

The MED-VERA dataset is designed to train evidence-free fact-checking models. For this reason, it is necessary to choose an appropriate classification model, which should correctly estimate the plausibility of a statement. There are different approaches to create an appropriate classification model:

- Using an encoder and adding a classification layer, either by taking an encoder-only model or an encoder from an encoder-decoder one. The encoder could be taken from a medical fine-tuned version of an existing model to optimize the performance, given that the medical field is the MED-VERA's domain of interest. For example, the encoder could be taken from:
  - PubMedBERT [20], which is an encoder-only model fine-tuned from BERT for the medical domain.
  - The Massive Text Embedding Benchmark (MTEB) Leaderboard [21], among the best encoders in the classification task.
  - SciFive [22], which is an encoder-decoder model fine-tuned from T5 for the medical domain.

  This type of model could predict a score of plausibility to classify the statements.
- Using an LLM decoder-only, treating the problem as a generative task, aiming to generate textual labels such as "Correct" and "Incorrect".

It's worth mentioning that Vera used both, taking the T5 encoder and taking LLaMA [15], which is a left-to-right decoder model. The results show that the Vera model built on top of T5 has a better performance than the one built on top of LLaMA. Therefore, the choice of the MED-VERA classification model choice could be oriented more towards the encoder option, also considering that, generally, encoders are more flexible and efficient.

[7]The statements generated are 110 fewer than the questions because these were ignored, having the answer "maybe".

Chapter 4

Experimental Setup

This chapter contains information about the hardware used during the work of MED-VERA and details about implementations, showing the libraries and the tools used.

4.1 Hardware

The experiments and the prompt engineering runs are performed using Google Colab,[1] a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs, and is especially well suited to machine learning, data science, and education. Specifically, the GPU used is the Tesla T4, with 16 GB of GDDR6 memory and 2560 CUDA cores.

The dataset conversions involving inference are performed using the University servers with Slurm [23] on an NVIDIA GeForce RTX 3090 GPU, which has 24 GB of GDDR6X memory and 10,496 CUDA cores.

4.2 Implementation Details

In the prompt engineering part, developed on Google Colab, initially, the tests were done using the llama-cpp-python library[2] and the model used was a quantized version of LLaMA-2-chat with 13B parameters[3]. Subsequently, the vLLM library [24] is used for consistency (see Paragraph 4.2), and the

[1] `https://colab.google/`

[2] `https://github.com/abetlen/llama-cpp-python`

[3] file llama-2-13b-chat.Q4_K_M.gguf in `https://huggingface.co/TheBloke/Llama-2-13B-chat-GGUF`

model used is a quantized version of LLaMa-2-chat with 13B parameters.[4] To embed the statements to calculate the cosine similarity (as explained in Section 3.2) Universal AnglE Embedding [25] is used, which was found in the MTEB Leaderboard at the top places in the category Semantic Textual Similarity (STS). The prompt scores are then tracked using `Weights & Biases` (`W&B`) [26]. An approximate code for the prompt engineering is:

```
# ...

# Run the prompts on the examples
for prompt in prompts:
    prompt_similarity = 0
    for input in statements:
        # generate the statement
        prompt_template = f'''{prompt}

{input}
Statement:'''
        response = llm.generate(prompt_template, sampling_params) #
            either with vLLM or llama-cpp-python

        # get the generated statement
        statement = response[2:].split('"')[0] # make sure to extract
            the actual response based on the chosen library

        # embed the generated statement and the expected one
        vecGenerated = angle.encode(statement, to_numpy=False)
        vecExpected = angle.encode(statements[input], to_numpy=False)

        # calculate cosine similarity
        max_length = max(len(vecGenerated), len(vecExpected))
        vecGenerated = torch.cat([vecGenerated, torch.zeros(max_length
            - len(vecGenerated)).to('cuda:0')]) # to obtain the same
            length
        vecExpected = torch.cat([vecExpected, torch.zeros(max_length -
            len(vecExpected)).to('cuda:0')]) # to obtain the same
            length
        vecGenerated = vecGenerated / vecGenerated.norm(2) # normalize
        vecExpected = vecExpected / vecExpected.norm(2) # normalize
        similarity =
            torch.nn.functional.cosine_similarity(vecGenerated,
            vecExpected)
```

[4] https://huggingface.co/TheBloke/Llama-2-13B-chat-AWQ

```
        prompt_similarity += similarity.item()

    # get the average similarity
    prompt_similarity /= len(statements)

    # log the prompt similarity on wandb
    wandb.log({"similarity": prompt_similarity})

# ...
```

For the dataset conversions, the runs are performed in Docker containers [27]. The vLLM library is used to speed up the inference times, and the model used is the one mentioned above. The LLM configuration and the sampling parameters used with vLLM are:

```
sampling_params = SamplingParams(
    n=1,
    temperature=0.0,
    max_tokens=256,
    use_beam_search=False
)

llm = LLM(
    model="TheBloke/Llama-2-13B-chat-AWQ",
    quantization='awq',
    dtype='half',
    gpu_memory_utilization=.95
)
```

