



SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Data Engineering and Analytics

**Step-By-Step Claim Verification Using LLMs
and Knowledge Graphs**

Ivana Hacajová



SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Data Engineering and Analytics

Step-By-Step Claim Verification Using LLMs and Knowledge Graphs

Schrittweise Überprüfung von Behauptungen mit LLMs und Wissensgraphen

Author: Ivana Hacajová
Supervisor: Prof. Dr. Florian Matthes
Advisor: Juraj Vladika, MSc.
Submission Date: June 17th 2024

I confirm that this master's thesis in data engineering and analytics is my own work and I have documented all sources and material used.

Location, Submission Date

Author

AI Assistant Usage Disclosure

Introduction

Performing work or conducting research at the Chair of Software Engineering for Business Information Systems (sebis) at TUM often entails dynamic and multi-faceted tasks. At sebis, we promote the responsible use of *AI Assistants* in the effective and efficient completion of such work. However, in the spirit of ethical and transparent research, we require all student researchers working with sebis to disclose their usage of such assistants.

For examples of correct and incorrect AI Assistant usage, please refer to the original, unabridged version of this form, located at this link.

Use of *AI Assistants* for Research Purposes

I have used AI Assistant(s) for the purposes of my research as part of this thesis.

Yes No

Explanation: After writing the Introduction, Conclusion, Abstract and Acknowledgements myself, I used ChatGPT to improve those sections in the following areas: language and style enhancement, clarity and conciseness. Then I used it to translate the abstract to the German language.

Additionally, I used ChatGPT for transforming some citation strings into bibtex entries.

Finally, I used Grammarly's browser extension to identify mistakes, typos or unclear sentences in the whole work. The license for Grammarly was provided by TUM.

I confirm in signing below, that I have reported all usage of AI Assistants for my research, and that the report is truthful and complete.

Location, Date

Author

Acknowledgments

I would like to express my deepest gratitude to Prof. Dr. Florian Matthes for supervising this thesis. I also extend my sincere thanks to Juraj Vladika, MSc., for being my advisor and providing valuable feedback and guidance during the entire duration of this work.

I am profoundly grateful to my family, my partner David, and my friends for their unwavering support not only during the completion of this thesis but throughout my entire course of studies at the Technical University of Munich.

Additionally, I would like to acknowledge the researchers who have laid the foundation for automated fact-checking. Their pioneering work in this crucial area has enabled me to contribute to an important cause with global significance.

Finally, I extend my heartfelt thanks to the state of Germany and its taxpayers for providing me the opportunity and support to study at such an excellent university.

Abstract

The spread of fake news and misinformation has become a significant global issue. Fact-checking, or claim verification, is one of the primary methods to counteract these effects. Recent advancements in natural language processing (NLP) have enabled researchers to explore automating this process. This thesis leverages the latest large language models (LLMs) for explainable, open-domain, knowledge-grounded claim verification.

We develop three modular claim verification pipelines utilizing the in-context learning capabilities of LLMs, specifically GPT-3.5. These pipelines decompose claims into subquestions, answer each subquestion individually, and perform final reasoning over the gathered evidence. The base pipeline employs reasoning over natural language, the predicate pipeline over symbolic representations of claims, and the knowledge graph (KG) pipeline reasons over evidence from DBPedia. We investigate various evidence sources for knowledge-grounded reasoning.

Our findings indicate that these pipelines outperform many established baselines (transformer based or LLM based) for general multi-hop claims, as well as real-life claims in specific domains such as medicine, climate change, and COVID-19. Furthermore, both quantitative results and a user survey demonstrated that using Mixtral-8x7b as the reasoning LLM significantly improved performance, more so than the choice of evidence source, whether knowledge-grounded or otherwise.

Kurzfassung

Die Verbreitung von Fake News und Fehlinformationen ist zu einem bedeutenden globalen Problem geworden. Faktenüberprüfung, ist eine der Hauptmethoden, um diesen Effekten entgegenzuwirken. Durch die jüngsten Fortschritte in der Verarbeitung natürlicher Sprache (NLP) können Forscher nun die Automatisierung dieses Prozesses untersuchen. Diese Arbeit nutzt die neuesten großen Sprachmodelle (LLMs) für erklärbare, wissensbasierte Claim Verification im offenen Bereich.

Wir entwickeln drei modulare Pipelines zur Überprüfung von Behauptungen, die die In-Context-Lernfähigkeiten von LLMs, insbesondere GPT-3.5, nutzen. Diese Pipelines zerlegen Behauptungen in Teilfragen, beantworten jede Teilfrage einzeln und führen abschließend eine Schlussfolgerung über die gesammelten Beweise durch. Die Basis-Pipeline verwendet Schlussfolgerungen über natürliche Sprache, die Prädikat-Pipeline über symbolische Darstellungen von Behauptungen und die Wissensgraphen-Pipeline nutzt Beweise aus DBPedia für die Schlussfolgerung. Wir untersuchen verschiedene Beweisquellen für wissensbasierte Schlussfolgerungen.

Unsere Ergebnisse zeigen, dass diese Pipelines viele etablierte Basislinien (transformer-basiert oder LLM-basiert) für allgemeine Multi-Hop-Behauptungen sowie für realistische Behauptungen in spezifischen Bereichen wie Medizin, Klimawandel und COVID-19 übertreffen. Darüber hinaus zeigten sowohl quantitative Ergebnisse als auch eine Benutzerbefragung, dass der Einsatz von Mixtral-8x7b als Schlussfolgerungs-LLM die Leistung erheblich verbesserte, mehr als die Wahl der Beweisquelle, ob wissensbasiert oder nicht.

Contents

Acknowledgments	iv
Abstract	v
Kurzfassung	vi
1 Introduction	1
1.1 Motivation	1
1.2 Research Questions	2
1.3 Outline	2
2 Background	3
2.1 Misinformation	3
2.2 Claim Verification	4
2.3 Large Language Models	6
2.4 Knowledge Graphs	8
3 Related Work	10
3.1 QACHECK: A Demonstration System for Question-Guided Multi-Hop Fact-Checking	10
3.2 Explainable Claim Verification via Knowledge-Grounded Reasoning with Large Language Models	12
3.3 KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models	13
4 Methodology	15
4.1 Datasets	15
4.1.1 HOVER	15
4.1.2 Climate-Fever	15
4.1.3 HealthFC	17
4.1.4 CoVERT	17
4.1.5 FactKG	18
4.2 Step-By-step Claim Verification	19
4.2.1 Base Pipeline	19
4.2.2 Predicate Pipeline	23
4.2.3 Knowledge Graph Pipeline	26

4.3	Evaluation	30
4.3.1	F1-score	30
4.3.2	Survey	31
4.4	Web GUI	31
5	Results	34
5.1	HOVER	34
5.1.1	Method Comparison	35
5.1.2	Evidence Source Comparison	38
5.1.3	Reasoner Comparison	41
5.1.4	Further Analysis	43
5.2	Domain Specific Datasets	45
5.2.1	HealthFC	45
5.2.2	Climate-Fever	47
5.2.3	CoVERT	48
5.3	FactKG	49
5.3.1	Experimental Results	49
5.3.2	Output Analysis	50
5.3.3	Further Analysis	51
5.4	Question Coverage Survey	53
6	Discussion	55
6.1	Key Findings	55
6.2	Limitations and Future Work	56
7	Conclusion	58
	List of Figures	60
	List of Tables	62
	Bibliography	64

1 Introduction

The first chapter provides an introduction to the topic, outlines the motivation behind this research, and states the research questions. Additionally, it offers an overview of the thesis structure.

1.1 Motivation

In recent years, most notably since the 2016 US presidential election, the issue of fake news has become a global problem [1]. This surge in misinformation has led to the emergence of independent fact-checking initiatives, also referred to as claim verification in the literature. Organizations such as PolitiFact and Snopes perform manual fact-checking by gathering evidence to determine whether a claim (e.g., a political quote or a tweet) is supported or refuted.

Advancements in natural language processing (NLP) have enabled the automation of this task. Pre-trained transformer models have become utilised to classify claims as supported, refuted (sometimes as not enough information as well) but also to search for relevant evidence within the corpus [2]. However, this approach requires large volumes of labeled training data and evidence corpora. Another significant challenge is the explainability of such classifications. It is not only crucial to obtain a verdict from the fact-checking procedure but also to understand the reasoning behind it, along with sources that support or refute the claim.

This is where novel large language models (LLMs), such as GPT-3.5 [3], offer potential. These models can interpret natural language, generate text, answer questions, and perform various tasks when provided with examples. Trained on vast amounts of data, LLMs can also serve as sources of information [4]. It has been demonstrated that LLMs can reason over symbolic representations of information, such as predicates [5]. These characteristics can be leveraged for claim verification.

However, information provided by LLMs can sometimes be inaccurate [6]. Therefore, integrating external sources of evidence, such as web search results or knowledge graphs, can be beneficial. Knowledge graphs typically contain accurate but sparse information.

Given the recent proliferation of large language models, the area of claim verification using LLMs remains relatively unexplored. Most approaches have been evaluated on general domain datasets [4], [5], [7]. Exploring how LLMs handle claims from various domains could provide valuable insights.

1.2 Research Questions

The research questions guiding this thesis are as follows:

RQ1 How can the use of LLMs help claim verification?

RQ2 Does leveraging knowledge from knowledge graphs and structured reasoning improve performance?

RQ3 How do different domains compare in this task?

1.3 Outline

The structure of this thesis is as follows: first, we introduce the necessary background for this research, encompassing topics such as misinformation, claim verification, large language models, and knowledge graphs. Next, we review recent related work utilizing large language models, particularly approaches that divide claims into smaller subclaims for step-by-step processing. Subsequently, we introduce our target datasets and three pipelines: the base pipeline, the predicate pipeline, and the knowledge graph pipeline, utilizing approaches from related work. Additionally, we organized a survey to compare the quality of explanations generated by these pipelines. Finally, we present and discuss the results of our experiments and analyze the outputs of the pipeline for further insights.

2 Background

This chapter introduces the essential topics of the thesis. The reader needs to familiarize themselves with the following areas: misinformation and fact-checking, automated claim verification, large language models and knowledge graphs.

2.1 Misinformation

The term misinformation, in the context of politics, was introduced as early as in 2000 by Kuklinski [8]. The resistance to facts within the American public, according to Kuklinski, did not come from the public being *uninformed*, but rather *misinformed*. Uninformed are those, who do not hold any factual beliefs. Misinformed people, on the other hand, are very confident in their wrong beliefs. This results in people having different preferences when it comes to public policies, as compared to if they were correctly informed about them in the first place.

Closely related concepts to misinformation are rumours and conspiracy theories. Rumours could be defined as claims about people, events or institutions, which have not been shown true, but their credibility comes from the fact other people believe them [9]. In contrast, conspiracy theories are “an effort to explain some event or practice by reference to the machinations of powerful people, who attempt to conceal their role” [9].

The current-age umbrella term for all kinds of misinformation, ranging from deliberately misleading attempts to undermine national security to news challenging the current status-quo is *fake news* [10]. The term became especially wide spread in relation to the 2016 United States presidential election campaign and the 2016 Brexit referendum campaign, with serious concerns about if and how much the spread of fake news influenced the election and referendum results respectively [11] [12].

Social media have played a major role in spreading fake news. A research focusing on Twitter’s influence on the 2016 US presidential election [13] examined 30 millions of tweets from 2.2 millions of users, which were related to the election and contained a link to a news source. The results show that 10% of the tweets spread fake news and 15% of all tweets shared extremely biased news. While the centre and left leaning news were usually shared by a small set of highly influential actors, the sharing of fake and biased news was a collective behaviour of many smaller actors.

The proportion of misinformation on social media is not the only alarming problem. It has been shown that false information spread further, faster, deeper and more broadly than truthful information [14]. In short, false news reached more people than the true news. They also argued that false news introduced more novelty than true news, causing people to be

more likely to share such news. Main topics of the news were politics, terrorism, natural disasters or science.

One of the most recent areas of the spread of misinformation is health and the COVID-19 pandemic specifically. The spread of such false news on social media and their impact has quickly become a focus of a lot of research [15]. During the pandemic, Twitter was used as the main outlet for people to express their feelings about the disease, their opinions on its origin or suggestions for cures. This introduced false, partly false or true claims into the online space [16]. A study about COVID-19 misinformation on Twitter [17] analysed 1500 tweets with either false or partly false claims. The analysis showed that 1274 claims were false and the rest partly false.

The spread of misinformation about the pandemic had a negative impact on individuals and also a society as a whole. The false information about the spread, treatment of the disease and vaccine development led to fatalities in some cases. Furthermore, people started to lose faith and trust in science [16]. Misinformation had, as well, influence on mental health. Panic, fear, depression and fatigue were reported to be affects of exposure to pandemic-related fake news [18].

Climate change is another target of misinformation. There still exists a politically polarised discussion on it. Al-Rawi et al. [19] examined 6.8 milion tweets that mentioned *fake news* and extracted those related to climate change. Half of the most retweeted posts were either claiming global warming was a natural occurrence or straight up denied it, whereas only around third of them stated it was caused by human notion.

2.2 Claim Verification

Misinformation could be overcome through correction - that is exposing people to true information. Attempts at correction have shown different results from successful, intermediate to failed [20]. Kuklinski et al. [8] stated that people “resisted correct information”. More recently, Porter and Wood [21] conducted a survey on effectiveness of fact-checking on a global scale. Participants from Nigeria, Argentina, South Africa and the United Kingdom were exposed to either misinformation, misinformation followed by a fact-check or control. Fact-checks reduced false beliefs in all countries, with effect lasting at least 2 weeks.

Fact-checking is a process of verifying factuality of a claim. Such fact-checks are usually provided by independent fact-checking organisations. Some of the most popular fact-checking websites are: PolitiFact, Snopes or FactCheck.org. The framework at PolitiFact [22] is as follows: If a claim by some person, website or organisation is found worthy of fact-checking (statement seems misleading or wrong, noteworthy or likely to be repeated by others), the author is contacted to provide their sources. Veracity of the claim is checked against information available from primary sources, government reports, academic studies or other data. Multiple researchers are involved in verifying a claim. PolitiFact have developed their own 6 scale rating system quantifying how true or how false a statement is. In this thesis and also other scientific works on automated fact-checking we refer to fact-checking as *claim verification*.

Manual claim verification is very time consuming. Since journalists and researchers, often multiple of them need to verify just one claim, have to go through many different sources and then provide a thorough explanation of their verdict. With new claims “emerging” every day, battling misinformation becomes increasingly more difficult. This was the motivation for researchers to focus on automated claim verification.

Vlachos and Riedel [23] define claim verification as a classification task - assignment of truth value to a claim. However, often claims cannot be seen as strictly false or strictly true, as sometimes there exists contrasting evidence. They proposed to separate claim verification into separate subtasks, inspired by the manual workflow of fact-checking. Such smaller tasks then can be tackled by NLP techniques [24].

The standard claim verification pipeline consists of three steps [25]: **document retrieval**, **evidence selection** and **verdict prediction**, as depicted in Figure 2.1. Document retrieval concerns finding the right document to verify a claim. That is a wikipedia page, scientific article, etc. Once such a document is found, we need to identify and collect sentences (or paragraphs) that can be used to verify the claim. And finally, given the claim and the evidence, make the prediction about the veracity of the claim.

Some of the methods to tackle the first step of the pipeline employ vector representations of the corpus [26]. One approach uses sparse retrieval. That is corpus is represented as a sparse TF-IDF matrix and then matched with keywords in the query. The other approach utilizes dense retrieval, where dense corpus vectors and the claim vector are compared with regards to their semantic similarity.

Evidence retrieval usually employs BERT [26]. Evidence is commonly selected on a sentence level, so the candidates are sentences most similar to the claim. Another approach can be treating evidence selection as a classification problem and training a multi-layer perceptron to decide which sentences to use.

For the last step, veracity prediction, BERT models are again commonly utilised [26]. This is a task of classification, where input is a claim and its evidence sentences. These are then used to produce the output that is the veracity label.

Once veracity is obtained, this might be not enough to present the results. Explainability of the model’s decision is especially desirable in such a task, otherwise the results do not possess much value. An approach utilising a BERT-like model models both prediction and justification jointly as extractive-summarization. [27]

For the task of claim verification, numerous datasets have been created. One of the baselines datasets is FEVER [28]. It consists of 185,445 claims generated by altering sentences found in Wikipedia. These were then manually verified by annotators and labeled into three classes: SUPPORTED, REFUTED and NOTENOUGHINFO. For each claim, annotators also provided a reference to one or multiple Wikipedia pages and respective evidence sentences supporting or refuting the claim.

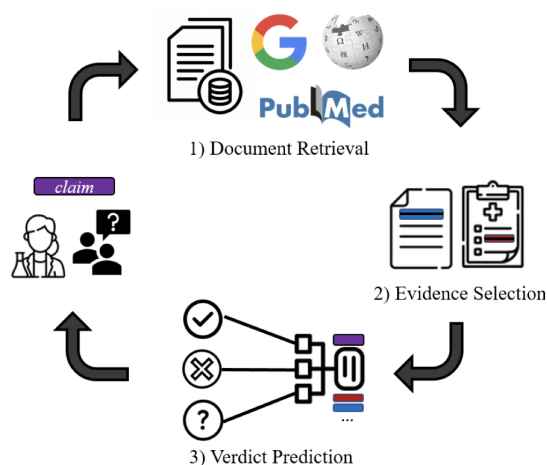


Figure 2.1: Steps in the claim verification pipeline: document retrieval, evidence selection and verdict prediction [26].

2.3 Large Language Models

Large Language Models (LLMs) are NLP (natural language processing) algorithms which belong to a family of GenerativeAI. These are neural networks based tools for generating text, images, video or other types of media [29]. LLMs are able to generate text - that is, based on a text input, repeatedly predict the next word [3]. They made their major breakthrough at the end of 2022 with the public release of ChatGPT by OpenAI [30]. It is a chatbot capable of answering human questions or perform various tasks with textual output.

