EviGenerate: Generative Evidence in Automated Fact-Checking

Uku Kangur[1*], **Krish Agrawal**[2*], **Roshni Chakraborty**[3], **Rajesh Sharma**[1,4,5]

[1]Institute of Computer Science, University of Tartu, Tartu, Estonia
[2]Indian Institute of Technology Indore, Indore, India
[3]ABV IIITM Gwalior, Gwalior, Madhya Pradesh, India
[4]Plaksha University, Punjab, India
[5]Kyiv School of Economics, Kyiv, Ukraine
uku.kangur@ut.ee, cse210001034@iiti.ac.in, roshni@iiitm.ac.in, rajesh.sharma@ut.ee

In the era of widespread misinformation, the imperative tasks of fact verification and correction have become essential, especially in the realm of online social media. Traditional manual fact-checking, while crucial, is time-consuming, emphasizing the need for innovative approaches. This research introduces an automated fact-checking system leveraging sophisticated language models for evidence generation to dynamically adapt to the evolving information landscape. The proposed system *EviGenerate* employs a novel evidence generation pipeline, integrating strategies such as named entity hints, question formulation, relation explanation, cross-examination, and a truthful critic. Utilizing a modified FEVER - a widely used automatic fact-checking dataset, the approach achieves a F1 score of 0.912 for claim verification based on DeBERTa. Our best claim correction result based on T5-3B gives a SARI Keep score of 0.721. The contribution of this work lies in its evidence generation approach and prompting strategies, fostering accuracy and adaptability in automated fact-checking systems.

Introduction

In the digital age, the tasks of fact verification and correction have become increasingly critical, particularly with the rapid spread of misinformation on social media platforms (Muhammed T and Mathew 2022). Fact verification assesses claim truthfulness with evidence, and fact correction fixes inaccuracies. These processes maintain information integrity and combat false narratives. The importance of these efforts is heightened by the challenges of manual fact-checking, which can take professional fact-checkers several days or even weeks to complete (Porter and Wood 2021; Kangur, Chakraborty, and Sharma 2024).

To improve fact-checking efficiency, researchers have developed automated methods to label or remove false information before it spreads online. These systems rely on external knowledge bases like Wikipedia for evidence (Thorne et al. 2018; Mayank, Sharma, and Sharma 2022; Nikopensius et al. 2023), but often lag behind rapidly evolving information, especially during critical events like the COVID-19 pandemic and the January 6th US Capitol attack (Ferreira Caceres et al. 2022; Heine 2021; Sharma, Sharma, and Datta 2024). Traditional models struggle to adapt to the dynamic nature of societal changes and new information (Guo, Schlichtkrull, and Vlachos 2022).

The emergence of large language models, such as ChatGPT (Ouyang et al. 2022), GPT-3.5 (Brown et al. 2020) and GPT-4 (OpenAI et al. 2024), represents a major advancement in overcoming the challenge of limited knowledge data. Trained on billions of text sources, these models develop their own knowledge bases, enable them to handle complex reasoning tasks (Hao et al. 2023). Continuously updated with new data, they play a crucial role in the

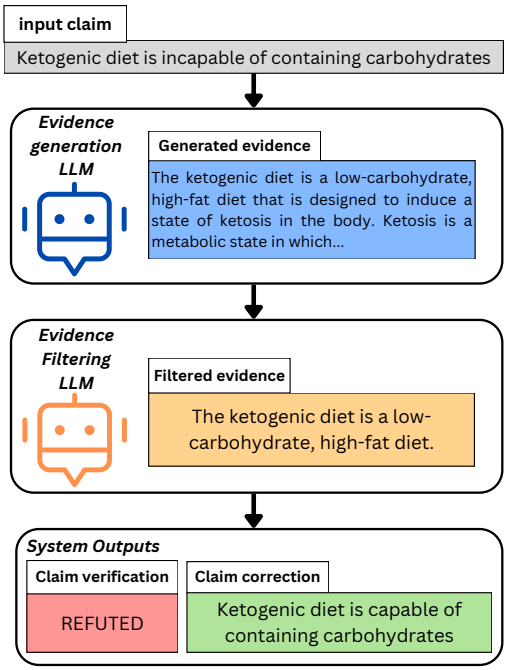

Figure 1: *EviGenerate* uses LLMs to generate the evidence needed for fact verification and correction. The pipeline first generates evidence about the entities in the claim. The generated evidence is then filtered down to only include the information important for fact-checking the claim. We verify and if needed correct the claim using the filtered evidence.