The docker file with the correct library versions is:

```
FROM nvidia/cuda:11.8.0-devel-ubuntu20.04
LABEL maintainer="disi-unibo-nlp"

# Zero interaction (default answers to all questions)
ENV DEBIAN_FRONTEND=noninteractive

# Set work directory
WORKDIR /workspace

# Install general-purpose dependencies
RUN apt-get update -y && \
    apt-get install -y curl \
```

```
    git \
    bash \
    nano \
    wget \
    python3.8 \
    python3-pip && \
    apt-get autoremove -y && \
    apt-get clean -y && \
    rm -rf /var/lib/apt/lists/*
RUN pip install --upgrade pip
RUN pip install wrapt --upgrade --ignore-installed
RUN pip install gdown

# Install Torchvision
RUN pip install torchvision==0.16.1+cu118 -f
    https://download.pytorch.org/whl/cu118/torch_stable.html

# Install other Python packages
RUN pip3 install --upgrade transformers==4.35.0
RUN pip3 install --upgrade peft==0.5.0
RUN pip3 install --upgrade datasets==2.14.5
RUN pip3 install --upgrade wandb==0.15.7
RUN pip3 install --upgrade tokenizers==0.14.1
RUN pip3 install --upgrade tqdm==4.63.1
RUN pip3 install --upgrade nltk==3.7
RUN pip3 install --upgrade scipy==1.10.1
RUN pip3 install --upgrade huggingface_hub==0.17.3

# required for flash attention
RUN pip3 install -q accelerate==0.24.1 optimum==1.14.1
RUN pip3 install packaging==23.2
RUN pip3 install ninja==1.11.1.1
RUN pip3 install flash-attn==2.3.3 --no-build-isolation
RUN pip3 install bitsandbytes==0.41.2.post2

# Install vLLM
RUN pip3 install vllm==0.2.2

# Install PyTorch and other necessary packages
RUN pip3 install torch==2.1.0
RUN pip3 install protobuf==4.25.1
RUN pip3 install safetensors==0.4.0
RUN pip3 install sentencepiece==0.1.99
RUN pip3 install xformers==0.0.22.post7
```

```
# Back to default frontend
ENV DEBIAN_FRONTEND=dialog
```

Chapter 5

Results

This chapter presents the results obtained in the work of MED-VERA, showing the prompt engineering results and the MED-VERA generated dataset.

5.1 Prompt Engineering Results

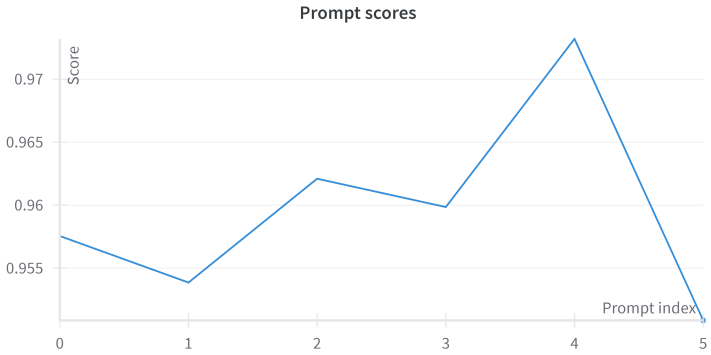
The `W&B` trackings of the prompt engineering of UMLS and MedMCQA report the results presented in Figure 5.1. The prompt scores are all above 0.940, indicating a great conversion quality, and the best ones are 0.9732 for UMLS and 0.9685 for MedMCQA. The corresponding best prompts are reported in Section 3.2.1 and Section 3.2.2, respectively.

5.2 Generated Dataset

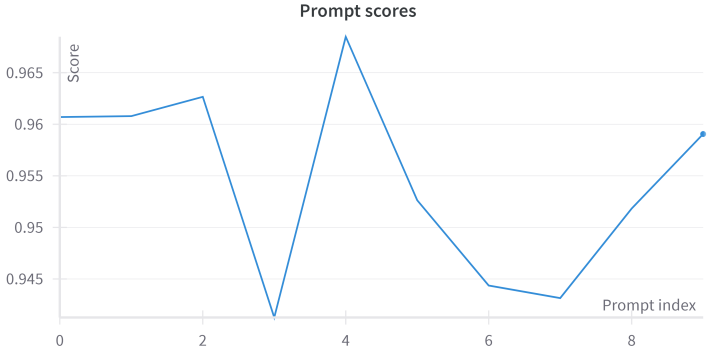
The MED-VERA generated dataset presents the characteristics reported in Table 5.1. A total of 1,036,202 statements have been generated, with a 44.33%

Dataset	**# Questions / Triplets**	**# Statements generated**	**# Correct statements**	**# Incorrect statements**
UMLS	100,000	200,000	100,000	100,000
BioASQ	5,046	5,046	5,046	0
MedMCQA	187,005	557,748	187,005	370,743
PubMedQA	273,518	273,408	167,328	106,080
TOTAL	565,569	1,036,202	459,379	576,823

Table 5.1: MED-VERA dataset characteristics. The number of statements generated (correct, incorrect, and total) is shown along with the number of initial questions and triplets, all differentiated by dataset.