The path towards LLMs was paved by language models called transformers [31]. They are a sequence-to-sequence model based on the encoder-decoder architecture. Given the input sequence, encoder produces a continuous vector z , which is then fed into the decoder and outputs a sequence of elements, one at a time [31]. Another important mechanism in transformers is attention. Attention helps dynamically assign relevance to input elements of a sequence [32]. Transformers reduced the need for task-specific models by creating pre-trained transformers, which could be then fine-tuned for specific tasks [33]. However, such fine-tuning requires a task-specific dataset. Curating a dataset for every new task is a difficult and time consuming task on its own. Additionally, it has been shown that the bigger the transformer is (has more parameters), the bigger the dataset and more computational power, the better performance in various NLP tasks [34].

Large language models built up on these developments and findings and leveraged in-context learning [35]. In-context learning encompasses training a language model to perform various tasks. Then, as input, the model receives a task to perform with possible examples (demonstrations) in natural language on how to perform this task. The important difference between fine-tuning and prompting (providing demonstrations) is that the model weights do not change.

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



(a)

One-shot

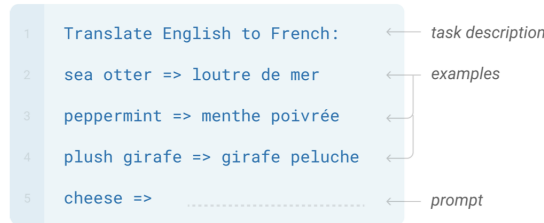
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



(b)

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



(c)

Figure 2.2: Examples of zero-shot (a), one-shot (b) and few-shot learning (c) [3].

When it comes to in-context learning, the following settings are used: few-shot learning, one-shot learning and zero-shot learning with examples in Figure 2.2. In few-shot learning, a number of demonstrations are provided. This significantly reduces the task specific data that would be otherwise needed if fine-tuning a model. One- and zero-shot learning analogously provide one and zero examples respectively.

GPT-3 [3], an LLM introduced by OpenAI in 2020 has 175 million parameters and showed a strong performance on many tasks, in some cases almost matched the performance of fine-tuned task-specific models. Even better results were reported for the latest GPT-4 model [36]. It managed to pass a bar exam and also match human performance in other academic tasks.

The model mostly adopted by the public is ChatGPT [30], fine-tuned version of GPT-3.5. It was developed to follow instructions and provide detailed explanations in an interactive chatbot based way. For the training, AI trainers provided conversations where they “played” both sides. Furthermore, they also ranked model responses by quality and this data was used in reinforcement learning.

One of the phenomenons and limitations to be aware of with LLMs is hallucination. That is generating text which is coherent and sounds confident in what it claims, but is factually incorrect or cannot be verified [6]. Two US lawyers were fined, after they used ChatGPT to write a court filing with fake citations, hallucinated by ChatGPT [37].

2.4 Knowledge Graphs

Knowledge graph (KG) could be defined as “a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities” [38]. For example, nodes could be people, locations or animals. Relations then could be relationships between people (parent, sibling, colleague, ...) or between people and locations (a person was born in, died in, ...) or hierarchical relations like animal taxonomy.

Knowledge is usually represented as a triple (*subject, predicate, object*), where subjects and objects are nodes and predicate is a relation. For example, (“*Leonardo da Vinci*”, “*born in*”, “*Italy*”) represents the information that Leonardo da Vinci was born in Italy. Such a representation is called RDF (Resource Description Framework) [39]. An example of a simple KG with living things and their relationships is depicted in Figure 2.3. Ontology of a graph is a formal definition of the kinds of objects and relations in a knowledge graph [40].

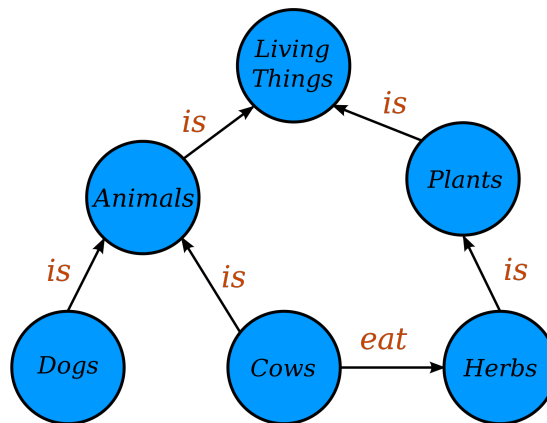


Figure 2.3: Example of a simple knowledge graph representing living creatures and their relationships [41].

The advantage over the relation databases is that there does not have to be an up-front specified schema. This allows for a more flexible approach when building a knowledge graph [38]. In comparison to NoSQL databases, graph query languages also provide the recursive functionality to find paths between entities [38].

Knowledge graphs represented as RDF can be queried by the SPARQL [42] language. The code snippet in Figure 2.4 shows an example of a SPARQL query. It should retrieve the genres of works by the author of Tokyo Mew Mew from the DBpedia knowledge graph [43].

DBpedia [44] is a community project that aims to extract information from Wikipedia into a knowledge graph available to everyone on the Web. Apart from SPARQL queries, it is possible to explore the graph through a Web browser. It automatically evolves as Wikipedia changes and is truly multilingual.

Another example of a knowledge graph is UMLS (Unified Medical Language System)

```
PREFIX dbprop: <http://dbpedia.org/ontology/>
PREFIX db: <http://dbpedia.org/resource/>
SELECT ?who, ?WORK, ?genre WHERE {
db:Tokyo_Mew_Mew dbprop:author ?who .
?WORK dbprop:author ?who .
OPTIONAL { ?WORK dbprop:genre ?genre } .
}
```

Figure 2.4: Example of a SPARQL code to retrieve genres of work of the author of Tokyo Mew Mew [43].

[45], which provides linking of medical terms, drug names and health information across different computer systems. It provides a semantic network with broad categories and their relationships. Another component of UMLS is its Metathesaurus, which is a biomedical thesaurus that is organised by concept or meaning and groups terms representing the same concept from different vocabularies.

Knowledge graphs have a wide use in the current age of AI. They can be utilised in recommendation systems by creating a KG with user data [46], question answering a natural language question by searching for evidence on a knowledge graph [47] or provide the needed context for drug discovery [48]. On the other hand, they can serve as an output of other machine learning algorithms, for instance, chatbots can be used to automatically populate a KG [49].

3 Related Work

In the previous chapter, we explained the problem of misinformation, its impact on our society and our main task - claim verification. Furthermore, two important technologies were introduced: LLMs and knowledge graphs. Unsurprisingly, these have been leveraged to solve the task of claim verification, especially in the past year.

This chapter describes some of the most recent developments in claim verification using LLMs and KGs. Claims usually require a multi-step reasoning to be verified. The following research deals with step-by-step multi-hop claim verification and was a base for this thesis.

3.1 QACHECK: A Demonstration System for Question-Guided Multi-Hop Fact-Checking

QACheck [4] is a question-based claim verification pipeline. It solves two problems of claim verification. First, the multi-hop nature of claims. Second, limited explainability of the veracity prediction in previous work.

The main idea is, to verify a claim, it can be decomposed into a series of questions, which can be answered separately. However, these questions are not generated at once, but iteratively, based on the previous question-answer pairs. For this reason, this system has the following modules: claim verifier, question generator, question answering model, validator and a reasoner. Each of the modules is powered by an LLM and prompted using few-shot learning. The iterative pipeline and the data flow are visualised in Figure 3.1.

The input claim first enters the claim verifier. Here, given the question-answer pairs collected so far, an LLM decides, if there is enough evidence to tell, whether the claim is true or not. The prompt looks like this:

```
Claim = CLAIM
We already know the following:
CONTEXT
Can we know whether the claim is
true or false now? Yes or no?
```

If there is enough evidence, we step out of the loop and continue to the reasoner module. Otherwise, which is always the case at the beginning, we proceed to question generation.

The question generator either generates the first question to ask, or the follow-up question based on already collected evidence. The prompt for the first question looks like this:

```
Claim = CLAIM
```

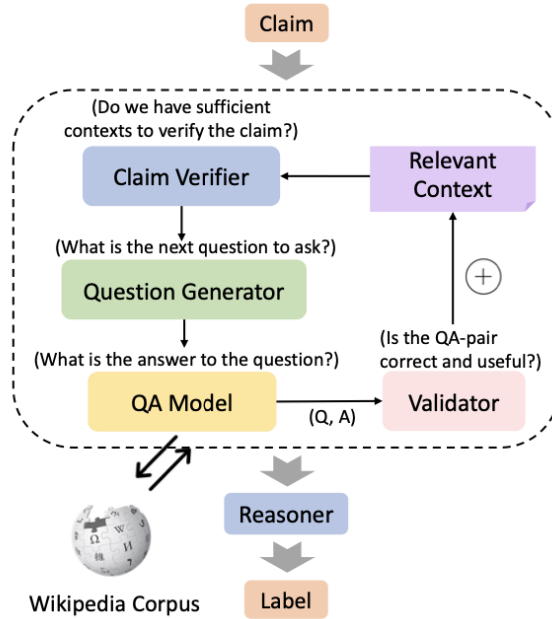


Figure 3.1: QACheck pipeline [4].

To verify the above claim, we can first ask a simple question:

Instructions for the follow-up question:

Claim = CLAIM

We already know the following:

CONTEXT

To verify the claim, what is the next question we need to know the answer to?

The generated question is then sent to the question answering model. Here, a few methods were employed. One of them was the *retriever-reader* approach, where first the relevant Wikipedia document is retrieved and then the relevant sentences or paragraphs are chosen. Then they introduced *GPT reciter-reader*, which first asks an LLM to “recite” the relevant Wikipedia article. This evidence is then used to answer the question.

The last step of the loop is the validator. It decides, whether the question and its retrieved answer make sense and should be added to the list of evidence. The prompt for this task is:

Claim = CLAIM

We already know the following:

CONTEXT

Now we further know:

NEW QA PAIR

Does the QA pair have additional
knowledge useful for verifying
the claim?

The final step of the model is the reasoner. Here, all the gathered evidence is combined and the final verdict is made. Again, using an LLM.

This approach has outperformed other baseline LLM approaches on 3-hop claims from the HOVER dataset and almost topped the best ones for 2 and 4-hop claims and the FEVEROUS dataset. The sequence of questions, their answers and evidence sentences serve as a great way to explain the reasoning behind each prediction.

3.2 Explainable Claim Verification via Knowledge-Grounded Reasoning with Large Language Models

First-Order-Logic-Guided Knowledge-Grounded (FOLK) Reasoning [5] focuses on three main aspects. Claim verification without having to rely on manually annotated data; provide human-readable explanations of the veracity prediction; and use knowledge-grounded reasoning, which means using information outside of the parameters of an LLM. Each LLM step has a set of demonstrations used for few-shot learning.

The FOLK approach utilises LLMs and their capacity to reason over symbolic knowledge. Therefore, each claim is separated into sub claims so that each sub claim can be represented in first-order logic. This is called FOL-guided claim decomposition.

Afterwards, the sub claims are verified separately. A question is generated from the subclaim and its predicate representation and then Google-searched using SerpAPI. This part ensures the LLM does not rely only on its internal knowledge but also on external sources.

The FOL-guided reasoning step looks at each sub claim and decides if it is true or false. Then, with the help of the predicate representation, prediction and also explanation are produced.

Figure 3.2 shows a demonstration of verifying a claim “*Lubabalo Kondlo won a silver medal in the 2012 SportAccord World Mind Games inaugurated in July 2011 in Beijing.*”. The claim is separated into two sub claims, each represented by a triple. For each triple, a question is generated and then answered. We can see that the second subclaim is false, since based on the provided answer, the Olympics took place in December, not July. Therefore, the whole claim is considered *not supported*.

FOLK overperformed the baseline models on the FEVEROUS and HOVER (except for 2-hop claim) datasets. This shows that FOL-like predicates improve the reasoning capability of LLMs. Experiments also proved that knowledge-grounding is more reliable than LLM’s internal knowledge. Lastly, FOLK’s explanations are informative, accurate and have excellent readability. Its limitations lie in that the tested claims are all easily dividable into sub claims, while real-world claims require implicit reasoning to verify.

Claim: Lubabalo Kondlo won a silver medal in the 2012 SportAccord World Mind Games inaugurated in July 2011 in Beijing.
Label: <i>[NOT_SUPPORTED]</i>
Predicates: Won(Lubabalo Kondlo, a silver medal) ::: Verify Lubabalo Kondlo won a silver medal Inaugurated(the 2012 SportAccord World Mind Games, July 2011, Beijing) ::: Verify the 2012 SportAccord World Mind Games was inaugurated in July 2011 in Beijing.
Follow-up Question: What did Lubabalo Kondlo win in the 2012 SportAccord World Mind Games? Grounded Answer: In 2012 he won the silver medal, ... in Beijing, China.
Follow-up Question: When and where was the 2012 SportAccord World Mind Games inaugurated? Grounded Answer: The International Mind Sports Association (IMSA) inaugurated the SportAccord World Mind Games December 2011 in Beijing ...
Prediction: Won(Lubabalo Kondlo, a silver medal) is True because In 2012 he won the silver medal at the SportAccord World Mind Games in Beijing, China. Inaugurated(the 2012 SportAccord World Mind Games, July 2011, Beijing) is False because The International Mind Sports Association (IMSA) inaugurated the SportAccord World Mind Games December 2011 in Beijing. Won(Lubabalo Kondlo, a silver medal) && Inaugurated(the 2012 SportAccord World Mind Games, July 2011, Beijing) is False . The claim is <i>[NOT_SUPPORTED]</i> .
Explanation: Lubabalo Kondlo won a silver medal in the 2012 SportAccord World Mind Games. However, the event was inaugurated in December 2012, not July 2011, in Beijing.

Figure 3.2: Claim verification using FOLK [5].

3.3 KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models

KG-GPT [7] explores multi-hop claim verification using LLMs with knowledge graphs as the source of evidence, which is a not very deeply researched area. It is comprised of three steps: sentence segmentation, graph retrieval and inference. The knowledge graph used for reasoning is DBPedia.

As with the previous papers we discussed, each stage is also directed by LLMs. In the first step of sentence segmentation, given the claim and ground truth entity set in the claim, it should be separated into simple subclaims. These subclaims should have a simple (subject, predicate, object) structure. Additionally, an entity set of the subclaims should be retrieved. As shown in the example in Figure 3.3 (1).

The second step is graph retrieval (Figure 3.3 (2)). Here, relations of the entities and relations associated with the entity’s type are considered. This process is done for all subclaims together. An overlap between relations of all entities and types is found and saved in a list. This list is then given to the LLM with the claim and asked to retrieve *TopK* relation candidates semantically related to the original claim. Finally, all KG triples containing the relation candidates are returned.

The last stage is inference (Figure 3.3 (3)), where all the related triples and the original

3.3. KG-GPT: A GENERAL FRAMEWORK FOR REASONING ON KNOWLEDGE GRAPHS USING LARGE LANGUAGE MODELS

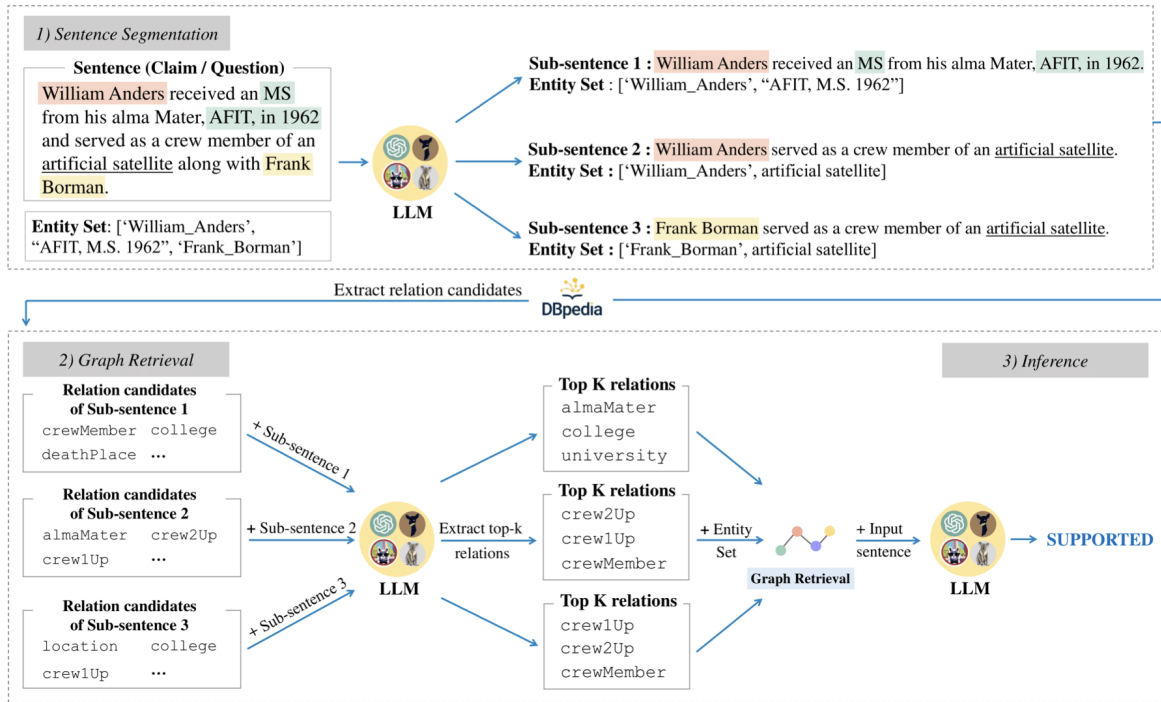


Figure 3.3: Claim verification using KG-GPT [7].

claim are fed to the LLM. Its job is to decide whether the claim is true or false and provide a short explanation of the answer.

The model managed to outperform other fully-supervised models. However, it still lagged behind fully-supervised KG-based models. The performance is highly dependent on the number of demonstrations in in-context learning.

4 Methodology

This chapter introduces the methods to answer the research questions stated in 1.2. First, the datasets used to evaluate the methods, the various claim verification pipelines powered by LLMs, web GUI for claim verification and a survey on qualitative comparison of different pipelines.

4.1 Datasets

In this section, we introduce datasets used to evaluate our models. Since we want to investigate claim verification on multiple domains, datasets were chosen accordingly. The domains are general, climate, health-related, and political claims from fact-checking websites. It is essential to understand the nature of the datasets.

4.1.1 HOVER

The dataset HOVER [50] has become a baseline for multi-hop reasoning testing. The claims require evidence from as many as four different English Wikipedia articles. Most of the 3/4-hop claims were written in multiple sentences, adding complexity. In this dataset, each claim has assigned a list of evidence sentences.