*These authors contributed equally.

fact-checking domain by enhancing accuracy and efficiency (Peng et al. 2023). However, attempts to use these models directly for fact-checking, such as verdict prediction, have often resulted in inaccurate evaluations, as demonstrated by studies like Lee et al. (2020a,b) and Pan et al. (2021), highlights the challenges in generative model applications (Guo, Schlichtkrull, and Vlachos 2022). Some works have also attempted to overcome these challenges by employing Retrieval Augmented Generation (RAG) models, however these models depend strongly on external knowledge bases which can outdate or even worse, publish unchecked inaccuracies (Gao et al. 2024).

To address the challenges in claim evaluation, we propose an evidence generation pipeline, *EviGenerate*, designed to produce accurate evidence for fact-checking claims. Instead of directly asking the model to make a verdict, which risks hallucinations if the model lacks sufficient knowledge, our pipeline splits the task into two subtasks: evidence generation and verdict classification. This approach ensures the model first generates evidence based on what it knows, before making a verdict solely on the generated evidence. By separating reasoning into these steps, we significantly reduce the chance of introducing hallucinated information during the process.

We leverage continuously updated language models like GPT-3.5 (Brown et al. 2020) and GPT-4 (OpenAI et al. 2024) to keep the fact-checking process aligned with evolving information. The pipeline employs strategies such as Named Entity hints, question formulation, relation explanation, cross-examination, and truthful critics (Brown et al. 2020; Cohen et al. 2023). Through human evaluation, we identify three optimal prompts, enhance clarity via filtering methods, and predict verdicts using models like BERT, RoBERTa, DeBERTa, OPT, GPT-3.5, and GPT-4.

For claim correction, we use models like T5, Flan T5, GPT-3.5, and GPT-4 to reconstruct claims with false information masked out (Thorne and Vlachos 2021). The pipeline, detailed in Figure 1, is evaluated on a modified FEVER dataset comprising 10,000 claims labeled as "Supports" or "Refutes".

Our contribution to the field can be succinctly summarized through the following key points:

Novel Evidence Generation Approach: We introduce a novel evidence generation approach for verdict prediction and correction tasks. Our best claim verification model based on DeBERTa provides to get a F1 score of 0.912. The best claim correction model T5-3B ensures a testing SARI Keep score of 0.721.

Prominent Prompting Strategies: We explore existing prompting strategies such as cross-examination and relation explanation, which have not been used previously in this domain. In addition, we introduce diverse and effective prompting strategies for both evidence generation and filtering. Our approach involves crafting prompts that stimulate informative responses and elucidate relationships between entities.

Related Works

In this section, we discuss works from the domain of automated fact-checking, LLMs in automated fact-checking and LLM prompting, on the intersection of which this work lies.

Automated Fact-Checking

The structured framework of automated fact-checking encompasses five key tasks: **claim detection**, **evidence retrieval**, **verdict prediction**, **justification production**, and **fact correction**. In the claim detection task, the focus is on identifying claims that are socially significant, determined by their relevance and importance to the general public (Hassan, Li, and Tremayne 2015). This preliminary step sets the stage for the subsequent processes.

Evidence retrieval deals with collecting relevant information, be it text, tables, or metadata, essential for generating convincing verdict justifications (Ma et al. 2016; Zubiaga et al. 2016; Gorrell et al. 2019). However, the challenge persists: not all available information is trustworthy. While some models have implicitly assumed access to trusted sources like encyclopedias or vetted search engine results, real-world applications demand curated evidence through manual journalism, automated means, or their combination (Thorne et al. 2018; Augenstein et al. 2019; Li et al. 2015).

Verdict prediction emerges as a critical part, where the system endeavors to determine the veracity of identified claims. This can range from binary classifications to nuanced multi-class labels, mirroring the approaches of journalistic fact-checking agencies (Nakashole and Mitchell 2014; Popat et al. 2016; Potthast et al. 2018; Wang 2017; Alhindi, Petridis, and Muresan 2018; Shahi and Nandini 2020). The challenge here lies in steering clear of overly definitive claims, acknowledging the inherent limitations of the models (Graves 2018).

The final stage, justification production, adds a layer of complexity. Justifying decisions is imperative to convince users of the model's interpretation of evidence. Attention weights, logical systems, and summarization techniques come into play to articulate how retrieved evidence was utilized, all underpinning the need for explainability in an era of black-box models (Popat et al. 2018; Shu et al. 2019; Lipton 2018; Atanasova et al. 2020; Gad-Elrab et al. 2019; Ahmadi et al. 2019).

More recently, studies have also delved into fact correction for both claims and abstractive summaries, addressing the challenge of correcting factual errors (Thorne and Vlachos 2021). While prior methods focused on entity-level errors, recent approaches, such as SpanFact (Dong et al. 2020) and token fact correction (Shin, Park, and Song 2023), demonstrate advancements in enhancing factual consistency in system-generated summaries, bridging gaps in real-world applications. In addition new approaches for mitigation of misinformation has also been proposed (Sharma, Datta, and Sharma 2024), which is out of scope of this work.