(a) Prompt scores obtained in the UMLS prompt engineering. The best prompt is the Prompt #4 with a score of 0.9732.



(b) Prompt scores obtained in the MedMCQA prompt engineering. The best prompt is the Prompt #4 with a score of 0.9685.

Figure 5.1: Prompt scores obtained in the prompt engineering part.

of correct statements and a 55.67% of incorrect ones. Some examples of the statements generated are reported in Table 5.2.

However, it must be mentioned that LLaMA-2-chat, given that LLMs are not infallible, has produced some statements that are not entirely accurate. Some conversions are critical, like patient stories. Figure 5.2 shows a successful statement generation, while Figure 5.3 shows two unsuccessful ones, also showing how the same question is converted differently (the first statement is clearly better than the second one). Another example of an unsuccessful statement generation can be seen in Figure 5.4 (the correct statement generation should be to similar to "Cyclobenzaprine hydrochloride 5 MG Oral Tablet has the trade name Flexeril.").

	Statement	Correctness
UMLS	MgCl2 is an ingredient of Doxy Lemmon Pill.	False
BioASQ	The protein Pannexin1 is localized to the plasma membranes.	True
MedMCQA	Kocher's incision is a muscle splitting incision.	False
PubMedQA	Interleukin-1 facilitates airway epithelial migration in response to injury.	True

Table 5.2: Some examples of the labeled statements present in the MED-VERA dataset, with the indication of the source dataset.

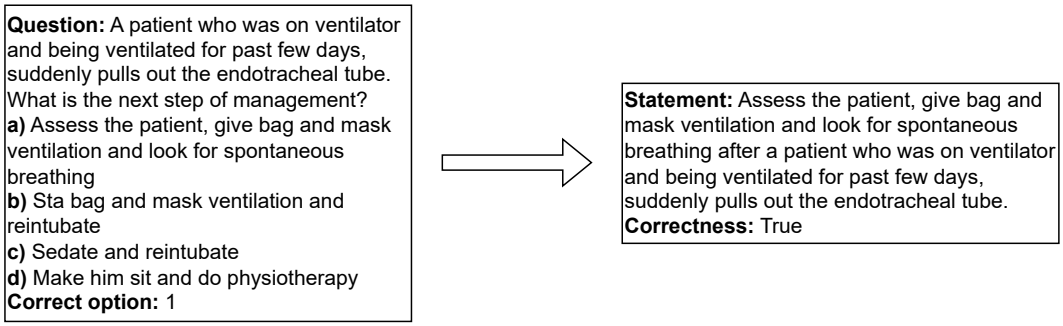


Figure 5.2: Example of a successful statement generation of a patient-story question.

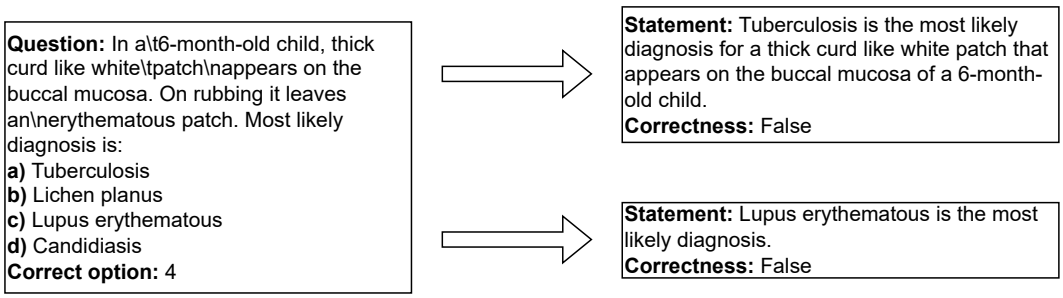


Figure 5.3: Examples of unsuccessful statement generations of a patient-story question.