This dataset was created in 3 steps: *claim creation*, *claim mutation* and *claim labelling*. In the first stage, 2-hop claims were manually rewritten from HotpotQA question-answer pairs. These were then extended to include evidence from more Wikipedia articles. In the second stage, these claims were mutated to introduce variety into the dataset. In the last stage, human annotators labelled the claims as SUPPORTED, REFUTED or NOTENOUGHINFO. The latter two were then merged into NOTSUPPORTED.

Figure 4.1 shows the types of claims in HOVER and their examples. In general, we have 2, 3 and 4-hop claims. We need 2, 3 and 4 Wikipedia documents to verify the claim. You can see the different relationships between documents (entities in the claim). They are "linked" based on hyperlinks between the Wikipedia articles. HOVER has over 26000 claims, split into train, dev, and test datasets.

4.1.2 Climate-Fever

The first domain-specific dataset we used is Climate-Fever [51]. Its claims are related to the climate change. Climate-Fever is based on the Fever [28] approach. The claims were collected from either climate denier/sceptics sources or scientifically informed sources. Hence, they are

4.1. DATASETS


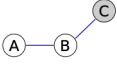
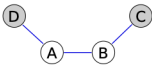
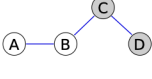
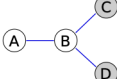
#H	Reasoning Graph	Examples
2		<p>Claim: Patrick Carpentier currently drives a Ford Fusion, introduced for model year 2006, in the NASCAR Sprint Cup Series.</p> <p>Doc A: Ford Fusion is manufactured and marketed by Ford. Introduced for the 2006 model year, ...</p> <p>Doc B: Patrick Carpentier competed in the NASCAR Sprint Cup Series, driving the Ford Fusion.</p>
3		<p>Claim: The Ford Fusion was introduced for model year 2006. <i>The Rookie of The Year in the 1997 CART season</i> drives it in the NASCAR Sprint Cup Series.</p> <p>Doc C: The 1997 CART PPG World Series season, the nineteenth in the CART era of U.S. open-wheel racing, consisted of 17 races, ... Rookie of the Year was Patrick Carpentier.</p>
4		<p>Claim: <i>The model of car Trevor Bayne drives</i> was introduced for model year 2006. The Rookie of The Year in the 1997 CART season drives it in the NASCAR Sprint Cup.</p> <p>Doc D: Trevor Bayne is an American professional stock car racing driver. He last competed in the NASCAR Cup Series, driving the No. 6 Ford Fusion...</p>
		<p>Claim: The Ford Fusion was introduced for model year 2006. It was driven in the NASCAR Sprint Cup Series by <i>The Rookie of The Year of a Cart season, in which the 1997 Marlboro 500 was the 17th and last round.</i></p> <p>Doc D: The 1997 Marlboro 500 was the 17th and last round of the 1997 CART season...</p>
		<p>Claim: The Ford Fusion was introduced for model year 2006. The Rookie of The Year in the 1997 CART season drives it in the series held by <i>the group that held an event at the Saugus Speedway.</i></p> <p>Doc D: Saugus Speedway is a 1/3 mile racetrack in Saugus, California on a 35 acre site. The track hosted one NASCAR Craftsman Truck Series event in 1995...</p>

Figure 4.1: Types of claims and their examples in the HOVER dataset [50].

considered real-world claims rather than artificial ones in HOVER [50]. Claims were labelled using Wikipedia as evidence.

The final dataset consists of 1535 claims, labelled into four groups: SUPPORTS (655), REFUTES (253), DISPUTED (153) and NOTENOUGHINFO (474). DISPUTED is for the cases in which both the supporting and refuting evidence was found. NOTENOUGHINFO denotes claims for which enough information could not be found among the chosen Wikipedia pages. Each claim comes with a set of five evidence sentences from Wikipedia.

The topic analysis of the claims found 21 different categories. From general ones to more specific ones concerning *sea-level rise*, *emmissions* or *polar areas*. Since the claims are real-world claims, their structure and phrasing might be challenging for fact checkers. Some example claims from Climate-Fever are:

- Droughts and floods have not changed since we’ve been using fossil fuels.
- Extreme weather isn’t caused by global warming.
- A paper by Ross McKittrick, an economics professor at the University of Guelph, and Patrick Michaels, an environmental studies professor at the University of Virginia, concludes that half of the global warming trend from 1980 to 2002 is caused by Urban Heat Island.

4.1.3 HealthFC

HealthFC [52] focuses on questions from the health domain, as people nowadays use the internet to answer their health-related questions and problems. The dataset consists of 750 claims in English and German labelled by medical experts into three categories: `SUPPORTED`, `REFUTED` and `NOTENOUGHINFO`. The claims were verified using evidence from systematic reviews and clinical trials.

Claims and their evidence documents were scraped from the web portal *Medizin Transparent*, which, apart from other things, also focuses on collecting medical questions and their verification. The medical experts manually search for scientific articles and studies to answer the questions. Then, they write a summary of their findings. This document is then used as an evidence document in the HealthFC dataset. The verdicts on the website are regularly updated based on the latest scientific knowledge. HealthFC consists of claims and their veracity, which were up to date in 2022.

This dataset covers multiple subtopics within the medical domain. The most prominent are dietary supplements (18%), nutrition (15%) and immune system (15%). Claims related to COVID-19 are also present in this dataset. Regarding the distribution of the labels, most of the claims (423) belong to the class `NOTENOUGHINFO`. The second most common label is `SUPPORTED` (202) and then `REFUTED` (125).

The claims are in the form of questions like this:

- Do health benefits increase with duration and intensity of exercise?
- Do milk or dairy products promote bladder cancer?
- Does electro-acupuncture on the ear help against pain?

To better fit the claim verification pipeline, we turned the questions into declarative sentences using GPT-4 [36]. Here is an example of the zero-shot prompt and GPT’s output:

PROMPT: Turn the following question into a declarative sentence: Can arthroscopy reduce pain or improve mobility?

OUTPUT: Arthroscopy can reduce pain or improve mobility.

4.1.4 CoVERT

The last domain-specific dataset is CoVERT (Covid VERified Tweet) [53]. It is a collection of 300 COVID-19-related claims from Twitter. Claims are labelled as `SUPPORTED`, `REFUTED` and `NOTENOUGHINFO`. Annotators searched for evidence for each claim using Google and results from reputable sources.

The style of the tweets is very informal, containing hashtags and user handles (which were anonymised) or URLs. This can be an additional challenge in claim verification. Examples of such Tweets are:

- #EU #covid #brexit disagreements re covid vaccines could cause more damage within the Eu than brexit did
- Yo, why would a vaccine that creates spike proteins smooth muscle contractions cause the kidney to start releasing large amounts of blood clotting compounds in susceptible people? Must be the adenovirus. <https://t.co/AvxlcaEDFv> oh <https://t.co/A1S4YrLGcP> <https://t.co/rq8kOxTWhP>
- @username @username @username The cytokines released in severe COVID can cross the BBB & cause inflammation. The “happy hypoxia” has neuro implications. The cytokine storm can also cause clots. Again, the mechanism of cell entry is identical to SARS 1. And I can’t find any info that SARS 1 did this.

4.1.5 FactKG

The last dataset FactKG [54] is a fact-checking dataset for multi-hop reasoning on knowledge graphs. The claims were generated based on data from DBpedia on a general domain. It consists of more than 108k claims, either *SUPPORTED* or *REFUTED*. The dataset consists of claims and their graphical evidence, which are the entities in the graph. There are five reasoning types of claims.

The types and their examples are shown in Figure 4.2. The first type is *one-hop*, where two entities are in the claim and a direct relation connects them. The second kind is *conjunction*, where there is one central entity in the claim and two others connected to it by a relation. The third type tests if there *exists* a relation from an entity, regardless of what the tail entity (object) is. *Multi-hop* claims consist of 3 or more entities; however, only 2 of them are explicitly mentioned. The last type of claim is *negation*, which tests the non-existence of some relation between entities. Furthermore, the claims were transferred to a colloquial style to resemble real-life claims more.

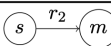
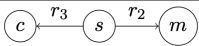
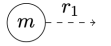


Reasoning Type	Claim Example	Graph
One-hop	AIDAstella was built by Meyer Werft.	
Conjunction	AIDA Cruise line operated the AIDAstella which was built by Meyer Werft.	
Existence	Meyer Werft had a parent company.	
Multi-hop	AIDAstella was built by a company in Papenburg.	
Negation	AIDAstella was not built by Meyer Werft in Papenburg.	

Figure 4.2: Types of claims and their examples in FactKG [54].

4.2 Step-By-step Claim Verification

This section describes our claim verification pipeline. It is powered by LLMs and consists of multiple steps (Figure 4.3): verifier, question generating module, question answering module and reasoner. A base pipeline and two other pipelines are derived from it: predicate pipeline and KG pipeline. Our verifier classifies claims into two classes: SUPPORTED and REFUTED.

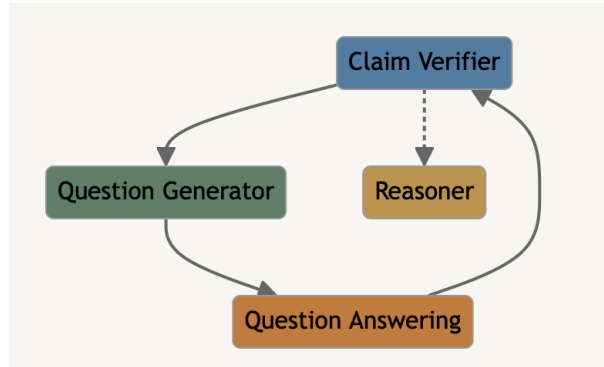


Figure 4.3: Workflow of our step-by-step claim verification.

4.2.1 Base Pipeline

Our base pipeline is based on the QACheck workflow [4]. It is based on the idea that to verify a claim, we can ask a series of simple separate questions and gather evidence to decide if the claim is supported or refuted. It is an iterative algorithm and the steps are: verifier, question generating module, question answering module and reasoner.

Input to this pipeline is a claim. The algorithm goes like this: based on the claim, we generate the first question that should be asked to verify the claim (*question generation module*). Then, we retrieve information relevant to the question and find an answer within it (*question answering module*). After finding this question-answer pair, we add it into our *evidence*. Then we ask, given the so far gathered evidence, do we have enough information to say if the claim is supported or refuted (*verifier*)? If we do, we go to the reasoning phase (*reasoner*), gather the evidence and make the final verdict about the veracity based on the evidence. If we do not have enough evidence, we continue the cycle and generate the following question to ask.

LLMs guide each of the steps. We utilise *gpt-3.5-turbo-0125* and *mixtral-8x7b-instruct*. Any LLM can be used, though. In the following subsection, each module is explained in more detail.

Question Generating Module

If we are at the beginning of the claim verification, we must ask the first question to verify a claim. If we have already gathered some question-answer pair or pairs, we must ask a follow up question. This is done by prompting an LLM using few-shot learning.

As you can see in Figure 4.4, first, the task is described. Then, multiple demonstrations of example input and desired output follow. In this case, the input claim and the expected first question to ask. There are ten demonstrations in total (7 directly from the QACHECK paper [4] and three developed by us). Ultimately, the actual claim is input and LLM is expected to output a question. Similarly, for follow-up questions (also ten demonstrations), the already gathered evidence (question-answer pairs) is provided as well (see in Figure 4.5).

```
Claim = Superdrag and Collective Soul are both rock bands.
To validate the above claim, the first simple question we need to ask is:
Question = Is Superdrag a rock band?

Claim = Jimmy Garcia lost by unanimous decision to a professional boxer that
challenged for the WBO lightweight title in 1995.
To validate the above claim, the first simple question we need to ask is:
Question = Who is the professional boxer that challenged for the WBO
lightweight title in 1995?
```

Figure 4.4: 2 of 10 demonstrations for generating the first question, based on prompts from QACheck [4].

```
Claim = Superdrag and Collective Soul are both rock bands.
To validate the above claim, we need to ask the following simple questions
sequentially:
Question 1 = Is Superdrag a rock band?
Answer 1 = Yes
Question 2 = Is Collective Soul a rock band?

Claim = Jimmy Garcia lost by unanimous decision to a professional boxer that
challenged for the WBO lightweight title in 1995.
To validate the above claim, we need to ask the following simple questions
sequentially:
Question 1 = Who is the professional boxer that challenged for the
WBO lightweight title in 1995?
Answer 1 = Orzubek Nazarov
Question 2 = Did Jimmy Garcia lose by unanimous decision to Orzubek Nazarov?
```

Figure 4.5: 2 out of 10 demonstrations for generating the follow-up question, based on prompts from QACheck [4].

Question Answering Module

The question from the previous module is forwarded to the question answering module. Here, the task is to retrieve information related to the question and extract the answer to the question. Then, this question-answer pair is added to the evidence.

This module can retrieve information from three different sources. One internal and two external sources. The external sources of evidence provide for so called knowledge-grounded-reasoning, which means the LLM based claim verification is enhanced by external information and does not rely only on its own "learned" knowledge. According to FOLK [5], this can improve the performance. This retrieved knowledge is called rationale. The sources are:

GPT Internal knowledge of GPT-3.5 is used as the knowledge source, based on the reciter-reader model from [4]. Wikipedia article with relevant information is retrieved from its "memory". Here is the zero-shot prompt:

```
[[QUESTION]]
Retrieve a Wikipedia article relevant to this question.
```

Google The question is put into the Google search using the googlesearch-python package [55]. The first result is taken, and then the snippet (relevant part of the result page shown on Google) is used as evidence. Furthermore, the link to the result is appended as a source to the rationale.

HuggingChat HuggingChat is a chatbot by HuggingFace powered by various open-source LLMs. To query the HuggingChat [56], an unofficial HuggingChat API [57] was used. The advantage of HuggingChat is its ability to search the web for answers. This was leveraged to obtain knowledge to answer our questions. The API also returns the list of websites scraped to generate the answer to a prompt. This list was appended to the rationale as the source. We chose CommandR+ [58], the latest model by Cohere, to be the LLM backbone of our chat instance. Since HuggingChat is a continuous chat (as opposed to just a single prompt OpenAI API), we should try to ensure it provides answers independent of the previous questions and answers. Therefore, the zero-shot prompt is:

```
[[QUESTION]]
Please be concise. Use only the web search results for your answer.
Answer with a whole sentence. Disregard the previous questions.
```

Once knowledge is retrieved using either source, the direct answer to the question needs to be generated. This is done using this prompt:

```
[[RATIONALE]]
Q: [[QUESTION]]
The answer is:
```

This question-answer pair is now added to the evidence. The rationale with sources is not discarded, though. It is an essential piece of information to provide to the user at the end to explain the reasoning process. Users can then also use the source websites to get more information or check the correctness of the answer.

Verifier

The following module is a verifier. Here, we check if we have gained enough evidence to say if the claim is supported or not. Given the obtained question-answer pairs, we ask LLM if we should keep generating questions or proceed to make the final veracity prediction. The prompt with demonstrations is in Figure 4.6.

As with the previous modules, here we first state the task, then ten demonstrations follow. Here, the input is a claim, the already gathered question-answer pairs and the desired output. That is if we can or cannot know if the claim is true or false. If the verifier returns true, the next stage is the reasoner. If not, we continue to question the generation stage and start a new cycle.

```
Claim = Superdrag and Collective Soul are both rock bands.
To validate the above claim, we have asked the following questions:
Question 1 =to explainAnswer 1 = Yes
Can we know whether the claim is true or false now?
Prediction = No, we cannot know.

Claim = Superdrag and Collective Soul are both rock bands.
To validate the above claim, we have asked the following questions:
Question 1 = Is Superdrag a rock band?
Answer 1 = Yes
Question 2 = Is Collective Soul a rock band?
Answer 2 = Yes
Can we know whether the claim is true or false now?
Prediction = Yes, we can know.
```

Figure 4.6: 2 out of 10 demonstrations for the verifier module, based on prompts from QACheck [4]. LLM decides if we have enough evidence to make the final veracity prediction.

Reasoner

Reasoner is the final step of the pipeline. All the evidence is assembled in this stage to make the final veracity prediction. This is done in two steps. First, the question-pairs are transformed into declarative sentences to create one paragraph of text. For each pair, this prompt was run:

```
Question = [[QUESTION]]
Answer = [[ANSWER]]
Convert the above question-answer pair into a statement.
```

Then, these sentences are concatenated and used for the veracity prediction as follows. Chain-of-thought reasoning is used here:

```
[[CLAIM]]
Is this claim true or false?
Answer: [[EVIDENCE]] Therefore, the final answer is:
```

With this approach, we can achieve a high degree of explainability, as we have the generated questions, rationale with sources, answers and a chain-of-thought-based prediction with the summarised evidence. In the following subsections, we introduce variations of this approach.

4.2.2 Predicate Pipeline

The second pipeline is based on the base one in subsection 4.2.1 and enhanced by predicates and reasoning from FOLK [5]. Here, we tried to leverage structured reasoning with predicates, enhancing the reasoning capabilities during fact checking. It consists of the same steps (modules) but with modifications described in the following parts.

Question Generating

The questions are generated the same way as in the base pipeline, but the task is to also create a predicate for each question. Each first and follow-up question should be represented by a predicate and a simple sub-task, which adds an additional guide for what exactly should be verified with that question. This should be helpful later in the reasoning step when the LLM "evaluates" all the gathered evidence. The claims and questions themselves in the prompt are the same as in the base pipeline.

The prompt and example demonstrations show how the predicates look (Figure 4.7). They have a simple *Predicate(subject, object)* structure. Then, a short verification command is attached to specify what should be verified to verify the whole claim.

Generating the follow-up questions is analogous (Figure 4.8, but here the evidence is represented by triples: (question, answer, predicate). The whole triple is provided for follow-up question generation.

Question Answering

This module is the same as in the base pipeline, as the generated questions follow the same structure. The predicates are not for question answering. Any source of evidence can be used.

Claim: Superdrag and Collective Soul are both rock bands.

To validate the above claim, we need to ask the first question with predicate:
Question:
Is Superdrag a rock band?
Predicate:
Genre(Superdrag, rock) ::: Verify Superdrag is a rock band

Claim : Jimmy Garcia lost by unanimous decision to a professional boxer that challenged for the WBO lightweight title in 1995.