LLMs in Automated Fact-Checking

The concept of employing large language models (LLMs) for claim correction and verification has seen significant advancements in recent years. Our inspiration originates from

the claim correction pipeline introduced by Thorne and Vlachos (2021), which marks a pivotal shift in this domain. However, the approach, like many others in this field, heavily relied on external knowledge bases for both verification and correction processes. These databases often require frequent manual updates, rendering them impractical for real-time applications.

An emerging trend, as evidenced in the works of Lee et al. (2020b,a), is the utilization of transformer-based language models' knowledge bases for automatic detection and fact-checking of claims. This approach offers considerable benefits in terms of speed, scalability, and versatility. Nonetheless, Guo, Schlichtkrull, and Vlachos (2022) highlights a critical drawback: these large language models, despite their efficiency, often harbor biases that can impede the production of accurate justifications.

To address these concerns, researchers like Barikeri et al. (2021) have explored various debiasing techniques. These include Language Model Debiasing Loss, Attribute Distance Debiasing, Hard Debiasing Loss, and Counterfactual Augmentation. Despite their effectiveness, such methods require extensive computational resources and risk compromising the integrity of the underlying knowledge base.

LLM Prompting

LLM Prompting refers to the process of providing a textual input or "prompt" to a large-scale language model, which then generates a response based on that input. Recent studies have shifted focus towards the use of prompting strategies to increase task performance and mitigate the logical errors inherent in LLMs. Cohen et al. (2023) pioneered an approach where LLMs are pitted against each other to cross-examine information, while Hu et al. (2023), Zeng and Gao (2023) explored the restructuring of claims through prompting before classification. Pan et al. (2023) innovatively employed an ensemble of language models to dissect and analyze sub-claims for a comprehensive evaluation of the original claim.

In contrast to these methods, our work proposes a novel approach that leverages the knowledge bases of language models not for direct fact-checking but for generating reliable evidence. This evidence serves as the foundation for the subsequent verification and correction of claims. Our methodology encompasses smart prompt engineering to extract necessary evidence, eliminate errors and biases, and filter relevant information for effective fact-checking. The subsequent sections will detail the steps involved in our approach.

Data

We use the modified FEVER dataset provided by Thorne & Vlachos (2021). The dataset contains 60000 different claims with the labels 'Supports", Refutes" or Not Enough Information". This modified FEVER dataset also includes corrected claims for the Refutes" labelled claims. These annotations allow us to build both a verification and a correction tool. The dataset also includes references to the evidence needed for the fact-check, but as we generate the evidence ourselves, we disregard the given evidence of the dataset. We disregard claims with the label Not Enough Information" from our dataset. We do this as LLMs have difficulty expressing uncertainy which is crucial for classifying claims that do not have enough information about them (Xiong et al. 2023). Additionally, we can not use claims with Not Enough Information" labels in correction as they do not have a corrected counterpart in the dataset (Thorne et al. 2018).

For computational efficiency, we sample 5,000 "Supports" and 5,000 "Refutes" claims, dividing the 10,000 selected claims into training (80%) and testing (20%) subsets. This yields 4,000 training and 1,000 testing claims per label.

Methodology

We introduce *EviGenerate*, a generated evidence based automatic fact-checking pipeline. The pipeline consists of four different phases:

- Phase I - Evidence generation
- Phase II - Evidence filtering
- Phase III - Claim verification
- Phase IV - Claim correction

The pipeline can be also seen in Figure 1. For all prompts we use the OpenAI API[1] to access both GPT-3.5-turbo-0125 and GPT-4-1106-preview. We choose these two as those were the latest available at the time of running the experiments. We run each prompt in a separate GPT instance to ensure the zero-shot setting of evidence generation.

Phase I - Evidence generation

In total, we integrate five distinct strategies to create a set of 15 prompts, each designed to generate evidence related to the claims. We select the strategies with the aim of extracting as much information as possible about the entities and the relation between the entities in the claim. The more detailed rationale behind choosing these strategies and their synergy in evidence generation is detailed in each subsections. We refer to the first Language Model (LLM) that receives the initial prompt as the 'primary LLM'. If we need to use another LLM to work with the output from the primary LLM, we call this the 'secondary LLM'. Essentially, the secondary LLM is employed when there is an interaction between the two models.