5.3 Execution times

This section details the execution times. Specifically, the PubMedQA conversion is taken as an example to present them. The conversion, as mentioned in Section 4.1, has been performed using the University servers. The results show that

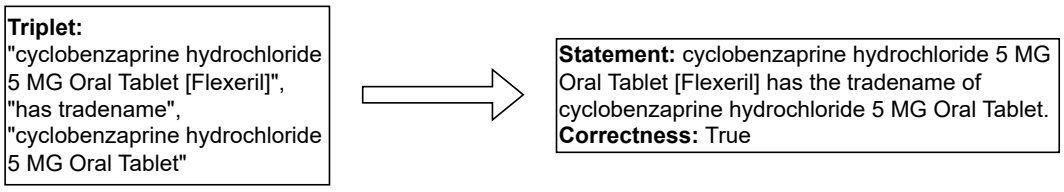


Figure 5.4: Example of an unsuccessful statement generation of a triplet.

the conversion of one QA into one statement takes about 0.73 seconds using the vLLM library and 8.96 seconds without it (i.e. using the llama-cpp-python library). Therefore, converting the datasets with vLLM provided a tremendous time-saving.

Conclusions and Future Work

Med-vera started as a work to partially make up for the lack of datasets for evidence-free fact-checking models. Med-vera has successfully generated an artificial medical dataset of over 1 million statements labeled as correct or incorrect, created from the most used medical resources. We thus believe that the initial goal has been achieved, and we look forward to the possibility of continuing this work. With more time, Med-vera could convert the remaining 20M triplets of UMLS, enriching the dataset. The conversions could also continue with other datasets, specifically QA ones, given that the generalized method used could be applied to them as well, allowing the dataset to become even richer. We also look forward to using the Med-vera dataset to train evidence-free fact-checking models. We plan to train the Vera model on the dataset to get a clear answer on the real performance of the dataset, and after that, we might even create our own medical evidence-free fact-checking model. Another direction could be a shift towards biomedical graphs [28, 29].

In conclusion, we think that Med-vera represents a step forward in the field of medical evidence-free fact-checking, providing a new large artificial dataset to train these types of models. As future work, we will focus on enriching the dataset and using it to train medical free-evidence fact-checking models.

Acknowledgements

I would like to thank Professor Gianluca Moro for giving me the opportunity to realize this project, from which I've learned a lot. I also want to thank Dr. Giacomo Frisoni for guiding me through this project, for his kindness and willingness to help, always ready to answer any questions or clarifications.

Last but not least, I thank my family and friends for always being there for me.

Bibliography

[1] Ali Borji. A categorical archive of chatgpt failures. *CoRR*, abs/2302.03494, 2023. doi: 10.48550/ARXIV.2302.03494. URL https://doi.org/10.48550/arXiv.2302.03494.

[2] Alessandro Bruno, Pier Luigi Mazzeo, Aladine Chetouani, Marouane Tliba, and Mohamed Amine Kerkouri. Insights into classifying and mitigating llms' hallucinations. In Alessandro Bruno, Arianna Pipitone, Riccardo Manzotti, Agnese Augello, Pier Luigi Mazzeo, Filippo Vella, and Antonio Chella, editors, *Proceedings of the 1st Workshop on Artificial Intelligence for Perception and Artificial Consciousness (AIxPAC 2023) co-located with the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023), Roma, Italy, November 8, 2023*, volume 3563 of *CEUR Workshop Proceedings*, pages 50–63. CEUR-WS.org, 2023. URL https://ceur-ws.org/Vol-3563/paper_12.pdf.

[3] Juraj Vladika and Florian Matthes. Scientific fact-checking: A survey of resources and approaches. *CoRR*, abs/2305.16859, 2023. doi: 10.48550/ARXIV.2305.16859. URL https://doi.org/10.48550/arXiv.2305.16859.

[4] Valentin Liévin, Christoffer Egeberg Hother, and Ole Winther. Can large language models reason about medical questions? *CoRR*, abs/2207.08143, 2022. doi: 10.48550/ARXIV.2207.08143. URL https://doi.org/10.48550/arXiv.2207.08143.

[5] Oded Nov, Nina Singh, and Devin M. Mann. Putting chatgpt's medical advice to the (turing) test. *CoRR*, abs/2301.10035, 2023. doi: 10.48550/ARXIV.2301.10035. URL https://doi.org/10.48550/arXiv.2301.10035.

[6] Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Vera: A general-purpose plausibility estimation model for commonsense statements. *CoRR*, abs/2305.03695, 2023. doi:

10.48550/ARXIV.2305.03695. URL https://doi.org/10.48550/arXiv.2305.03695.

[7] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. Generating scientific claims for zero-shot scientific fact checking. *CoRR*, abs/2203.12990, 2022. doi: 10.48550/ARXIV.2203.12990. URL https://doi.org/10.48550/arXiv.2203.12990.

[8] Neset Tan, Trung Nguyen, Joshua Bensemann, Alex Yuxuan Peng, Qiming Bao, Yang Chen, Mark Gahegan, and Michael Witbrock. Multi2claim: Generating scientific claims from multi-choice questions for scientific fact-checking. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2644–2656. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.194. URL https://doi.org/10.18653/v1/2023.eacl-main.194.

[9] 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. URL https://doi.org/10.1145/3490238.