To validate the above claim, we need to ask the first question with predicate:
Question:
Who is the professional boxer that challenged for the WBO lightweight title in 1995?
Predicate:
Challenged(player, WBO lightweight title in 1995) ::: Verify name of the professional boxer that challenged for the WBO lightweight title in 1995.

Figure 4.7: 2 out of 10 demonstrations for question generation in the predicate pipeline. Each generated question is accompanied by a predicate defining the question and a simple instruction on what to verify. Based on prompts from QACheck [4] and FOLK [5].

```
Claim: Superdrag and Collective Soul are both rock bands.

Question 1:
Is Superdrag a rock band?
Predicate 1:
Genre(Superdrag, rock) ::: Verify Superdrag is a rock band
Answer 1:
Yes

To validate the above claim, we need to ask the follow-up question with predicate:
Follow-up Question:
Is Collective Soul a rock band?
Predicate:
Genre(Collective Soul, rock) ::: Verify Collective Soul is a rock band
```

Figure 4.8: 1 out of 10 demonstrations of follow-up question generation for the predicate pipeline. As already gathered, evidence and predicate from the previous question are present. Based on prompts from QACheck [4] and FOLK [5].

Verifier

Except for the evidence, the verifier step is the same as in the base pipeline. As in the previous step, the evidence here is represented by triples (question, answer, predicate). They are used to decide if the claim can already be verified or not. Part of the prompt is depicted in Figure 4.9.

```
Claim: Superdrag and Collective Soul are both rock bands.

Question 1: Is Superdrag a rock band?
Predicate 1: Genre(Superdrag, rock) ::: Verify Superdrag is a rock band
Answer 1: Yes

Can we know whether the claim is true or false now? Yes or no?

No, we can't tell.
```

Figure 4.9: 1 out of 10 demonstrations for the verification module with predicates. Triples (question, answer, predicate) are used as evidence. Based on prompts from QACheck [4] and FOLK [5].

Reasoner

The reasoning part with predicates differs significantly. It is the exact mirror of the one from FOLK [5] with the same prompt and four examples (in Figure 4.10). Here, we want to utilize LLM’s reasoning capabilities over symbolic representations, as shown in FOLK [5]. In the base pipeline, the explanation is put together from the question-answer pairs, and LLM then only provides the verdict (supported/refuted). In this approach, LLM generates the explanation in natural language itself.

It could be separated into a few sections: task, question, context, question-answer pairs, prediction and explanation. First, instructions on what to do are provided, followed by four demonstrations. Each demonstration states the claim as a question, lists the predicates with subtasks, and then lists the collected question-answer pairs. Then, the LLM should go through each predicate triple and its sub-task, decide if it was verified, and make the veracity prediction. In the end, an explanation for the verdict is returned.

4.2.3 Knowledge Graph Pipeline

The third pipeline is designed to work with knowledge graphs, DBpedia specifically. It was built on the base pipeline, where the significant difference is how evidence is retrieved and collected. In most other aspects, it follows the base pipeline.

Question Generation

Question generation is the same as in the base pipeline. However, the demonstrations in the prompt are different. Claims used for demonstrations were taken from the sentence segmentation prompt from KGGPT [7]. The reason for that was that the claims they used were from the FactKG dataset [54], hence structurally more suitable for this task.

Examples of these demonstrations in Figure 4.11 are our work. Apart from different example claims, the nature of the generated questions also differs from those in the base pipeline in Figure 4.4. Here, rather than the questions being open-ended (wh questions), they can be answered as yes or no (*Is Ahmad Kadhim Assad’s club Al-Zawra’a SC* rather than *Which team does Ahmad Kadhim play for?*). The motivation behind this is that in the later step of question answering, the more entities we have present in the question, the more evidence and connections we can retrieve from the graph.

Question Answering

The module of question answering is the core of the KG pipeline. Instead of searching for evidence and answering the question by asking an LLM or Google, the evidence is extracted from DBpedia. This module consists of a few subtasks: *entity linking*, *triple extraction*, *target relation selection* and *question answering*, as shown in Figure 4.12.

To query a knowledge graph, first, we need to know the entity or entities to work with. We used DBpedia Spotlight [59] to extract entity URIs from the generated questions. It returns

Question:
Is it true that The writer of the song Girl Talk and Park So-yeon have both been members of a girl group.?

Context:
Write(the writer, the song Girl Talk) ::: Verify that the writer of the song Girl Talk
Member(Park So-yeon, a girl group) ::: Verify that Park So-yeon is a member of a girl group
Member(the writer, a girl group) ::: Verify that the writer of the song Girl Talk is a member of a girl group

Who is the writer of the song Girl Talk? Tionne Watkins is the writer of the song Girl Talk.
Is Park So-yeon a member of a girl group? Park Soyeon is a South Korean singer. She is a former member of the kids girl group I& Girls.
Is the writer of the song Girl Talk a member of a girl group? Watkins rose to fame in the early 1990s as a member of the girl-group TLC

Prediction:
Write(Tionne Watkins, the song Girl Talk) is True because Tionne Watkins is the writer of the song Girl Talk.
Member(Park So-yeon, a girl group) is True because Park Soyeon is a South Korean singer. She is a former member of the kids girl group I& Girls.
Member(Tionne Watkins, a girl group) is True because Watkins rose to fame in the early 1990s as a member of the girl-group TLC
Write(Tionne Watkins, the song Girl Talk) && Member(Park So-yeon, a girl group) && Member(Tionne Watkins, a girl group) is True.
The claim is [SUPPORTED].

Explanation:
Tionne Watkins, a member of the girl group TLC in the 1990s, is the writer of the song "Girl Talk."
Park Soyeon, a South Korean singer, was formerly part of the girl group I& Girls. Therefore, both Watkins and Park Soyeon have been members of girl groups in their respective careers.

Figure 4.10: A demonstration for the reasoning module with predicates. Directly taken from FOLK [5].

```

Claim = Ahmad Kadhim Assad's club is Al-Zawra'a SC.
To validate the above claim, we need to ask the following first simple
subject-predicate-object question:
Question = Is Ahmad Kadhim Assad's club Al-Zawra'a SC?

Claim = Yeah! I know that Bananaman, which starred Tim Brooke-Taylor,
first aired on 3rd October 1983!
To validate the above claim, we need to ask the following first simple
subject-predicate-object question:
Question = Did Bananaman star Tim Brooke-Taylor?

```

Figure 4.11: 2 out of 14 demonstrations for question generation in the KG pipeline based on the claim used in sentence segmentation prompt in KG-GPT [7] and QACheck [4].

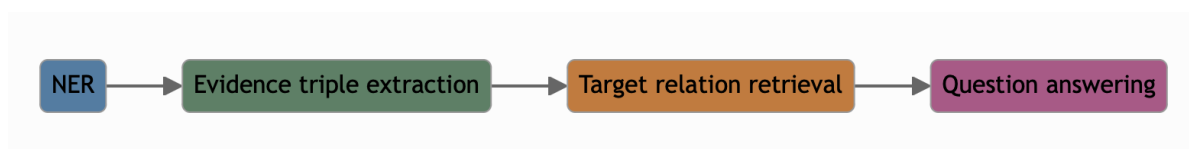


Figure 4.12: Intermediate steps in the question answering module in the KG pipeline.

DBpedia resources (e.g. https://dbpedia.org/page/Harry_Potter) for entities mentioned in the question.

We noticed that sometimes Spotlight does not recognise some of the entities. As a fallback and to broaden the recognised entity set, we implemented a simple LLM prompt to return the relevant URIs from the questions:

```

From a given sentence, find named entities within the sentence
and return only its DBpedia URI.
Write only the URI.
Sentence: [[QUESTION]]

```

Results from both approaches are then combined into one entity set.

Having these entity URIs for DBpedia, we can query the graph. For querying DBpedia and manipulating with the extracted subgraph, we used RDFLib [60]. This library allows for parsing and querying data in RDF format without using SPARQL. For each URI (entity), we extract all DBpedia triples in which the entity is a subject. The result is a subgraph.

Sometimes, one entity has multiple URIs that point to it. These "aliases" are often returned by the Spotlight API, not the "parent" URI. When such an "alias" is queried for triples, almost no triples are usually returned. However, each such "alias" has a link to its "parent" URI, stored as an object in the predicate named <http://dbpedia.org/ontology/wikiPageRedirects>.

Therefore, for each of the URIs, we check if it has a "parent" URI and extract triples from there, too.

Then, we proceed to select the final subgraph as shown in 1. We want to reduce its size to make the reasoning over it more straightforward. If there are two or more entities, we try to extract their shared relations ($(entity1, relation, entity2)$). If there are no shared relations, we select the whole graph. If we have only one entity, we select the entire graph again. FactKG [54] provides a list of around 500 relations present in the claims. These are used to further reduce the size of the subgraph by selecting only the triples that could possibly serve as evidence.

Algorithm 1 Selection of the final subgraph in the KG pipeline

Input: Entity set E , Graph triples G

Output: Selected triples S

```
 $S = []$ 
if  $len(E) \geq 2$  then
  for  $e$  in  $E$  do
    for  $f$  in  $E$  do
      if  $e \neq f$  then
         $S.append(G(subject = e, object = f))$ 
      end if
    end for
  end for
if  $len(S) == 0$  then
   $S \leftarrow G$ 
end if
else if  $len(E) == 1$  then
   $S \leftarrow G$ 
end if
```

Once we have the final subgraph, we extract all the relations from the triples. In this step, we need to choose the relation that is semantically closest to the subquestion we are dealing with. This selection is, again, done by LLM, providing 10 demonstrations (examples in ??). An empty list is returned if no relation is relevant to the question. This approach was taken and adjusted from KG-GPT [7].

Once we have the relevant relation, we select all the triples from our subgraph that have this relation. Now, we need to convert these triples into natural language using an LLM so they can serve as evidence. A simple 3-shot prompt with the following demonstrations was used:

```
Triple set: [(Leonardo DaVinci, deathPlace, France)]
Sentence: Leonardo Davinci's place of death is France.
```

```
Triple set: [(Leonardo DiCaprio, starring, Catch Me If You Can),
```

```

Question = Is there a film produced by Anatole de Grunwald?
Words set = ['composer', 'starring', 'runtime', 'director',
'writer', 'producer', 'cinematography']
Semantically most similar word from the word set to the question is:
Answer = producer

Question = Did Milan Hodža have a religion?
Words set = ['deathYear', 'leaderName', 'awards', 'award']
Answer = []

```

Figure 4.13: 2 out of 10 demonstrations for relation selection based on prompts from KG-GPT [7].

(Leonardo DiCaprio, starring, Titanic)]

Sentence: Leonardo DiCaprio starred in Catch Me If You Can and Titanic.

The sentences are used as retrieved evidence and follow the same paradigm as in the base pipeline: it is used to answer the generated question.

Verifier

The verifier module from the base pipeline is utilised.

Reasoner

The reasoner module from the base pipeline is utilised.

4.3 Evaluation

4.3.1 F1-score

Veracity prediction is a classification task. One of the evaluation metrics which is often used for claim verification is F1-score [4][5][7]. It is a harmonic mean of precision and recall. They are calculated using occurrences of true positives (TP), false positives (FP) and false negatives (FN) as below:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

For multi-class classification, F1 macro-average score is used. That is computed as the arithmetic mean of F1 scores for each class.

4.3.2 Survey

Since a significant part of our solution focuses on explainability, we would like to evaluate and compare the models not only based on quantitative measures (veracity prediction performance), but also on the quality of the explanations.

We built a survey to measure the quality of the generated questions and their answers based on two measures:

Coverage The questions and their answers cover all salient information and important points to verify the claim. No more questions need to be asked to verify the claim.

Overall Rank the question sets by their overall quality - how helpful would they be for you to easily tell if the claim is true or not.

The survey follows the system from [27], where respondents are presented with a claim and question sets (questions, answers and evidence) generated by three different models. Based on the two metrics, they should rank each set from best (1) to worst (3). If there is a draw, the same rank is allowed for multiple question sets.

There are six randomly sampled questions, one from each dataset: HOVER 2-hop, HOVER 3-hop, HOVER 4-hop, HealthFC, Climate-Fever and Covert. The models to be compared are: base pipeline with GPT-3.5-turbo in each step, predicate pipeline with GPT-3.5-turbo in each step, and base pipeline with mixtral as the reasoning LLM and GPT-3.5-turbo evidence source.

Respondents do not see the veracity label or explanation in natural language. This survey aims to measure how well each model can generate questions so that all relevant information to verify a claim is collected. Furthermore, how well these subquestions and answers help user when verifying a fact checking result.

4.4 Web GUI

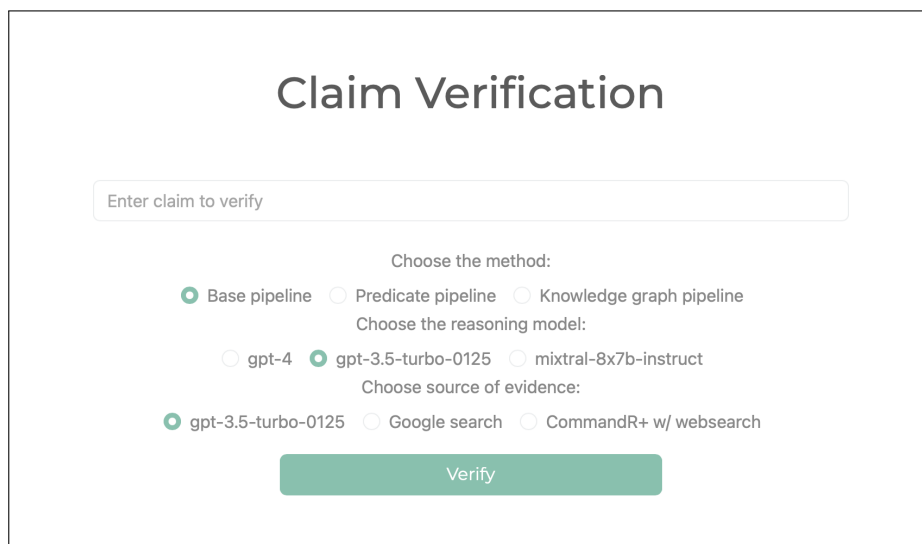
Since one of the motivations of this thesis is to provide explainable claim verification, we should also present the results and explanations in a user-friendly way. For this reason, we developed a web GUI for claim verification.

It was built using Dash Plotly [61], an open-source framework for creating web data apps in Python. The components for inputting and presenting data are from Dash Bootstrap Components [62].

The GUI is a single-page web app where users can input the claim they want to verify and then modify the method pipeline (depicted in Figure 4.14). Since our claim verification approach is very modular, various aspects can be customised. First, the kind of pipeline can be chosen (*base*, *predicate* or *knowledge graph pipeline*). The next feature is the reasoning

backbone of the pipeline, that is, LLM responsible for generating questions, making veracity predictions, etc. - the available methods are *GPT-4*, *GPT-3.5-TURBO-0125* and *Mixtral-8x7b-instruct*. The last customisable feature is the source of evidence in the question answering modules, and those are: *GPT-3.5-TURBO-0125*, *Google search* and *CommandR+ with web search* via HuggingChat.

After the claim and the pipeline settings are submitted, the specified pipeline is formed, and the claim is verified. Results are presented to the user as shown in Figure 4.15. The first box contains the claim, veracity label (true/false) and then the explanation for the decision. This box has either a green or red background based on the veracity label (green for true, red for false). Under this box, the intermediate questions (generated during claim verification) and their answers are listed. Each pair is also accompanied by the retrieved evidence (base for the answer of the question) and its sources. In the case of the predicate pipeline, the explanation does not contain the part with predicates for more clarity.



The screenshot shows a web interface titled "Claim Verification". It features a text input field labeled "Enter claim to verify". Below the input field, there are three sections of radio button options: "Choose the method:" with options "Base pipeline" (selected), "Predicate pipeline", and "Knowledge graph pipeline"; "Choose the reasoning model:" with options "gpt-4", "gpt-3.5-turbo-0125" (selected), and "mixtral-8x7b-instruct"; and "Choose source of evidence:" with options "gpt-3.5-turbo-0125" (selected), "Google search", and "CommandR+ w/ websearch". A green "Verify" button is positioned at the bottom center of the form.

Figure 4.14: Claim verification screen with the input field for the claim and the option to build a custom pipeline with chosen LLMs and methods.

Before I Go to Sleep stars an Australian actress, producer and occasional singer.
True

Nicole Kidman, Mark Strong, Colin Firth, and Anne-Marie Duff star in Before I Go to Sleep. Nicole Kidman is an Australian actress, producer, and occasional singer. Therefore, the final answer is: True.

Who stars in Before I Go to Sleep?
Nicole Kidman, Mark Strong, Colin Firth, and Anne-Marie Duff.

Before I Go to Sleep is a 2014 British-American psychological thriller film directed by Rowan Joffé and starring Nicole Kidman, Mark Strong, Colin Firth, and Anne-Marie Duff. The film is based on the 2011 novel of the same name by S.J. Watson. For more information, you can visit the Wikipedia page for the film: [https://en.wikipedia.org/wiki/Before_I_Go_to_Sleep_\(film\)](https://en.wikipedia.org/wiki/Before_I_Go_to_Sleep_(film))

Is the Australian actress, producer, and occasional singer Nicole Kidman?
Yes, Nicole Kidman is an Australian actress, producer, and occasional singer.

Yes, Nicole Kidman is an Australian actress, producer, and occasional singer. Here is a link to her Wikipedia article: https://en.wikipedia.org/wiki/Nicole_Kidman

Figure 4.15: Presentation of the claim verification results. At the top is the verified claim, the verdict and the explanation in natural language. The second part of the screen shows the questions asked to get to the verdict, answers to these questions, and evidence with sources for the answers.

5 Results

In this chapter we present our experiments, their results, outputs and comparison between them and other methods from literature. It is separated into sections by datasets: HOVER, domain specific datasets and then FactKG. The last section is dedicated to the survey for evaluation of question coverage, which encompasses examples from all datasets.

When comparing different methods, both quantitative and qualitative evaluation is used. Quantitative results are reported in tables, quantitative as comparison between the models outputs. Results from this thesis are separated by a double line from the ones from literature for more clarity. Since our approach is very modular and a part of the experiments is to test different combinations of the modules, the result tables follow the same structure: method, reasoning, evidence and then F1 scores.