[10] Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database-Issue): 267–270, 2004. doi: 10.1093/NAR/GKH061. URL https://doi.org/10.1093/nar/gkh061.

[11] Anastasios Nentidis, Georgios Katsimpras, Eirini Vandorou, Anastasia Krithara, Antonio Miranda-Escalada, Luis Gascó, Martin Krallinger, and Georgios Paliouras. Overview of bioasq 2022: The tenth bioasq challenge on large-scale biomedical semantic indexing and question answering. *CoRR*, abs/2210.06852, 2022. doi: 10.48550/ARXIV.2210.06852. URL https://doi.org/10.48550/arXiv.2210.06852.

[12] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *CoRR*, abs/2203.14371, 2022. doi: 10.48550/ARXIV.2203.14371. URL https://doi.org/10.48550/arXiv.2203.14371.

[13] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *CoRR*, abs/1909.06146, 2019. URL `http://arxiv.org/abs/1909.06146`.

[14] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL `https://doi.org/10.48550/arXiv.2307.09288`.

[15] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL `https://doi.org/10.48550/arXiv.2302.13971`.

[16] 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. *CoRR*, abs/1910.10683, 2019. URL `http://arxiv.org/abs/1910.10683`.

[17] Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. Evidence-based fact-checking of health-related claims. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3499–3512. Association for Computational Lin-

guistics, 2021. doi: 10.18653/V1/2021.FINDINGS-EMNLP.297. URL https://doi.org/10.18653/v1/2021.findings-emnlp.297.

[18] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.703. URL https://doi.org/10.18653/v1/2020.acl-main.703.

[19] Hyeryun Park, Jiye Son, Jeongwon Min, and Jinwook Choi. Selective umls knowledge infusion for biomedical question answering. *Scientific Reports*, 13(1):14214, 2023.

[20] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779, 2020. URL https://arxiv.org/abs/2007.15779.

[21] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: massive text embedding benchmark. *CoRR*, abs/2210.07316, 2022. doi: 10.48550/ARXIV.2210.07316. URL https://doi.org/10.48550/arXiv.2210.07316.

[22] Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. Scifive: a text-to-text transformer model for biomedical literature. *CoRR*, abs/2106.03598, 2021. URL https://arxiv.org/abs/2106.03598.

[23] Andy B. Yoo, Morris A. Jette, and Mark Grondona. SLURM: simple linux utility for resource management. In Dror G. Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing, 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003, Revised Papers*, volume 2862 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2003. doi: 10.1007/10968987_3. URL https://doi.org/10.1007/10968987_3.

[24] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient

memory management for large language model serving with pagedattention. *CoRR*, abs/2309.06180, 2023. doi: 10.48550/ARXIV.2309.06180. URL https://doi.org/10.48550/arXiv.2309.06180.

[25] Xianming Li and Jing Li. Angle-optimized text embeddings. *CoRR*, abs/2309.12871, 2023. doi: 10.48550/ARXIV.2309.12871. URL https://doi.org/10.48550/arXiv.2309.12871.

[26] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

[27] Dirk Merkel et al. Docker: lightweight linux containers for consistent development and deployment. *Linux j*, 239(2):2, 2014.

[28] Giacomo Frisoni, Gianluca Moro, and Lorenzo Balzani. Text-to-text extraction and verbalization of biomedical event graphs. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2692–2710. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.238.

[29] Giacomo Frisoni, Gianluca Moro, Giulio Carlassare, and Antonella Carbonaro. Unsupervised event graph representation and similarity learning on biomedical literature. *Sensors*, 22(1):3, 2022. doi: 10.3390/S22010003. URL https://doi.org/10.3390/s22010003.

Appendix

This appendix contains all the prompts and conversion examples used in the prompt engineering for both UMLS and MedMCQA. The code presented can be used along with the prompt engineering code reported in Paragraph 4.2.