Method is the name of the approach. Reasoning refers to the LLM used to make decisions and reasoning over the claim. There are approaches, for example BERT-based methods, where no LLM or advanced reasoning process is involved, so this parameter is blank. Evidence denotes the source of evidence when verifying a claim, this can be an LLM or a corpus of text like Wikipedia.

In case of our pipelines from this thesis, reasoning refers to the LLM backbone responsible for the modules of question generation, verifier and reasoner. Evidence is the evidence source used for retrieving evidence in the question answering module.

Regarding the experiments set up, openai python package was used to query both the GPT models and also the Mixtral-8x7b model [63]. Mixtral is hosted at Llama API. Model hyperparameters were kept the same as in QACheck [4], that is: *temperature* = 0.01, *top_p* = 1.0, *frequency_penalty* = 0.0, *presence_penalty* = 0.0.

The exact GPT model we used was *gpt-3.5-turbo-0125* and Mixtral *mixtral-8x7b-instruct*. In the rest of the work we refer to them as GPT-3.5 and Mixtral-8x7b for clarity.

5.1 HOVER

HOVER is an important benchmarking dataset for claim verification, so most of our experiments and analysis were performed on it. Since the test split of the data is not publicly available, ours and other papers have evaluated their models on the validation split.

In this section, we compare different methods, reasoning models and evidence sources in terms of F1-score and other statistics. Furthermore, we analyze actual outputs and explanations of the verification process.

5.1.1 Method Comparison

In this subsection, we analyze example outputs from the base and predicate pipelines and compare these methods to methods from the literature.

Base Pipeline Output Analysis

First, let's look at some of the outputs of the base pipeline using GPT-3.5 for reasoning and evidence in Figure 5.1. The outputs are screenshots from the WebGUI, so they are presented the same way a user would see them. Each of them represents a type of behaviour of the base pipeline.

The first example in 5.1a shows a correct output (based on the retrieved evidence) without any logical flaws, inconsistencies or missing evidence. The claim to verify is *"The Kentucky Department of Correction is headquartered along the Kentucky River."*. The model correctly asks about the location of the department headquarters and then about the river that flows through that location. The answers suggest that the Kentucky River does, indeed, flow along the headquarters. Given these question-answer pairs, the model classifies the claim as true. No unnecessary or irrelevant questions were asked.

The second example (5.1c) depicts a moment of flawed reasoning. The claim has a negation: *"The establishment, where Robert Cunningham Humphreys was a student (1926-7), and the University of Colorado are not private universities."*. The model asks the first reasonable question: Is the University of Colorado a private university? The answer is that it is not. This piece of evidence supports the claim. However, the model seems to misunderstand the original claim as positive, thinks the collected evidence had refuted the claim and regards it as false.

The last output in Figure 5.1 shows a situation when the model does not collect all the evidence. The claim *"The award-winning CD, that Jacqueline Horner-Kwiatek was a featured soloist on, includes the theme for the game "Civilization IV" written by the man who released The Drop That Contained the Sea which is a classical crossover album. This album was released in 2014."* is a highly complex claim, requiring multiple pieces of evidence to be verified. However, the model stops asking after one question.

Furthermore, with this limited evidence, it predicts it as true. In fact, the gold label for this claim is SUPPORTED. This is not uncommon when the model makes a correct prediction despite incomplete evidence. It shows that even when the prediction is accurate, the reasoning behind it might be flawed.

Base Pipeline Experimental Results

In Table 5.1, we can see the performance of the base pipeline together with other approaches from the literature (figures from them come from the QACheck [4] paper). There are two approaches, where an LLM model was just directly asked to provide a veracity label for the claim (InstructGPT and Codex). ProgramFC is also a step-by-step approach, where a "program" to verify a claim by verifying its subclaims is generated and then executed. Then, we naturally report the original QACheck model.

5.1. HOVER

The Kentucky Department of Corrections is headquarted along the Kentucky River.
True

The headquarters of the Kentucky Department of Corrections is located in Frankfort, Kentucky. The Kentucky River runs along Frankfort, Kentucky. Therefore, the final answer is: True.

Where is the headquarters of the Kentucky Department of Corrections located?
Frankfort, Kentucky

The headquarters of the Kentucky Department of Corrections is located in Frankfort, Kentucky. For more information, you can refer to the Wikipedia article on the Kentucky Department of Corrections: https://en.wikipedia.org/wiki/Kentucky_Department_of_Corrections

Which river runs along Frankfort, Kentucky?
The Kentucky River.

The Kentucky River runs along Frankfort, Kentucky. Here is a relevant Wikipedia article: https://en.wikipedia.org/wiki/Kentucky_River

(a)

The establishment, where Robert Cunningham Humphreys was a student (1926-7), and the University of Colorado are not both private universities.
False

The University of Colorado is not a private university; it is a public university. Therefore, the final answer is: False.

Is the University of Colorado a private university?
No, the University of Colorado is a public university.

No, the University of Colorado is a public university. It is part of the University of Colorado system, which also includes campuses in Denver and Colorado Springs. Here is a relevant Wikipedia article: https://en.wikipedia.org/wiki/University_of_Colorado_Boulder

(b)

The award-winning CD, that Jacqueline Horner-Kwiatek was a featured soloist on, includes the theme for the game \"Civilization IV,\" written by the man who released The Drop That Contained the Sea which is a classical crossover album. This album was released in 2014.
True

Composer Christopher Tin released The Drop That Contained the Sea, a classical crossover album, in 2014. Therefore, the final answer is: True.

Who released The Drop That Contained the Sea, a classical crossover album released in 2014?
Composer Christopher Tin

The Drop That Contained the Sea is an album by composer Christopher Tin, released in 2014. It is a classical crossover album that features performances by the Royal Philharmonic Orchestra, Le Mystère des Voix Bulgares, Soweto Gospel Choir, and others. The album is a follow-up to Tin's Grammy-winning album Calling All Dawns. You can find more information about the album on its Wikipedia page: https://en.wikipedia.org/wiki/The_Drop_That_Contained_the_Sea

(c)

Figure 5.1: Example outputs of the base pipeline with GPT-3.5 reasoning and evidence.

Our base pipeline outperformed the other direct or step-by-step models for the 2- and 3-hop claims (F1-score 60.02 and 55.66, respectively). For the most complicated 4-hop claims, ProgramFC performed the best, scoring 52.88.

While the performance of our base pipeline drops significantly with the number of hops, this is not the case for ProgramFC or QACheck. We can conclude our approach is more sensitive to the increasing complexity of the claims.

It is important to note that there are only a few differences between our base pipeline and its "parent" QACheck. Those are, namely, different reasoning and evidence LLM and missing the validator module in the base pipeline, and they differ in the in-context demonstrations for each module. All these aspects influenced the performance of the base pipeline.

Table 5.1: F1 scores of the base pipeline and other models (taken from [4]) on the HOVER dataset.

Method	Reasoning	Evidence	HOVER		
			2-HOP	3-HOP	4-HOP
InstructGPT - direct	-	InstructGPT	56.51	51.75	49.68
Codex - direct	-	Codex	55.57	53.42	45.59
ProgramFC	Codex	FLAN-T5	54.27	54.18	52.88
QACheck	InstructGPT	InstructGPT	55.67	54.67	52.35
Base pipeline	GPT-3.5	GPT-3.5	60.02	55.66	49.52

Predicate Pipeline Output Analysis

As an example output from the predicate pipeline (with GPT-3.5 for reasoning and evidence), we chose the same claim as in Figure 5.1 for easier comparison between the methods. The verification process is shown in Figure 5.2. The GUI only shows the generated questions and explanation (5.2a; it does not include the predicate reasoning, so the users are not confused with the symbolic representations. The generated predicates and the full prediction are in 5.2b.

Here, the generated questions and predicates were nice and straightforward but too few. Again, the model stopped asking questions before collecting all the needed evidence. When we look at the first part of the prediction, the FOLK reasoning, both predicates were evaluated as true. Therefore, their conjunction was true, and the whole claim was labelled as SUPPORTED. The reasoning here is correct but based on insufficient evidence.

The generated explanation then mentions things which were not in the evidence at all, specifically "*Jacqueline Horner-Kwiatek was a featured soloist on the award-winning CD that includes the theme for "Civilization IV."*". This was either retrieved from its internal memory or "copied" from the original claim to provide *some* explanation of the prediction.

When we compare this output to the one from the base pipeline, both prematurely stopped asking questions. However, the predicate pipeline asked two questions, not just one. Also,

its questions were simpler and easier to answer than the questions from the base pipeline. This suggests the predicates might help with decomposing the claim into simple questions. Simple questions are also more accessible for people to comprehend when seeing the outputs, so it might increase explainability.

Predicate Pipeline Experimental Results

The motivation behind implementing the predicate pipeline was the outstanding performance of structured reasoning in FOLK [5] on the HOVER dataset. However, in our case, enhancing the pipeline by predicates and using the FOLK reasoning did not improve the performance almost at all (see Table 5.2).

In fact, the predicate pipeline yields worse results than the base pipeline for 2- and 3-HOP claims. On the other hand, it is better for 4-HOP claims, where it improved from F1-score 49.52 to 51.9. Looking at Table 5.2 and Table 5.2, we can see that the most significant positive difference between FOLK and other models is on 4-HOP claims, demonstrating solid capabilities at the most involved claims. These two observations suggest the positive impact of predicates is most pronounced on complicated claims.

Table 5.2 suggests the factor playing a role in FOLK’s performance could be the source of evidence: Google snippets. In the following subsection, we look at the impact of evidence sources on the performance.

Table 5.2: F1 scores of the base pipeline, predicate pipeline and other models (FOLK from [5] and QACheck from [4]) on the HOVER dataset.

Method	Reasoning	Evidence	HOVER		
			2-HOP	3-HOP	4-HOP
FOLK	GPT-3.5	Google snippets	66.26	54.80	60.35
QACheck	InstructGPT	InstructGPT	55.67	54.67	52.35
Base pipeline	GPT-3.5	GPT-3.5	60.02	55.66	49.52
Predicate pipeline	GPT-3.5	GPT-3.5	58.55	54.39	51.9

5.1.2 Evidence Source Comparison

The reason for trying different knowledge sources was to see the effect of knowledge-grounded reasoning - verifying based on not only the parametrical knowledge of an LLM, which is often inaccurate (due to hallucinations) but also external sources. The other is to explore ways in which LLMs can help claim verification.

First, let’s look at the outputs from different evidence sources. We randomly chose the question *What is the name of the British sitcom co-written by Sian Gibson?*. The output for each evidence source for this question is in Figure 5.3.

5.1. HOVER

The award-winning CD, that Jacqueline Horner-Kwiatek was a featured soloist on, includes the theme for the game "Civilization IV" written by the man who released The Drop That Contained the Sea which is a classical crossover album. This album was released in 2014.

True

Explanation: Christopher Tin wrote the theme for the game "Civilization IV" and released the classical crossover album "The Drop That Contained the Sea" in 2014. Jacqueline Horner-Kwiatek was a featured soloist on the award-winning CD that includes the theme for "Civilization IV."

Who wrote the theme for the game "Civilization IV"?

Christopher Tin

The theme for the game "Civilization IV" was composed by Christopher Tin. You can find more information about him and his work on the Wikipedia page: https://en.wikipedia.org/wiki/Christopher_Tin

Who released The Drop That Contained the Sea?

Christopher Tin

The Drop That Contained the Sea was released by composer Christopher Tin. Here is the Wikipedia article for more information: https://en.wikipedia.org/wiki/The_Drop_That_Contained_the_Sea

(a) Predicate pipeline output

PREDICATES:

Predicate 1 = WroteTheme("Civilization IV", writer) ::: Verify the writer of the theme for the game "Civilization IV"

Predicate 2 = Released("The Drop That Contained the Sea", artist) ::: Verify the artist who released The Drop That Contained the Sea.

PREDICTION:

"WroteTheme("Civilization IV", writer) is True because Christopher Tin wrote the theme for the game "Civilization IV".

Released("The Drop That Contained the Sea", artist) is True because Christopher Tin released The Drop That Contained the Sea.

WroteTheme("Civilization IV", writer) && Released("The Drop That Contained the Sea", artist) is True.

The claim is [SUPPORTED].

Explanation:

Christopher Tin wrote the theme for the game "Civilization IV" and released the classical crossover album "The Drop That Contained the Sea" in 2014. Jacqueline Horner-Kwiatek was a featured soloist on the award-winning CD that includes the theme for "Civilization IV."

(b) Generated predicates and full prediction during the verification process in a)

Figure 5.2: Example outputs and predicates with prediction for the predicate pipeline.

5.1. HOVER

In the case of GPT-3.5 (5.3a), the source link is provided as a part of the LLM’s response. For the Google evidence (5.3b), we appended the first result’s link from the Google search. With CommandR+ via HuggingChat(5.3c), the links were provided by the HuggingChat API and appended to the response.

We can see that the GPT-3.5 evidence is the most coherent, and the response length is just a simple paragraph with a bit of context. On the other hand, CommandR+ answers only precisely what it was asked about, without any more words. In both instances, the retrieved evidence is relevant though. However, the Google snippet result is of the lowest quality. We can see the answer to our question is not present there. It seems it would follow right after the three dots. This was a common occurrence with Google snippets. Getting irrelevant evidence for otherwise relevant and well-posed questions might negatively impact the whole pipeline’s performance.

```
The British sitcom co-written by Sian Gibson is called "Peter Kay\'s Car Share."
You can find more information about the show on its Wikipedia page:
https://en.wikipedia.org/wiki/Peter_Kay%27s_Car_Share'
```

(a) Evidence retrieved from GPT-3.5

```
Siân Gibson (née Foulkes) is a Welsh comedy actress and television writer.
She has collaborated with Peter Kay, including starring in and co-writing the...
Retrieved from: https://en.wikipedia.org/wiki/Sian_Gibson
```

(b) Evidence retrieved from the first Google snippet after performing Google Search

```
The Power of Parker.
Retrieved from: https://en.wikipedia.org/wiki/Sian_Gibson
https://en.wikipedia.org/wiki/The_Power_of_Parker
https://tv.apple.com/us/person/sian-gibson/umc.cpc.3dduzgx0gkjrnrcrvxpq1naetg
https://www.bbc.co.uk/mediacentre/mediapacks/the-power-of-parker
```

(c) Evidence retrieved from CommandR+ with websearch via HuggingChat

Figure 5.3: Retrieved evidence from different evidence sources for the question *What is the name of the British sitcom co-written by Sian Gibson?*

Based on the expectations of using knowledge-grounded reasoning, the actual performance of the base pipeline, based on different evidence sources, does not have much impact. In Table 5.3, we see that using CommandR+ with web search spiked performance from 60.02 for GPT-3.5 to 64.55 for 2-HOP claims. However, it again dropped for 3-HOP claims. For 4-HOP claims, all three evidence sources yield almost the same performance. This implies that the more complicated the claim, the less impact the evidence source has.

Table 5.3: F1 scores for the base pipeline and different evidence sources (GPT-3.5, Google snippets and CommandR+ with web search) on the HOVER dataset, separated by the number of hops.

Method	Reasoning	Evidence	HOVER		
			2-HOP	3-HOP	4-HOP
Base pipeline	GPT-3.5	GPT-3.5	60.02	55.66	49.52
		Google snippets	60.05	53.55	50.21
		CommandR+ w/ web search	64.55	52.47	49.89

5.1.3 Reasoner Comparison

We are also interested in seeing the performance of different reasoning backbones using the same source of evidence. We used the base pipeline with GPT-3.5 evidence and then GPT-3.5 and Mixtral-8x7b reasoning. For comparison, we used the same claim as in Figure 5.1 for Mixtral-8x7b. Note that we look at the quality of reasoning, regardless of the quality or truthfulness of the provided evidence.

At first sight, it is evident that Mixtral-8x7b (see Figure 5.4) generated many more questions than GPT-3.5. The answer to the second question was irrelevant to our claim, but the model then asked the same question with more details to get the needed evidence. It generated four questions, covering almost the whole claim. One last missing piece of information to verify was whether the theme song for *Civilization IV* was on the album *The Drop That Contained the Sea*.

Table 5.4 shows the performance for both reasoning LLMs. The difference between the reasoning capabilities is evident: Mixtral-8x7b vastly outperforms GPT-3.5. The results are similar for 2-HOP claims (60.02 and 61.25, respectively), but the more complicated the claim is, the more significant the difference. 60.14 vs 55.66 for 3-HOP and 58.94 vs 49.52 for 4-HOP. This means Mixtral-8x7b performance does not drop as significantly as GPT-3.5 with more hops, showing a solid reasoning capability.

Table 5.4: F1 scores for the base pipeline and different reasoning LLMs: GPT-3.5 and Mixtral-8x7b on the HOVER dataset, separated by the number of hops.

Method	Reasoning	Evidence	HOVER		
			2-HOP	3-HOP	4-HOP
Base pipeline	GPT-3.5	GPT-3.5	60.02	55.66	49.52
	Mixtral-8x7b	GPT-3.5	61.25	60.14	58.94

5.1. HOVER

The award-winning CD, that Jacqueline Horner-Kwiatek was a featured soloist on, includes the theme for the game "Civilization IV" written by the man who released The Drop That Contained the Sea which is a classical crossover album. This album was released in 2014.

True

The man who released the classical crossover album "The Drop That Contained the Sea" and wrote the theme for the game "Civilization IV" is Christopher Tin. The award-winning CD, which features Jacqueline Horner-Kwiatek as a soloist, was released in the year 2008. The classical crossover album "The Drop That Contained the Sea" was released on May 7, 2014. Jacqueline Horner-Kwiatek is identified as a featured soloist on the album "The Drop That Contained the Sea." Therefore, the final answer is: True.

Who is the man who released the classical crossover album "The Drop That Contained the Sea" and also wrote the theme for the game "Civilization IV"?

Christopher Tin

The man who released the classical crossover album "The Drop That Contained the Sea" and wrote the theme for the game "Civilization IV" is Christopher Tin. You can find more information about him on his Wikipedia page: https://en.wikipedia.org/wiki/Christopher_Tin

When was the award-winning CD, that Jacqueline Horner-Kwiatek was a featured soloist on, released?

2008

The award-winning CD that Jacqueline Horner-Kwiatek was a featured soloist on is called "Carols from the Old and New Worlds" and it was released in 2008. You can find more information about this CD on the Wikipedia page for Jacqueline Horner-Kwiatek: https://en.wikipedia.org/wiki/Jacqueline_Horner-Kwiatek

When was the classical crossover album "The Drop That Contained the Sea" released?

May 7, 2014

"The Drop That Contained the Sea" is a classical crossover album by composer Christopher Tin. It was released on May 7, 2014. For more information, you can refer to the Wikipedia article on the album: https://en.wikipedia.org/wiki/The_Drop_That_Contained_the_Sea

Was Jacqueline Horner-Kwiatek a featured soloist on the "The Drop That Contained the Sea" album?

Yes, Jacqueline Horner-Kwiatek was a featured soloist on the album "The Drop That Contained the Sea."

Yes, Jacqueline Horner-Kwiatek was a featured soloist on the album "The Drop That Contained the Sea." She is a member of the vocal ensemble Roomful of Teeth, which performed on the album. You can find more information about her and the album on the Wikipedia page for "The Drop That Contained the Sea": https://en.wikipedia.org/wiki/The_Drop_That_Contained_the_Sea

Figure 5.4: Base pipeline output for Mixtral-8x7b as the reasoning LLM with GPT-3.5 evidence.

5.1.4 Further Analysis

In this part of the work, we look at some additional aspects of the models, such as the number of generated questions, the impact of the question limit on performance or the qualitative analysis of the model outputs.

Average Number of Generated Questions

Another measure to show the quality of the claim verification and its reasoning can be the number of questions generated to verify a claim. We computed the average number of questions for various pipelines separately for 2-, 3- and 4-HOP claims.

Since there are more hops in a HOVER claim, the more complicated it is, and more evidence needs to be gathered to verify it. A hop corresponds to the number of Wikipedia articles required to verify it. Hence, a good reasoning model should be able to ask more questions with the increasing hops.

This number can also indicate how good a model is at producing simple questions instead of convoluted ones. Asking simple questions leads to retrieving clear evidence and reasoning over it for more successful verification.

We can see a few trends when we look at the measurements in Table 5.5. First, all base pipelines with GPT-3.5 show only a slight increase in the number of questions over the number of hops, and their values are very similar, all in the range of 2.14 to 2.34.

The base pipeline with Mixtral-8x7b as the reasoning LLM shows that it asks more questions on average and an apparent increase in the average questions number: 2.56, 2.82 and 3.39 for 2-4 hops. This further supports findings from the previous subsection 5.1.3 that Mixtral-8x7b is the superior LLM for reasoning in claim verification between these two models.

A positive effect can also be seen on the predicate pipeline, where the question number is around 0.5 questions higher than for the base pipeline. This suggests the predicate pipeline is better at decomposing the claim into simple questions.

Table 5.5: Average number of generated questions for different pipelines on the HOVER dataset, separated by the number of hops.

Method	Reasoning	Evidence	HOVER		
			2-HOP	3-HOP	4-HOP
Base pipeline	GPT-3.5	GPT-3.5	2.30	2.28	2.34
		Google	2.24	2.28	2.28
		CommandR+ w/ web search	2.14	2.29	2.32
	Mixtral-8x7b	GPT-3.5	2.56	2.82	3.39
Predicate pipeline	GPT-3.5	GPT-3.5	2.75	2.86	2.89

Limit on the Number of Questions

One of the parameters of the pipeline is the upper limit for how many questions can be asked before the final verdict has to be made. The value of this parameter from the QACheck paper is 5, and it was used for all the experiments.

We ran the base pipeline for $MAX_ROUND = 5, 6$ and 7 to see, if the model had the possibility to ask more questions, how it would impact the performance. Results in Table 5.6 show that for 2-HOP claims, increasing the parameter even decreases the performance. For 3-HOP, there is some positive impact: increase from 55.66 for 5 questions to 58.3 for 6 questions and 57.84 for 7 questions. This increase, however, does not have any effect on 4-HOP claims.

This suggests that past some point of the complexity of the claims, the number of generated questions does not help the performance, as the reasoning backbone cannot make sense of it. It would be interesting to see its effect on a pipeline with Mixtral-8x7b as the reasoner.

Table 5.6: F1 scores of the base pipeline with GPT-3.5 reasoning and evidence for 5, 6, and 7 maximum number of asked questions on the HOVER dataset.

Method	Reasoning	Evidence	Max questions	HOVER		
				2-HOP	3-HOP	4-HOP
Base pipeline	GPT-3.5	GPT-3.5	5	60.02	55.66	49.52
			6	59.5	58.3	49.37
			7	58.01	57.84	49.52

Reasoning Using Parametric Knowledge

When testing the pipeline, we noticed the following phenomenon. The claim *"Petr Pavel is the president of the Czech Republic."* is true as of the day of this experiment (5 June 2024). However, the GPT-3.5 model we use for reasoning was trained on data up to September 2021. Petr Pavel was inaugurated as the president of the Czech Republic on 9 March 2023, and before him, the president was named Miloš Zeman [64].

In Figure 5.5, we can see claim verification of the claim above with GPT-3.5 reasoner and evidence from CommandR+ with a web search. The retrieved proof is correct, and so is the answer to the question *"Who is the president of the Czech Republic?"*, Petr Pavel. However, the final prediction is that the claim is false.

We attribute this to the fact that the underlying "knowledge" of GPT-3.5 still "remembers" Miloš Zeman as the president. Hence, it "overrides" the provided evidence. This means the decision was not knowledge-grounded, as was the intention to provide external evidence to the model. Therefore, we should be cautious in saying that claim verification using LLMs can be knowledge-grounded, as there is no way to find out if the decision came from external evidence or within the reasoning LLM.



Figure 5.5: Example of the base pipeline completely ignoring the provided evidence and presumably using its internal “knowledge” to make the prediction.

5.2 Domain Specific Datasets

One of the essential parts of this thesis is to evaluate the models on claims from different domains. We must compare the performance of our models between them and establish LLM benchmarks for these datasets. So far, claim verification using LLM has been tested primarily on general datasets like HOVER or FEVER.

All three datasets of our choice (HealthFC, Climate-Fever, CoVert) have 3 or 4 labels: SUPPORTED, REFUTED, NOTENOUGHINFORMATION (NEI) and Climate-Fever even has a label DISPUTED. For this reason, we perform both binary classification (SUPPORTED vs. REFUTED and ternary classification (SUPPORTED vs. REFUTED vs. NOTENOUGHINFORMATION).

The pipelines described in chapter 4 are defined for binary classification. An adjusted pipeline was implemented to simulate predicting NEI. In the base pipeline, once the maximum number of questions was asked, the decision between the two classes was made based on the gathered evidence.

In the NEI pipeline, once the model hits the maximum number of questions, it goes to the reasoner one last time to ask whether we have enough information. If not, NEI is returned. Otherwise, a prediction is made based on the collected evidence.

We report results for both the binary and ternary classification for each dataset. In the case of the binary classification, claims from other classes are skipped. For the binary setting, three pipelines are tested: a base pipeline with GPT-3.5 evidence, a base pipeline with CommandR+ with web search evidence and a predicate pipeline. We only tested the base pipeline with GPT-3.5 evidence for the ternary setting.

5.2.1 HealthFC

The first dataset we look at is HealthFC, consisting of health-related claims manually verified using scientific papers. The claims were initially stated as questions but then transformed into claims as described in subsection 4.1.3.

Binary Classification

Results for the binary classification are reported in Table 5.7. The figures from the literature are taken from [65], where entailment prediction with DeBERTa was used to make the veracity prediction. The model was not fine-tuned on HealthFC; however, in one setting, it was provided with gold evidence to make the prediction and, in the other, with evidence retrieved from Wikipedia.

All three of our pipelines slightly surpass DeBERTa with Wikipedia evidence (F1-score 76.5). The best one is the predicate pipeline, with F1-score 79.22, followed by the base pipeline with GPT-3.5 evidence (78.22) and the base pipeline with CommandR+ evidence, with a score of 76.83. The good performance of the predicate pipeline could suggest that HealthFC claims are of a more complicated nature, as shown for HOVER 4-HOP claims.

On the other hand, DeBERTa, with the golden evidence, is the best-performing model in this task (F1-score 81.9). That is understandable, as it was provided with the "perfect" evidence.

Table 5.7: F1 score for the HealthFC dataset and different pipelines. DeBERTa numbers from [65].

Method	Reasoning	Evidence	HealthFC
DeBERTa-v3	-	Gold evidence	81.9
	-	Wikipedia	76.5
Base pipeline	GPT-3.5	GPT-3.5	78.22
		CommandR+ w/ web search	76.83
Predicate pipeline	GPT-3.5	GPT-3.5	79.22

Ternary Classification

For the three-class setting (Table 5.8), our base pipeline vastly underperforms both baselines from the original HealthFC paper [52]. There are two different methods utilising DeBERTa. The pipeline consists of two models: the first selects evidence sentences, and the second makes the prediction using the evidence. In the joint method, one model is trained for both evidence retrieval and veracity prediction tasks. For the veracity prediction, both were fine-tuned using the gold evidence.

The joint and pipeline DeBERTa set-up achieved F1-score of 67.5 and 65.1, respectively. Compared to our base pipeline of only 34.74. We also report the F1 scores for each class, where we can see that the base pipeline had the most success with the supported claims (47.92), then NEI claims (42.50), and drastically worst were the refuted claims, with a score of only 13.8.

Table 5.8: F1 Scores for HealthFC dataset for multi-class classification for classes REFUTED, SUPPORTED and NOTENOUGHINFO. DeBERTa model results were reported in [52].

Method	Reasoning	Evidence	HealthFC			
			F1-macro	REFUTED	SUPPORTED	NEI
DeBERTa joint	-	Scientific papers	67.5	-	-	-
DeBERTa pipeline	-	Scientific papers	65.1	-	-	-
Base pipeline	GPT-3.5	GPT-3.5	34.74	13.8	47.92	42.50

5.2.2 Climate-Fever

This section presents results for the Climate-Fever dataset. These claims were annotated based on Wikipedia articles.

Binary Classification

For this task, we could not find any results in the literature, so we are setting the baseline for binary classification on Climate-Fever. Table 5.10 shows that we achieved high F1-score with the base pipeline. 85.25 with evidence from CommandR+ and 84.85 with GPT-3.5 evidence. In this case, the predicate pipeline showed a much worse performance of 75.8.

Table 5.9: F1 score for the Climate-Fever dataset and different pipelines.

Method	Reasoning	Evidence	Climate-Fever
Base pipeline	GPT-3.5	GPT-3.5	84.85
		CommandR+ w/ web search	85.25
Predicate pipeline	GPT-3.5	GPT-3.5	75.8

Ternary Classification

The dataset has a fourth class DISPUTED, which the authors filtered out for evaluation, so the results are comparable to other major datasets like FEVER. Therefore, we also removed the class.

They evaluated their dataset on a pre-trained ALBERT, which was then fine-tuned on the FEVER dataset for the task of entailment prediction, and they used Wikipedia articles as the evidence source. Their paper reported only on class-wise F1-score, and we used those to calculate the macro average score.

Our base pipeline outperformed the baseline by more than 12 points (see in ??, where ALBERT achieved 36.05 F1-score and our pipeline 48.74. Looking at the class scores, the trend

is the same: the most successful class is SUPPORTED, then REFUTED and the last one is NEI. The most significant improvement was recorded for the SUPPORTED label, where ALBERT yielded a score of 47.79 and our base pipeline 67.54.

Table 5.10: F1 Scores for Climate-Fever dataset for multi-class classification for classes REFUTED, SUPPORTED and NOTENOUGHINFO. The results from ALBERT come from [51].

Method	Reasoning	Evidence	Climate-Fever			
			F1-macro	REFUTED	SUPPORTED	NEI
ALBERT	-	Wikipedia	36.05	41.81	47.79	18.57
Base pipeline	GPT-3.5	GPT-3.5	48.74	46.21	67.54	32.49

5.2.3 CoVERT

The last domain-specific dataset is CoVERT, a collection of very informal claims from Twitter on COVID-19. Annotators used Google to search for evidence to verify the claim.

Binary Classification

The original paper does not provide results for the binary classification; those are from [65]. The methods are the same as used for HealthFC and described in subsection 5.2.1.

Our base pipelines outperformed the DeBERTa models (83.4 for gold evidence and 82.5 for Wikipedia evidence). The pipeline with the evidence from GPT-3.5 achieved an F1-score of 85.64 and CommandR+ evidence of 83.51. These results are impressive because providing gold evidence makes the task much easier. However, the predicate pipeline again lagged behind the other methods. The figures are in Table 5.11.

Table 5.11: F1 score for the CoVert dataset and different pipelines. DeBERTa numbers from [65].

Method	Reasoning	Evidence	CoVert
DeBERTa	-	Gold evidence	83.4
DeBERTa	-	Wikipedia	82.5
Base pipeline	GPT-3.5	GPT-3.5	85.64
		CommandR+ w/ web search	83.51
Predicate pipeline	GPT-3.5	GPT-3.5	78.48

Ternary Classification

In the original paper, the three-class classification was performed using multi-layer perceptrons. MLP-FEVER was fine-tuned on the FEVER dataset, and the MLP-Evidence was fine-tuned using evidence and text pairs from CoVERT. Both are provided with gold evidence on inference.

Our base pipeline (48.45) performs slightly better than the MLP-FEVER method (46.00) (shown in Table 5.12). However, it still lags behind the fine-tuned MLP-Evidence with F1-score of 69.00. They do not report scores for each class, but our results show the best performance for the SUPPORTED label (68.25), then REFUTED (53.33) and the last one is NEI (23.78).

Table 5.12: F1 Scores for CoVert dataset for multi-class classification for classes REFUTED, SUPPORTED and NOTENOUGHINFO. MLP results come from [53].

Method	Reasoning	Evidence	CoVert			
			F1-macro	REFUTED	SUPPORTED	NEI
MLP-FEVER	-	Gold evidence	46.00	-	-	-
MLP-Evidence	-	Gold evidence	69.00	-	-	-
Base pipeline	GPT-3.5	GPT-3.5	48.45	53.33	68.25	23.78

5.3 FactKG

In this section, we look at the results from the KG pipeline, which were evaluated on the FactKG dataset (test split). We compare them to other methods from the literature, as well as the base pipeline with GPT-3.5 reasoning and evidence. There are many kinds of claims (as described in subsection 4.1.5), so there are results also for each category. Literature has reported the results using the accuracy measure only rather than the F1-score, so we provide both. We also discuss the model output and other aspects of claim verification on the FactKG dataset. The KG pipeline uses GPT-3.5 for reasoning.

5.3.1 Experimental Results

Table 5.13 shows results for the accuracy metric and Table 5.14 for F1-score. The (overlapping) claim categories are one-hop, conjunction, existence, multi-hop and negation. The results from the literature use BERT, GEAR (transformers) and KG-GPT(LLM approach).

BERT is trained on Wikipedia articles and then fine-tuned on the test set of FactKG ([54]). This model makes decisions based only on its internal knowledge. The GEAR model, on the other hand, enables reasoning over multiple texts. In the case of this task, it was trained to reason over graphical evidence, provided by 2 BERT models trained to retrieve subgraphs [54]. KG-GPT works as described in section 3.3.

In terms of the total accuracy (accuracy over the whole test split), our KG pipeline slightly outperformed the base pipeline (68.95% vs 67.84% respectively), while their position swapped for F1-score (60.87 vs 61.76 respectively). This shows the total performance is very similar between these two pipelines.

Compared to the models from the literature, GEAR and KG-GPT remain unbeaten by our models, with the accuracy of 77.65% and 72.68%, respectively. It is essential to remember that GEAR was fine-tuned for this task, and KG-GPT provided golden entities with the claim. In contrast, the BERT model yielded worse performance than our pipelines (65.20%).

We can also look at each type of claim. In general, one-hop claims seem the easiest, and multi-hop claims are the most difficult to verify. GEAR was best in all categories, except for multi-hop, where the BERT model performed the best with an accuracy of 70.06%.

Looking solely at our models, both accuracy and F1-score show the same trends. For the result analysis, we will therefore focus only on the accuracy. In one-hop and conjunction claims, the performance did not differ by much. The base pipeline achieved 71.54%, and the KG pipeline achieved 72.22% for one-hop claims and 71.08% and 71.02%, respectively, for conjunction. The only category where the base pipeline dominated was multi-hop claims, with 62.5% as opposed to 57.74% for the KG pipeline.

In contrast, the KG pipeline vastly outperformed the base pipeline in existence and negation claims. For existence, the difference was around 14% and in negation, 10%. With 75.59% accuracy, existence claims were the most successful category for the KG pipeline.

Also, for GEAR, model working with graph data, existence and negation were categories with the most significant differences from other methods. We can say that the KG evidence benefits these types of claims.

Table 5.13: Accuracy on the FactKG dataset by claim types and in total.

Method	Accuracy % on FactKG					
	Total	One-hop	Conjunction	Existence	Multi-hop	Negation
BERT	65.20	69.64	63.31	61.84	70.06	63.62
GEAR	77.65	83.23	77.68	81.61	68.84	79.41
KG-GPT	72.68	-	-	-	-	-
Base pipeline	67.84	71.54	71.08	61.35	62.5	59.97
KG pipeline	68.95	72.22	71.02	75.59	57.74	69.93

5.3.2 Output Analysis

To demonstrate how the KG pipeline works, we chose the claim *"Well the Acura TLX has no V6 engine and was not assembled in Marysville Ohio."* This claim belongs to the categories *negation* and *conjunction*. Two pieces of evidence are needed to verify this. The verification process is shown in Table 5.15.

Table 5.14: F1-score on the FactKG dataset by claim types and in total.

Method	F1-score on FactKG					
	Total	One-hop	Conjunction	Existence	Multi-hop	Negation
Base pipeline	61.76	67.18	59.03	62.59	58.56	51.3
KG pipeline	60.87	69.13	59.16	72.60	43.19	70.76

The first generated question is whether the Acura TLX has a V6 engine. Entities found within this question were the V6 engine and Acura TLX. 14 relations from the graph were selected, and the LLM then chose *engine* as the most semantically similar to the question. With this relation, we got two evidence tuples: ("*Acura_TLX*", "*engine*", 2.4) and ("*Acura_TLX*", "*engine*", 3.5). Evidence sentence from these tuples was formed and used to answer the question. The answer says "*Yes, the Acura TLX does have a V6 engine available in the 3.5 model.*" However, the evidence does not mention whether it is a V6 engine or not. This piece of information was "added" by the LLM and is not grounded in external knowledge.