Prompts

All the prompts evaluated in the prompt engineering for UMLS are reported below.

```
prompts = {
    '''SYSTEM: Your job is to transform the given triplet into a
       statement. The generated statement must make sense.

USER: You must transform the given triplet into a statement.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."''',


    '''SYSTEM: Your job is to transform the given triplet into a
       statement. The generated statement must make sense.

USER: You must transform the given triplet into a statement.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."

Triplet: "Butilfenin", "subset includes concept", "FDA Established
   Names and Unique Ingredient Identifier Codes Terminology"
Statement: "FDA Established Names and Unique Ingredient Identifier
   Codes Terminology includes the concept of Butilfenin."''',


    '''SYSTEM: Your job is to transform the given triplet into a
       statement. The generated statement must make sense.
```

```
USER: You must transform the given triplet into a statement.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."

Triplet: "Butilfenin", "subset includes concept", "FDA Established
    Names and Unique Ingredient Identifier Codes Terminology"
Statement: "FDA Established Names and Unique Ingredient Identifier
    Codes Terminology includes the concept of Butilfenin."

Triplet: "tyrosine", "inverse isa", "amino acid"
Statement: "Tyrosine is an amino acid."''',


    '''SYSTEM: Your job is to transform the given triplet into a
        statement. The generated statement must be correct.

USER: You must transform the given triplet into a statement. Keep all
    the words received, don't change them.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."''',


    '''SYSTEM: Your job is to transform the given triplet into a
        statement. The generated statement must be correct.

USER: You must transform the given triplet into a statement. Keep all
    the words received, don't change them.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."

Triplet: "Butilfenin", "subset includes concept", "FDA Established
    Names and Unique Ingredient Identifier Codes Terminology"
Statement: "FDA Established Names and Unique Ingredient Identifier
    Codes Terminology includes the concept of Butilfenin."''',


    '''SYSTEM: Your job is to transform the given triplet into a
        statement. The generated statement must be correct.

USER: You must transform the given triplet into a statement. Keep all
    the words received, don't change them.
Triplet: "amino acid", "is", "lysine"
Statement: "Lysine is an amino acid."
```

```
Triplet: "Butilfenin", "subset includes concept", "FDA Established
    Names and Unique Ingredient Identifier Codes Terminology"
Statement: "FDA Established Names and Unique Ingredient Identifier
    Codes Terminology includes the concept of Butilfenin."

Triplet: "tyrosine", "inverse isa", "amino acid"
Statement: "Tyrosine is an amino acid."'''
}
```

All the prompts evaluated in the prompt engineering for MedMCQA are reported below.

```
prompts = {
  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence. If the
    question is preceded by a sentence, repeat the sentence in the
    statement you create.

USER: You must transform the question and the answer given into a
    statement. Just make a substitution, do not rephrase. Do not pay
    attention to the
correctness of the sentence. If the question is preceded by a
    sentence, repeat the sentence in the statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."''',


  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement. Just make a substitution, do not rephrase. Do not pay
    attention to the
correctness of the sentence. If the question is preceded by a
    sentence, repeat the sentence in the statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."''',
```

```
  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement.
If the question is preceded by a sentence, repeat the sentence in the
    statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."''',


  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement. Each piece of the question must be kept in the
    statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."''',


'''SYSTEM: You must transform the question and the answer given into
    a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence. If the
    question is preceded by a sentence, repeat the sentence in the
    statement you create.

USER: You must transform the question and the answer given into a
    statement. Just make a substitution, do not rephrase. Do not pay
    attention to the
correctness of the sentence. If the question is preceded by a
    sentence, repeat the sentence in the statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."''',
```

```
  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement. Just make a substitution, do not rephrase. Do not pay
    attention to the
correctness of the sentence. If the question is preceded by a
    sentence, repeat the sentence in the statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."''',


  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement.
If the question is preceded by a sentence, repeat the sentence in the
    statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."''',


  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement. Each piece of the question must be kept in the
    statement you create.
Question: "Which vitamin is supplied from only animal source:"
```

```
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."''',


'''SYSTEM: You must transform the question and the answer given into
    a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence. If the
    question is preceded by a sentence, repeat the sentence in the
    statement you create.

USER: You must transform the question and the answer given into a
    statement. Just make a substitution, do not rephrase. Do not pay
    attention to the
correctness of the sentence. If the question is preceded by a
    sentence, repeat the sentence in the statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."

Question: "A 47-year-old man suddenly develops high fever and
    hypotension. He has a generalized erythematous macular rash, and
    over the next day, develops gangrene of his left leg. Which of
    the following is the most likely organism?"
Answer: "Streptococcus group C"
Statement: "Streptococcus group C is the most likely organism in a
    47-year-old man that suddenly develops high fever and
    hypotension, has a generalized erythematous macular rash, and
    over the next day, develops gangrene of his left leg."''',


  '''SYSTEM: You must transform the question and the answer given
      into a statement. Just make a substitution, do not rephrase.
Do not pay attention to the correctness of the sentence.

USER: You must transform the question and the answer given into a
    statement. Just make a substitution, do not rephrase. Do not pay
```

```
    attention to the
correctness of the sentence. If the question is preceded by a
    sentence, repeat the sentence in the statement you create.
Question: "Which vitamin is supplied from only animal source:"
Answer: "Vitamin C"
Statement: "Vitamin C is supplied from only animal source."

Question: "Egg shell calcification is seen in all except"
Answer: "Sarcoidosis"
Statement: "Egg shell calcification is not seen in sarcoidosis."

Question: "A 47-year-old man suddenly develops high fever and
    hypotension. He has a generalized erythematous macular rash, and
    over the next day, develops gangrene of his left leg. Which of
    the following is the most likely organism?"
Answer: "Streptococcus group C"
Statement: Streptococcus group C is the most likely organism in a
    47-year-old man that suddenly develops high fever and
    hypotension, has a generalized erythematous macular rash, and
    over the next day, develops gangrene of his left leg."'''
}
```