In the second round, the next question asks for the remaining piece of evidence, whether Acura TLX was assembled in Marysville, Ohio. In this question, 3 entities were identified: Marysville, Ohio; Marysville Motorcycle Plant and Acura TLX. 33 relations related to those entities were retrieved from the graph. *Assembly* was picked by GPT-3.5 as most representing of the question, and then two tuples retrieved: ("*Acura_TLX*", "*assembly*", "*United States: Marysville, Ohio*") and ("*Acura_TLX*", "*assembly*", "*Guangzhou, China*"). The evidence tuples were turned into natural language sentences and used as evidence. The answer, stemming from the evidence, is positive.

After gathering these two question-answer pairs, the model made the final prediction. Concatenating the evidence, the label is REFUTED. The final reasoning was correct, given the collected evidence, as the car does, in fact, have a V6 engine and was assembled in Marysville, Ohio.

5.3.3 Further Analysis

Both the dataset and the KG pipeline have some specifics from which stem scenarios in which the claim verification might or will fail.

Failure to Identify Entities

One problem occurs when both the Spotlight API and LLM do not identify any entity in the question. This renders the claim false by default, as no evidence could be found to support the claim. This situation happened for 5.8% of the test claims. Some questions with no identified entity are: "*Did Mobyland have a successor?*", "*Did Jeffrey A. Hart have a spouse?*" or "*Is Death of a Train published in Hardcover?*".

Table 5.15: Example of claim verification with the KG pipeline. For each round there is a generated question, DBPedia URIs extracted from the question, retrieved relations from the graph using the entities, selected evidence tuples, rationale generated from the tuples and the answer. At the end, the final prediction is made.

Claim: Well the Acura TLX has no V6 engine and was not assembled in Marysville Ohio.

Round 1	
Question	Does the Acura TLX have a V6 engine?
DBPedia URI	http://dbpedia.org/resource/V6_engine http://dbpedia.org/resource/Acura_TLX
Retrieved relations	manufacturer, class, modelYears, layout, width, aka, productionStartYear, designer, related, engine, predecessor, assembly, bodyStyle, transmission
Retrieved tuples	("Acura_TLX", "engine", 3.5) ("Acura_TLX", "engine", 2.4)
Rationale	The Acura TLX has engines available in 3.5 and 2.4.
Answer	Yes, the Acura TLX does have a V6 engine available in the 3.5 model.
Round 2	
Question	Was the Acura TLX assembled in Marysville, Ohio?
DBPedia URI	http://dbpedia.org/resource/Marysville,_Ohio http://dbpedia.org/resource/Marysville_Motorcycle_Plant http://dbpedia.org/resource/Acura_TLX
Retrieved relations	class, website, areaTotal, bodyStyle, populationDensity, subdivisionName, utcOffset, modelYears, areaCode, type, productionStartYear, areaWater, designer, related, predecessor, assembly, elevation, timeZone, populationTotal, engine, width, settlementType, nickname, transmission, manufacturer, layout, areaLand, location, leaderTitle, leaderName, postalCode, aka, governmentType
Retrieved tuples	("Acura_TLX", "assembly", "United States: Marysville, Ohio") ("Acura_TLX", "assembly", "Guangzhou, China")
Rationale	The Acura TLX is assembled in the United States in Marysville, Ohio and in Guangzhou, China.
Answer	Yes.
Prediction	The Acura TLX has a V6 engine available in the 3.5 model. The Acura TLX was assembled in Marysville, Ohio. Therefore, the final answer is: False.

Unverifiable Claims

The dataset contains a group of claims which are of very general terms or do not contain any subject. Such claims are: *"it is 84 metres above sea level and serves the city of Fallujah!"*, *"A musical artist is a singer in the Guaranian genre?"* or *"He was born in Paraguay, and died in Asuncion!"*. These claims have no subject and only serve to prove the existence of an entity fulfilling the description. However, they are almost meaningless in real life.

The KG pipeline cannot verify This type of claim (always returns the REFUTED label), as the graph retrieval phase cannot retrieve such evidence. To verify the last claim, we would have to look at all entities to which Paraguay is connected as a subject. There would be many triples and relations to retrieve and filter through. Therefore, this direction of reasoning is not supported in the KG pipeline.

This design decision also means that many multi-hop claims cannot be verified, which might be a reason for such a bad performance. Although *"I wish that an airport is 84 metres above sea level and serves the city of Fallujah."* makes sense as a claim to be verified, the name of the airport, if there is any, will never be retrieved, unless there exists a tuple ('Fallujah', 'hasAirport', some airport).

Question Claims

There are also claims stated as questions. The test set has 602 claims in this category. It is not a reasoning type but rather a style type. One of these claims is *"When did Deportivo Toluca F.C. operate Agra Airport?"* They might not seem like claims or verifiable, but the goal here is to verify if the Deportivo Tulca F.C. really operated Agra Airport, not when it did that.

Our pipeline is not explicitly equipped to correctly interpret these claims. However, it achieved an accuracy of 72.92% and an F1-score of 64.48, which is higher than the total performance.

5.4 Question Coverage Survey

We conducted a survey as described in subsection 4.3.2. It was implemented as a Google Form and distributed among annotators. 12 annotators took part. Results for both categories are in Table 5.16.

For each annotator, the average rank of each model is computed. All three models use GPT-3.5 evidence. Model 1 is the base pipeline with GPT-3.5 reasoning; Model 2 is the predicate pipeline with GPT-3.5 reasoning; and Model 3 is the base pipeline with Mixtral-8x7b reasoning. These averages are then averaged again, getting the Mean Average Rank (MAR).

The category coverage (Table 5.17) measures how well the question set covers all the information needed to verify a claim. We can see the best ranks pipeline Model 3 (MAR 1.6), then Model 1 (MAR 1.85) and Model 2 (MAR 2.13). Results for the overall category (Table 5.18) are very similar. The order remains the same; Model 3 slightly decreased the rank (1.71) while Model 1 (1.81) and 2 (2.03) increased. Based on these rankings, reasoning with Mixtral-8x7b produces the highest quality question sets.

Table 5.16: Results for the question coverage survey in two categories: coverage and overall quality. Model 1 is the base pipeline with GPT-3.5 reasoner. Model 2 is the predicate pipeline with GPT-3.5 reasoner. Model 3 is the base pipeline with Mixtral-8x7b reasoner. Each row represents one annotator and their average ranking for each model. These are then averaged across the annotators (Mean Average Rank).

Table 5.17: Average ranks for coverage.				Table 5.18: Average ranks for overall.			
Annotator	Model 1	Model 2	Model 3	Annotator	Model 1	Model 2	Model 3
Coverage				Overall			
1	2,17	2,00	1,83	1	2,00	2,17	1,83
2	2,17	1,67	1,83	2	1,83	1,83	1,83
3	1,50	1,67	1,33	3	1,83	1,83	1,50
4	1,83	2,33	1,83	4	2,00	2,17	1,83
5	1,83	2,00	1,50	5	2,00	1,67	1,67
6	2,00	2,50	1,50	6	2,00	2,17	1,83
7	2,00	2,33	1,83	7	1,50	2,17	2,17
8	1,67	2,50	1,83	8	1,67	2,17	2,17
9	1,67	2,50	1,33	9	1,50	2,17	1,00
10	1,83	2,33	1,33	10	2,00	2,67	1,33
11	1,33	1,83	1,67	11	1,17	1,83	2,00
12	2,17	1,83	1,33	12	2,17	1,50	1,33
Avg	1,85	2,13	1,60	Avg	1,81	2,03	1,71

6 Discussion

In this chapter, we discuss the results and limitations and outline the potential direction for future research.

6.1 Key Findings

The base pipeline outperformed other methods, utilizing LLMs for claim verification either step-by-step or directly prompting with the claim for 2- and 3-HOP claims, including its “parent” method QACheck. This is probably due to some changes between the two models, like different LLMs being used, slightly different architecture, and different prompts. However, the model FOLK remained unbeaten for 4-HOP claims.

Incorporating predicates into the pipeline was expected to boost performance the same way it happened in FOLK [5]. In fact, the results of the predicate pipeline lagged behind the base pipeline, except for the 4-HOP claims. The predicate pipeline also generates more questions on average on the base pipeline, suggesting the strength of structured reasoning lies in the most involved claims. This can also be observed for FOLK, which shows the biggest gap between other models in 4-HOP claims.

In addition to reasoning, LLMs can be used to retrieve evidence. Examining different evidence sources (GPT-3.5, CommandR+ with web search and Google), we saw that the choice of evidence did not have a great effect for more complicated claims. On the other hand, the choice of reasoning LLM showed that using Mixtral-8x7b boosted the performance even for the most complex claims, as opposed to GPT-3.5. We can see the impact of using different reasoning LLMs on the number of generated questions; Mixtral-8x7b generates more questions, leading to higher-quality explanations for the user.

We ran binary and ternary classification on three domain-specific datasets: HealthFC, Climate-Fever and CoVert. The most difficult was HealthFC for both binary and ternary classification. It was the only dataset out of three where the predicate pipeline improved the performance, only further suggesting it is successful with the most challenging claims.

Compared to the HOVER dataset, the domain-specific claims are easier to verify. We can conclude that real-life claims are not as complicated as the ones in HOVER and do not really reflect the reality of fact-checking. Furthermore, we showed the pipelines are robust towards the style of the claims, where it handles both formal (Climate-Fever) and very informal claims (CoVERT) or even questions (FactKG).

Experiments with different domains also showed our LLM-based domain agnostic model, with no training or fine-tuning, outperformed transformer models trained for the task with gold labels and gold evidence. With both HealthFC and CoVert, the base pipeline beat

DeBERTa trained for entailment prediction and provided with Wikipedia evidence (for CoVert even with gold evidence) in the inference step. Similarly, the base pipeline outperformed ALBERT on Climate-Fever, which was fine-tuned on the FEVER dataset and provided with Wikipedia evidence.

This shows that our model has competitive results without needing large amounts of labelled data and labelled evidence. The adjustment for 3 classes can also approximate the situation when not enough evidence is found. Furthermore, we set the baseline for binary classification on Climate-Fever.

The knowledge graph claim dataset FactKG was evaluated on two pipelines: the base one and KG pipeline. The KG pipeline slightly outperformed the base pipeline in accuracy (but not F1-score). It also beat BERT fine-tuned on train data. GEAR and KG-GPT achieved the best results. However, the former was fine-tuned for this task, and KG-GPT provided gold entities with the claims, while the KG pipeline works without any gold evidence.

Categories where evidence from KG seems to have the most significant impact are existence and negation claims. We can see that for the KG pipeline and GEAR. This can be attributed to the high recall of retrieval from knowledge graphs: if there is a relation between two entities, it will be retrieved. If there is not, nothing is returned. Meanwhile, LLMs can hallucinate, and even if there is a relation, they might not "remember" it, or when there is not one, they might "hallucinate" it. So, knowledge graphs can, indeed, improve performance.

One of the reasons for KG-GPT having such an advantage over our KG pipeline could be that they worked with a DBpedia dump, limited to just those entities and relations relevant to the claims in FactKG. By using this, they solved the problem of unverifiable claims described in subsection 5.3.3. On the other hand, we used the online DBpedia, as it simulates the open-world setting better and is not as demanding regarding resources.

The question coverage survey tested how well different models generate questions to cover the whole claim. In both categories (coverage and overall), the best-performing pipeline was the base pipeline with Mixtral-8x7b as the reasoner, then the base pipeline with GPT-3.5 and the last one was the predicate pipeline with GPT-3.5 reasoner. This qualitative assessment supports the findings from the quantitative results, which is that the Mixtral-8x7b model is superior as the reasoning backbone of the model. It not only provides only the most accurate veracity labels but also the most user-friendly outputs, which serve for better explainability.

6.2 Limitations and Future Work

Mixtral-8x7b turned out to be the best LLM for reasoning. However, because of resources and costs, we did not have the opportunity to test it on other datasets or the predicate pipeline. We could then see how it deals with structured reasoning or different kinds of language. Since the choice of the reasoner turned out to be so essential, future work could focus on testing some more recent models, such as GPT-4. We have tested it with our approach, and the intermediate results looked promising. Furthermore, testing Mixtral-8x7b reasoning with different sources of evidence could amplify the actual differences between the quality of various evidence sources.

One of the biggest limitations of this work was the graph retrieval. Future efforts should focus on implementing fast and efficient graph retrieval without having to have a static dump of the whole DBPedia. Other knowledge graphs exist, such as UMLS for medical information. They could be explored for the use of claim verification, too.

It should also be examined to what extent LLMs decide to ignore provided evidence when making a decision. This tendency might vary between different models, and understanding these mechanisms and choosing the best model would lead to better claim verification.

Incorporating structured reasoning in the form of predicates into the predicate pipeline did not boost the performance as shown in [5]. Exploring the reasons why it worked in one setting and not in another could uncover potential improvements for our highly explainable pipeline.

We have not dedicated too much time for qualitative analysis of the outputs for different domains or the predicate pipeline. Getting some more insights into their differences would require more advanced techniques or more manual labour.

Similarly, more than 6 examples should be used to make the survey results more robust. The paper we followed for this procedure used 30 examples with just three annotators. However, the task of assessing the coverage and overall quality of the questions demands quite some time and a lot of patience and focus. The respondents had problems finishing even this short survey.

7 Conclusion

In this thesis, we investigated the use of large language models (LLMs) for claim verification. We developed three highly modular pipelines for explainable, step-by-step, question-led claim verification based on methods from the literature: the base pipeline, the predicate pipeline, and the knowledge graphs pipeline. The explainability is supported by presenting the outputs in a WebGUI. We also extended the pipelines to classify the third label, NOTENOUGHINFORMATION.

Our experiments involved various evidence sources and different LLMs for reasoning. We evaluated the pipelines on real-world, domain-specific claims, including medical claims, claims related to climate change, and COVID-19-related claims. Additionally, we conducted a survey to assess the quality of outputs from different pipelines.

Here, we restate the research questions and our conclusions:

RQ1 How can the use of LLMs help claim verification?

LLMs can assist in multiple steps of the claim verification process. They can generate simple questions to verify a claim, simulating the workflow of a human fact-checker. LLMs can help with gathering evidence either by retrieving it from their parametric memory or by processing multiple texts and returning relevant evidence. For knowledge graph (KG) retrieval, they can extract relevant evidence triples from a retrieved subgraph. Based on the claim, generated questions, and gathered evidence, LLMs can determine the veracity label. This step-by-step process and its intermediate outputs also serve as explanations for the user. Unlike traditional approaches that depend on training and fine-tuning models, this approach does not require large amounts of labeled claims and evidence corpora, as it learns from a few demonstrations through in-context learning. In many instances, our pipelines outperformed models fine-tuned for this task.

RQ2 Does leveraging knowledge from knowledge graphs and structured reasoning improve performance?

The performance of the predicate pipeline decreased for most datasets compared to the base pipeline, although it improved for the two most difficult datasets. This observation, along with the performance of FOLK [5], suggests that structured reasoning might have a positive impact, particularly for the most complex claims. Reasoning over knowledge graphs is another form of structured reasoning. Although the overall performance of the KG pipeline was comparable to the base pipeline, we identified types of claims where utilizing knowledge from knowledge graphs boosted performance due to their high recall. Therefore, in some instances, knowledge graphs can enhance the performance.

RQ3 How do different domains compare in this task?

The HOVER dataset was the most challenging, followed by medical claims. Claims

related to climate change and COVID-19 exhibited similar performance for both binary and ternary classification. For these categories we achieved results surpassing those reported in literature. Generally, real-life claims are less complex and less unnaturally structured than those in HOVER, requiring fewer reasoning steps to be verified.

Further experiments and analysis suggest that the quality of our step-by-step claim verification heavily depends on the underlying LLM rather than the source of evidence. This was demonstrated through quantitative and qualitative analysis, as well as a user survey on the quality of generated questions.

We showcased numerous applications of LLMs in explainable claim verification, the effects of structured reasoning, and the use of knowledge graphs as a source of evidence. Until now, domain-specific datasets were evaluated mainly using traditional methods. We identified the characteristics, strengths, and pitfalls of these approaches, outlining directions for future research and improvements in explainable claim verification.

List of Figures

2.1	Steps in the claim verification pipeline: document retrieval, evidence selection and verdict prediction [26].	6
2.2	Examples of zero-shot (a), one-shot (b) and few-shot learning (c) [3].	7
2.3	Example of a simple knowledge graph representing living creatures and their relationships [41].	8
2.4	Example of a SPARQL code to retrieve genres of work of the author of Tokyo Mew Mew [43].	9
3.1	QACheck pipeline [4].	11
3.2	Claim verification using FOLK [5].	13
3.3	Claim verification using KG-GPT [7].	14
4.1	Types of claims and their examples in the HOVER dataset [50].	16
4.2	Types of claims and their examples in FactKG [54].	18
4.3	Workflow of our step-by-step claim verification.	19
4.4	2 of 10 demonstrations for generating the first question, based on prompts from QACheck [4].	20
4.5	2 out of 10 demonstrations for generating the follow-up question, based on prompts from QACheck [4].	20
4.6	2 out of 10 demonstrations for the verifier module, based on prompts from QACheck [4]. LLM decides if we have enough evidence to make the final veracity prediction.	22
4.7	2 out of 10 demonstrations for question generation in the predicate pipeline. Each generated question is accompanied by a predicate defining the question and a simple instruction on what to verify. Based on prompts from QACheck [4] and FOLK [5].	24
4.8	1 out of 10 demonstrations of follow-up question generation for the predicate pipeline. As already gathered, evidence and predicate from the previous question are present. Based on prompts from QACheck [4] and FOLK [5].	25
4.9	1 out of 10 demonstrations for the verification module with predicates. Triples (question, answer, predicate) are used as evidence. Based on prompts from QACheck [4] and FOLK [5].	25
4.10	A demonstration for the reasoning module with predicates. Directly taken from FOLK [5].	27

4.11	2 out of 14 demonstrations for question generation in the KG pipeline based on the claim used in sentence segmentation prompt in KG-GPT [7] and QACheck [4].	28
4.12	Intermediate steps in the question answering module in the KG pipeline.	28
4.13	2 out of 10 demonstrations for relation selection based on prompts from KG-GPT [7].	30
4.14	Claim verification screen with the input field for the claim and the option to build a custom pipeline with chosen LLMs and methods.	32
4.15	Presentation of the claim verification results. At the top is the verified claim, the verdict and the explanation in natural language. The second part of the screen shows the questions asked to get to the verdict, answers to these questions, and evidence with sources for the answers.	33
5.1	Example outputs of the base pipeline with GPT-3.5 reasoning and evidence.	36
5.2	Example outputs and predicates with prediction for the predicate pipeline.	39
5.3	Retrieved evidence from different evidence sources for the question <i>What is the name of the British sitcom co-written by Sian Gibson?</i>	40
5.4	Base pipeline output for Mixtral-8x7b as the reasoning LLM with GPT-3.5 evidence.	42
5.5	Example of the base pipeline completely ignoring the provided evidence and presumably using its internal "knowledge" to make the prediction.	45

List of Tables

5.1	F1 scores of the base pipeline and other models (taken from [4]) on the HOVER dataset.	37
5.2	F1 scores of the base pipeline, predicate pipeline and other models (FOLK from [5] and QACheck from [4]) on the HOVER dataset.	38
5.3	F1 scores for the base pipeline and different evidence sources (GPT-3.5, Google snippets and CommandR+ with web search) on the HOVER dataset, separated by the number of hops.	41
5.4	F1 scores for the base pipeline and different reasoning LLMs: GPT-3.5 and Mixtral-8x7b on the HOVER dataset, separated by the number of hops.	41
5.5	Average number of generated questions for different pipelines on the HOVER dataset, separated by the number of hops.	43
5.6	F1 scores of the base pipeline with GPT-3.5 reasoning and evidence for 5, 6, and 7 maximum number of asked questions on the HOVER dataset.	44
5.7	F1 score for the HealthFC dataset and different pipelines. DeBERTa numbers from [65].	46
5.8	F1 Scores for HealthFC dataset for multi-class classification for classes REFUTED, SUPPORTED and NOTENOUGHINFO. DeBERTa model results were reported in [52].	47
5.9	F1 score for the Climate-Fever dataset and different pipelines.	47
5.10	F1 Scores for Climate-Fever dataset for multi-class classification for classes REFUTED, SUPPORTED and NOTENOUGHINFO. The results from ALBERT come from [51].	48
5.11	F1 score for the CoVert dataset and different pipelines. DeBERTa numbers from [65].	48
5.12	F1 Scores for CoVert dataset for multi-class classification for classes REFUTED, SUPPORTED and NOTENOUGHINFO. MLP results come from [53].	49
5.13	Accuracy on the FactKG dataset by claim types and in total.	50
5.14	F1-score on the FactKG dataset by claim types and in total.	51
5.15	Example of claim verification with the KG pipeline. For each round there is a generated question, DBPedia URIs extracted from the question, retrieved relations from the graph using the entities, selected evidence tuples, rationale generated from the tuples and the answer. At the end, the final prediction is made.	52

5.16	Results for the question coverage survey in two categories: coverage and overall quality. Model 1 is the base pipeline with GPT-3.5 reasoner. Model 2 is the predicate pipeline with GPT-3.5 reasoner. Model 3 is the base pipeline with Mixtral-8x7b reasoner. Each row represents one annotator and their average ranking for each model. These are then averaged across the annotators (Mean Average Rank).	54
5.17	Average ranks for coverage.	54
5.18	Average ranks for overall.	54

Bibliography

- [1] E. Calvo-Gutiérrez and C. Marín-Lladó. “Combatting Fake News: A Global Priority Post COVID-19”. In: *Societies* 13.7 (2023). ISSN: 2075-4698. DOI: 10.3390/soc13070160. URL: <https://www.mdpi.com/2075-4698/13/7/160>.
- [2] G. Bekoulis, C. Papagiannopoulou, and N. Deligiannis. *A Review on Fact Extraction and Verification*. 2021. arXiv: 2010.03001 [cs.CL].
- [3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].
- [4] L. Pan, X. Lu, M.-Y. Kan, and P. Nakov. *QACHECK: A Demonstration System for Question-Guided Multi-Hop Fact-Checking*. 2023. arXiv: 2310.07609 [cs.CL].
- [5] H. Wang and K. Shu. *Explainable Claim Verification via Knowledge-Grounded Reasoning with Large Language Models*. 2023. arXiv: 2310.05253 [cs.CL].
- [6] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. “Survey of Hallucination in Natural Language Generation”. In: *ACM Computing Surveys* 55.12 (Mar. 2023), pp. 1–38. ISSN: 1557-7341. DOI: 10.1145/3571730. URL: <http://dx.doi.org/10.1145/3571730>.
- [7] J. Kim, Y. Kwon, Y. Jo, and E. Choi. *KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models*. 2023. arXiv: 2310.11220 [cs.CL].
- [8] J. H. Kuklinski, P. J. Quirk, J. Jerit, D. Schwieder, and R. F. Rich. “Misinformation and the Currency of Democratic Citizenship”. In: *The Journal of Politics* 62.3 (2000), pp. 790–816. DOI: 10.1111/0022-3816.00033. eprint: <https://doi.org/10.1111/0022-3816.00033>. URL: <https://doi.org/10.1111/0022-3816.00033>.
- [9] A. V. Cass R. Sunstein. “Conspiracy Theories: Causes and Cures”. In: *The Journal of Political Philosophy* 17 (2 2009), pp. 202–227. DOI: <https://doi.org/10.1111/j.1467-9760.2008.00325.x>. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1467-9760.2008.00325.x?saml_referrer.
- [10] D. Tambini. *Fake News: Public Policy Responses*. London, 2017. URL: <http://eprints.lse.ac.uk/id/eprint/73015>.

- [11] J. L. Nelson and H. Taneja. "The small, disloyal fake news audience: The role of audience availability in fake news consumption". In: *New Media & Society* 20.10 (2018), pp. 3720–3737. DOI: 10.1177/1461444818758715. eprint: <https://doi.org/10.1177/1461444818758715>. URL: <https://doi.org/10.1177/1461444818758715>.
- [12] H. Marshall and A. Drieschova. "Post-Truth Politics in the UK's Brexit Referendum". In: *New Perspectives* 26.3 (2018), pp. 89–106. ISSN: 2336825X, 23368268. URL: <https://www.jstor.org/stable/26675075> (visited on 04/01/2024).
- [13] A. Bovet and H. A. Makse. "Influence of fake news in Twitter during the 2016 US presidential election". In: *Nature Communications* 10 (2019), p. 7. DOI: 10.1038/s41467-018-07761-2. URL: <https://doi.org/10.1038/s41467-018-07761-2>.
- [14] S. Vosoughi, D. Roy, and S. Aral. "The spread of true and false news online". In: *Science* 359.6380 (2018), pp. 1146–1151. DOI: 10.1126/science.aap9559. eprint: <https://www.science.org/doi/pdf/10.1126/science.aap9559>. URL: <https://www.science.org/doi/abs/10.1126/science.aap9559>.
- [15] F. Adebessin, H. Smuts, T. Mawela, G. Maramba, and M. Hattingh. "The Role of Social Media in Health Misinformation and Disinformation During the COVID-19 Pandemic: Bibliometric Analysis". In: *JMIR Infodemiology* 3 (Sept. 2023), e48620. DOI: 10.2196/48620.
- [16] M. M. Ferreira Caceres, J. P. Sosa, J. A. Lawrence, C. Sestacovschi, A. Tidd-Johnson, M. H. U. Rasool, V. K. Gadamidi, S. Ozair, K. Pandav, C. Cuevas-Lou, M. Parrish, I. Rodriguez, and J. P. Fernandez. "The impact of misinformation on the COVID-19 pandemic". In: *AIMS Public Health* 9.2 (Jan. 2022), pp. 262–277. DOI: 10.3934/publichealth.2022018.
- [17] G. K. Shahi, A. Dirkson, and T. A. Majchrzak. "An exploratory study of COVID-19 misinformation on Twitter". In: *Online Social Networks and Media* 22, 33623836 (Mar. 2021), p. 100104. DOI: 10.1016/j.osnem.2020.100104.
- [18] Y. M. Rocha, G. A. de Moura, G. A. Desidério, C. H. de Oliveira, F. D. Lourenço, and L. D. de Figueiredo Nicolete. "The impact of fake news on social media and its influence on health during the COVID-19 pandemic: a systematic review". In: *Zeitschrift für Gesundheitswissenschaften*, 34660175 (Oct. 2021), pp. 1–10. DOI: 10.1007/s10389-021-01658-z.
- [19] A. Al-Rawi, D. O'Keefe, O. Kane, and A.-J. Bizimana. "Twitter's Fake News Discourses Around Climate Change and Global Warming". In: *Frontiers in Communication* 6 (2021). ISSN: 2297-900X. DOI: 10.3389/fcomm.2021.729818. URL: <https://www.frontiersin.org/articles/10.3389/fcomm.2021.729818>.
- [20] J. Jerit and Y. Zhao. "Political Misinformation". In: *Annual Review of Political Science* 23 (2020). DOI: <https://doi.org/10.1146/annurev-polisci-050718-032814>. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-polisci-050718-032814#html_fulltext.

- [21] E. Porter and T. J. Wood. “The global effectiveness of fact-checking: Evidence from simultaneous experiments in Argentina, Nigeria, South Africa, and the United Kingdom”. In: *Proceedings of the National Academy of Sciences* 118.37 (2021), e2104235118. DOI: 10.1073/pnas.2104235118. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.2104235118>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.2104235118>.
- [22] PolitiFact. *The Principles of the Truth-O-Meter: PolitiFact’s methodology for independent fact-checking*. 2024. URL: <https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/#Truth-O-Meter%20ratings> (visited on 04/04/2024).
- [23] A. Vlachos and S. Riedel. “Fact Checking: Task definition and dataset construction”. In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Ed. by C. Danescu-Niculescu-Mizil, J. Eisenstein, K. McKeown, and N. A. Smith. Baltimore, MD, USA: Association for Computational Linguistics, June 2014, pp. 18–22. DOI: 10.3115/v1/W14-2508. URL: <https://aclanthology.org/W14-2508>.
- [24] Z. Guo, M. Schlichtkrull, and A. Vlachos. “A Survey on Automated Fact-Checking”. In: *Transactions of the Association for Computational Linguistics* 10 (2022). Ed. by B. Roark and A. Nenkova, pp. 178–206. DOI: 10.1162/tacl_a_00454. URL: <https://aclanthology.org/2022.tacl-1.11>.
- [25] Z. Zhang, J. Li, F. Fukumoto, and Y. Ye. “Abstract, Rationale, Stance: A Joint Model for Scientific Claim Verification”. In: *CoRR* abs/2110.15116 (2021). arXiv: 2110.15116. URL: <https://arxiv.org/abs/2110.15116>.
- [26] J. Vladika and F. Matthes. *Scientific Fact-Checking: A Survey of Resources and Approaches*. 2023. arXiv: 2305.16859 [cs.CL].
- [27] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. “Generating Fact Checking Explanations”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 7352–7364. DOI: 10.18653/v1/2020.acl-main.656. URL: <https://aclanthology.org/2020.acl-main.656>.
- [28] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. *FEVER: a large-scale dataset for Fact Extraction and VERification*. 2018. arXiv: 1803.05355 [cs.CL].
- [29] R. Gozalo-Brizuela and E. C. Garrido-Merchán. *A survey of Generative AI Applications*. 2023. arXiv: 2306.02781 [cs.LG].
- [30] OpenAI, J. Schulman, B. Zoph, C. Kim, J. Hilton, J. Menick, J. Weng, J. F. C. Uribe, L. Fedus, L. Metz, M. Pokorny, R. G. Lopes, S. Zhao, A. Vijayvergiya, E. Sigler, A. Perelman, C. Voss, M. Heaton, J. Parish, D. Cummings, ..., and C. Hesse. “Introducing ChatGPT”. In: *OpenAI* (Nov. 22, 2022). URL: <https://openai.com/blog/chatgpt#OpenAI> (visited on 04/14/2024).
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

- [32] A. Galassi, M. Lippi, and P. Torrioni. “Attention in Natural Language Processing”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.10 (Oct. 2021), pp. 4291–4308. ISSN: 2162-2388. DOI: 10.1109/tnnls.2020.3019893. URL: <http://dx.doi.org/10.1109/TNNLS.2020.3019893>.
- [33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].
- [34] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG].
- [35] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui. *A Survey on In-context Learning*. 2023. arXiv: 2301.00234 [cs.CL].
- [36] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, ..., and B. Zoph. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].
- [37] D. Milmo and agency. “Two US lawyers fined for submitting fake court citations from ChatGPT”. In: *The Guardian* (June 23, 2023). URL: <https://www.theguardian.com/technology/2023/jun/23/two-us-lawyers-fined-submitting-fake-court-citations-chatgpt> (visited on 04/14/2024).
- [38] A. Hogan, E. Blomqvist, M. Cochez, C. D’amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. “Knowledge Graphs”. In: *ACM Computing Surveys* 54.4 (July 2021), pp. 1–37. ISSN: 1557-7341. DOI: 10.1145/3447772. URL: <http://dx.doi.org/10.1145/3447772>.
- [39] *Resource Description Framework (RDF) Model and Syntax Specification*. 1999. URL: <https://www.w3.org/TR/PR-rdf-syntax/Overview.html> (visited on 04/14/2024).
- [40] T. R. Gruber. “A translation approach to portable ontology specifications”. In: *Knowledge Acquisition* 5.2 (1993), pp. 199–220. ISSN: 1042-8143. DOI: 10.1006/knac.1993.1008. URL: <https://doi.org/10.1006/knac.1993.1008>.
- [41] *Knowledge graph*. URL: https://en.wikipedia.org/wiki/Knowledge_graph (visited on 04/14/2024).
- [42] *SPARQL 1.1 Query Language*. 2013. URL: <https://www.w3.org/TR/sparql11-query/> (visited on 04/14/2024).
- [43] *DBpedia*. URL: <https://en.wikipedia.org/wiki/DBpedia> (visited on 04/14/2024).
- [44] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer. “DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia”. In: *Semantic Web Journal* 6 (Jan. 2014). doi: 10.3233/SW-140134.

- [45] O. Bodenreider. “The Unified Medical Language System (UMLS): integrating biomedical terminology”. In: *Nucleic Acids Research* 32.Database issue (Jan. 2004), pp. D267–D270. DOI: 10.1093/nar/gkh061.
- [46] Z. Shokrzadeh, M.-R. Feizi-Derakhshi, M.-A. Balafar, and J. Bagherzadeh Mohasefi. “Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding”. In: *Ain Shams Engineering Journal* 15.1 (2024), p. 102263. ISSN: 2090-4479. DOI: 10.1016/j.asej.2023.102263. URL: <https://doi.org/10.1016/j.asej.2023.102263>.
- [47] S. Srivastava, M. Patidar, S. Chowdhury, P. Agarwal, I. Bhattacharya, and G. Shroff. “Complex Question Answering on knowledge graphs using machine translation and multi-task learning”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by P. Merlo, J. Tiedemann, and R. Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 3428–3439. DOI: 10.18653/v1/2021.eacl-main.300. URL: <https://aclanthology.org/2021.eacl-main.300>.
- [48] T. James and H. Hennig. “Knowledge Graphs and Their Applications in Drug Discovery”. In: *Methods in Molecular Biology* 2716 (2024), pp. 203–221. DOI: 10.1007/978-1-0716-3449-3_9.
- [49] P. Rutesic, D. Pfisterer, S. Fischer, and H. Paulheim. “Ontology-Based Models of Chatbots for Populating Knowledge Graphs”. In: (Sept. 2023). DOI: 10.13140/RG.2.2.30341.32488.
- [50] Y. Jiang, S. Bordia, Z. Zhong, C. Dognin, M. Singh, and M. Bansal. *HoVer: A Dataset for Many-Hop Fact Extraction And Claim Verification*. 2020. arXiv: 2011.03088 [cs.CL].
- [51] T. Diggelmann, J. Boyd-Graber, J. Bulian, M. Ciaramita, and M. Leippold. *CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims*. 2021. arXiv: 2012.00614 [cs.CL].
- [52] J. Vladika, P. Schneider, and F. Matthes. *HealthFC: A Dataset of Health Claims for Evidence-Based Medical Fact-Checking*. 2023. arXiv: 2309.08503 [cs.CL].
- [53] I. Mohr, A. Würl, and R. Klinger. *CoVERT: A Corpus of Fact-checked Biomedical COVID-19 Tweets*. 2022. arXiv: 2204.12164 [cs.CL].
- [54] J. Kim, S. Park, Y. Kwon, Y. Jo, J. Thorne, and E. Choi. *FactKG: Fact Verification via Reasoning on Knowledge Graphs*. 2023. arXiv: 2305.06590 [cs.CL].
- [55] *googlesearch-python*. URL: <https://pypi.org/project/googlesearch-python/> (visited on 06/11/2024).
- [56] *HuggingChat*. URL: <https://huggingface.co/chat/> (visited on 06/11/2024).
- [57] *HuggingChat API*. URL: <https://github.com/Soulter/hugging-chat-api> (visited on 06/11/2024).
- [58] Cohere. *CommandR+*. URL: <https://docs.cohere.com/docs/command-r-plus> (visited on 06/11/2024).

- [59] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. “Improving Efficiency and Accuracy in Multilingual Entity Extraction”. In: *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*. 2013.
- [60] C. Boettiger. *rdflib: A high level wrapper around the redland package for common rdf applications*. Zenodo, 2018. DOI: 10.5281/zenodo.1098478. URL: <https://doi.org/10.5281/zenodo.1098478>.
- [61] *Dash Plotly*. URL: <https://dash.plotly.com> (visited on 05/12/2024).
- [62] *Dash Bootstrap Components*. URL: <https://dash-bootstrap-components.opensource.faculty.ai> (visited on 05/12/2024).
- [63] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. *Mixtral of Experts*. 2024. arXiv: 2401.04088 [cs.LG].
- [64] *List of Presidents of the Czech Republic*. URL: https://en.wikipedia.org/wiki/List_of_presidents_of_the_Czech_Republic (visited on 06/05/2024).
- [65] J. Vladika and F. Matthes. *Comparing Knowledge Sources for Open-Domain Scientific Claim Verification*. 2024. arXiv: 2402.02844 [cs.CL].