Conversion examples

All the conversion examples used to evaluate the UMLS prompts in the prompt engineering are reported below.

```
statements = {
    ("Multisection^WO contrast:Find:Pt:Lower extremity.left:Doc:CT",
        "is presence of lateral location", "True")
    : "Multisection^WO contrast:Find:Pt:Lower extremity.left:Doc:CT
        indicates presence of lateral location.",

    ("Mature B-Cell Neoplasm", "is not abnormal cell of disease",
        "Neoplastic T-Lymphocyte and Neoplastic Natural Killer Cell")
    : "Neoplastic T-Lymphocyte and Neoplastic Natural Killer Cell are
        not abnormal cell of Mature B-Cell Neoplasm.",

    ("Laboratory", "has class", "s100 calcium binding protein b
        [mass/vol]")
    : "Laboratory has class 's100 calcium binding protein b
        [mass/vol]'",
```

```
("Laboratory", "has class", "Follitropin & Lutropin
    panel:ACnc:Pt:Ser/Plas:Qn")
: "Laboratory has class: 'Follitropin & Lutropin
    panel:ACnc:Pt:Ser/Plas:Qn'",

("magnesium chloride Oral Solution", "ingredient of", "MgCl2")
: "MgCl2 is an ingredient of magnesium chloride Oral Solution.",

("Ethinyl Estradiol 0.035 MG / Norethindrone 0.5 MG Oral Tablet",
    "active ingredient of", "ethinyl estradiol")
: "Ethinyl estradiol is the active ingredient of Ethinyl
    Estradiol 0.035 MG / Norethindrone 0.5 MG Oral Tablet.",

("Abnormal ossification of the trapezoid bone", "inverse isa",
    "Abnormality of carpal bone ossification")
: "Abnormal ossification of the trapezoid bone is an abnormality
    of carpal bone ossification.",

("Immunoglobulin Constant Regions", "gene product has structural
    domain or motif", "immunoglobulin heavy constant gamma 1")
: "The immunoglobulin heavy constant gamma 1 gene product has
    structural domain or motif.",

("AVOBENZONE 30 mg in 1 g / HOMOSALATE 80 mg in 1 g / OCTISALATE
    40 mg in 1 g / OCTOCRYLENE 80 mg in 1 g / OXYBENZONE 50 mg in
    1 g TOPICAL AEROSOL, SPRAY [CVS Pharmacy SPF 30 Wet and Dry]",
    "inactive ingredient of", "trimethylpentanediol-adipic
    acid-glycerin crosspolymer (25000 MPA.S)")
: "Trimethylpentanediol-adipic acid-glycerin crosspolymer (25000
    MPA.S) is an inactive ingredient of AVOBENZONE 30 mg in 1 g /
    HOMOSALATE 80 mg in 1 g / OCTISALATE 40 mg in 1 g /
    OCTOCRYLENE 80 mg in 1 g / OXYBENZONE 50 mg in 1 g TOPICAL
    AEROSOL, SPRAY [CVS Pharmacy SPF 30 Wet and Dry].",

("Bifidobacterium", "subset includes concept", "Clinical Data
    Interchange Standards Consortium Terminology")
: "Clinical Data Interchange Standards Consortium Terminology
    includes the concept of Bifidobacterium.",

("cyclobenzaprine hydrochloride 5 MG Oral Tablet [Flexeril]",
    "has tradename", "cyclobenzaprine hydrochloride 5 MG Oral
    Tablet")
: "Cyclobenzaprine hydrochloride 5 MG Oral Tablet has the trade
```

```
        name Flexeril.",

    ("CTRP Terminology", "concept in subset", "Carbon C 11 YJH08")
    : "Carbon C 11 YJH08 is a concept in the subset of CTRP
        Terminology.",

    ("Cortinarius coalescens", "mapped from", "Cortinarius")
    : "Cortinarius coalescens is mapped from Cortinarius.",

    ("Sphenocavernous Meningioma", "may be molecular abnormality of
        disease", "PIK3CA Gene Mutation")
    : "PIK3CA Gene Mutation may be a molecular abnormality of the
        Sphenocavernous Meningioma.",

    ("HLA-A*23 antigen", "mapped from", "HLA-A Antigens")
    : "The HLA-A*23 antigen is mapped from HLA-A antigens."
}
```

All the conversion examples used to evaluate the MedMCQA prompts in the prompt engineering are reported below.

```
statements = {
    "Question: \"Growth hormone has its effect on growth
        through?\"\nAnswer: \"Intranuclear receptors\""
    : "Growth hormone has its effect on growth through intranuclear
        receptors.",

    "Question: \"Growth hormone has its effect on growth
        through?\"\nAnswer: \"Directly\""
    : "Growth hormone directly affects growth.",

    "Question: \"All of the following are surgical options for morbid
        obesity except -\"\nAnswer: \"Adjustable gastric banding\""
    : "Adjustable gastric banding is not a surgical option for morbid
        obesity.",

    "Question: \"Chronic urethral obstruction due to benign prismatic
        hyperplasia can lead to the following change in kidney
        parenchyma\"\nAnswer: \"Dyplasia\""
    : "Chronic urethral obstruction due to benign prismatic
        hyperplasia can lead to dyplastic changes in the kidney
        parenchyma.",

    "Question: \"A 60 yr old chronic smoker presents with painless
```

```
    gross hematuria of 1 day duration. Investigation of choice to
    know the cause of hematuria\"\nAnswer: \"X-ray KUB\""
: "A 60 yr old chronic smoker presents with painless gross
    hematuria of 1 day duration. Investigation of choice to know
    the cause of hematuria is X-ray KUB.",

"Question: \"Following endaerectomy on the right common carotid,
    a patient is found to be blind in the right eye. It is appears
    that a small thrombus embolized during surgery and lodged in
    the aery supplying the optic nerve. Which aery would be
    blocked?\"\nAnswer: \"Central aery of the retina\""
: "Following endaerectomy on the right common carotid, a patient
    is found to be blind in the right eye. It is appears that a
    small thrombus embolized during surgery and lodged in the aery
    supplying the optic nerve. The central aery of the retina is
    blocked.",

"Question: \"A 6hours old snake bite patient comes to emergency
    with mild local edema at the injury site. On examination no
    abnormalities detected and lab repos are normal. Most
    appropriate management is\"\nAnswer: \"Incision and suction\""
: "A 6hours old snake bite patient comes to emergency with mild
    local edema at the injury site. On examination no
    abnormalities detected and lab repos are normal. Most
    appropriate management is incision and suction.",

"Question: \"Which of the following agents is most commonly
    associated with recurrent meningitis due to CSF
    leaks?\"\nAnswer: \"Meningococci\""
: "Meningococci is the agent most commonly associated with
    recurrent meningitis due to CSF leaks.",

"Question: \"Superior vena cava is derived from:\"\nAnswer:
    \"Aortic arch\""
: "Superior vena cava is derived from the aortic arch.",

"Question: \"With which of the following receptors theophylline
    has an antagonistic interaction ?\"\nAnswer: \"Histamine
    receptors\""
: "Theophylline has an antagonistic interaction with histamine
    receptors.",

"Question: \"For a positively skewed curve which measure of
    central tendency is largest\"\nAnswer: \"Median\""
```

```
: "The median is the largest measure of central tendency for a
    positively skewed curve.",

"Question: \"The process of hardening a cement matrix through
    hydration with oral fluids to achieve greater mechanical
    strength is known as:\"\nAnswer: \"Mineralization\""
: "Mineralization is the process of hardening a cement matrix
    through hydration with oral fluids to achieve greater
    mechanical strength.",

"Question: \"Lymph vessel which drain the posterior 1\/3 rd of
    the tongue:\"\nAnswer: \"Central vessel.\""
: "The central vessel is the lymph vessel which drain the
    posterior 1\/3 rd of the tongue.",

"Question: \"Risk factors associated with post-operative nausea
    and vomiting following strabismus surgery are all except
    -\"\nAnswer: \"Duration of anesthesia > 30 mins\""
: "Duration of anesthesia > 30 mins is not a risk factor
    associated with post-operative nausea and vomiting following
    strabismus surgery.",

"Question: \"All are True about Acute Osteomyelitis
    except\"\nAnswer: \"Common in children\""
: "Acute Osteomyelitis is not common in children.",

"Question: \"All are True about Acute Osteomyelitis
    except\"\nAnswer: \"Severe pain\""
: "Acute Osteomyelitis is not characterized by severe pain.",

"Question: \"Anterolateral ahroscopy of knee is for:\"\nAnswer:
    \"To see the posterior cruciate ligament\""
: "Anterolateral ahroscopy of knee is performed to see the
    posterior cruciate ligament.",

"Question: \"25 year old patient Suspected to have a
    pneumoperitoneum. Patient is unable to stand. Best x-ray view
    is\"\nAnswer: \"Right lateral decubitus view\""
: "25 year old patient Suspected to have a pneumoperitoneum.
    Patient is unable to stand. Best x-ray view is the right
    lateral decubitus.",

"Question: \"What change will be seen in vertebral column in
    ochronosis-\"\nAnswer: \"Increased disc space\""
```

```
    : "Increased disc space will be seen in vertebral column in
       ochronosis.",

    "Question: \"Which vitamin is supplied from only animal
       source:\"\nAnswer: \"Vitamin C\""
    : "Vitamin C is supplied from only animal source."
}
```