Named Entity and Keyphrase Hints: In order to disambiguate entities within a claim and enhance the model's understanding, we employed Named Entity hints. By explicitly providing Named Entity Recognition (NER) tags and entity names as hints in our prompts, we aimed to guide the Language Model (LLM) in accurately identifying the intended entity within the claim. This strategy was designed to reduce ambiguity arising from words with multiple meanings, enabling the model to generate evidence with a more precise interpretation of the specified entities.

To implement this approach, we used both SpaCy[2] and StanfordNLP[3] for the NER tagging task. In addition to using NER tags, we also explored an alternative strategy based

[1] https://openai.com/blog/openai-api

[2] https://spacy.io/api/entityrecognizer

[3] https://stanfordnlp.github.io/CoreNLP/ner.html

on keyphrase extraction, where the model was provided with only the extracted entity names and phrases, omitting the use of explicit NER tags. For keyphrase extraction, we used KeyBERT[4], which after manual evaluation of 50 examples demonstrated better performance compared to the NER-based approach and was therefore selected as the preferred method for this task.

Question formulation: To stimulate informative responses, we create prompts that involve generation of questions based on NER tokens and keyphrases. The LLM was assigned the dual role of formulation and answering these questions, which aimed to get comprehensive information about specific entities and their interconnections. This approach promotes an interactive and iterative dialogue with the model, encouraging it to explore and provide comprehensive insights, thereby enriching the information retrieval process.

Relation explanation: To elucidate relationships between entities, we develop prompts that instruct the LLM to explain connections between NER tokens and keyphrases. This strategy aim to uncover contextual information and provide insights into the associations between entities. For example, by understanding the relationship between "The Boston Celtics" and "TD Garden" one can determine the relation of "The Boston Celtics is an NBA basketball team which plays its home games at TD Garden arena"

Cross-examination: In a simulated cross-examination scenario, we tasked the examiner (primary LLM) with verifying the correctness of a given claim ("The Boston Celtics play their home games at TD Garden.") as introduced by Cohen et al. (2023). The examiner interacted with the examinee (secondary LLM) by posing short questions, gradually building a case to confirm or refute the claim. This strategy leveraged iterative questioning to assess the accuracy of provided information.

Truthful critic: We use a modified role-playing prompt strategy introduced by Kong et al. (2023). In our version the primary LLM played the role of a truthful critic, whose job is to ask questions from a claimant (secondary LLM) in order to gather truth about the claim. The critic then summarizes the acquired information while not giving judgements to the truthfulness of the generated evidence. While the truthful critic and cross-examination methods seems similar, the aspect of taking a specific role and not giving judgements is what differenciates these two from each other.

Selection of best prompts: We evaluate the quality of the 15 prompts on a subset of 150 claims. We ask three human annotators to rank the prompts and select the top 3 best performing prompts based on mutual agreement. When evaluating the human evaluators focused on **conciseness** (how short the evidence was), **usefulness** (can the claim be proven/disproven given the evidence) and **clarity** (how clear the evidence was). It is crucial to note that these annotators were not associated with the paper writing and were provided with specific instructions to assess the prompts based on their outputs. Based on the agreement among the three annotators, the prompts selected were:

- Generate Question and Answers Using Keyphrases (QA-Prompt)
- Relation Based Prompt Using Keyphrases (Relation-Prompt)
- Cross-Examination (CE-Prompt)

Phase II - Evidence filtering

The generated evidence may contain noise, parts that are irrelevant or unclear, or it might be excessively lengthy. To tackle this issue, we employ different filtering methods to make the generated evidence even more concise, useful and clear. We employ three different strategies:

- Selection of the last paragraph
- Summary request to the primary LLM
- Summary request to a secondary LLM

We employ all of these filtering methods to our previously selected best prompts (exception here being selecting last paragraph of cross-examination).

Selection of the last paragraph: The intuition behind this strategy stems from the observation that LLMs often provide a general explanation or definition at the beginning of their responses and consolidate the final summary at the end. We hypothesized that the last paragraph of the response would contain all the essential information needed to make predictions based on the provided evidence. However, during the evaluation process, our annotators reviewed a sample of responses to validate this assumption. Upon analyzing a representative set of question-answer pairs, it was determined that the last paragraph predominantly contained the final answer to the most recent question, rather than a comprehensive summary of all preceding questions. Consequently, we decided not to use this strategy for prompts following the cross-examination question-answer format, as it did not meet our objective of summarizing the entire interaction.

Summary request to the primary LLM: We notice that the model sometimes gave long essay answers in which the important details were hidden across different paragraphs, thus making it cumbersome to find the important details. Therefore, in this strategy, we additionally ask the same model, i.e., the primary LLM, to give an overview of his answer in a more concise (around 100 word) format.

Summary request to a secondary LLM: Apart from the previous subsection where we additionally ask the primary LLM to summarize, we also tried summarizing using another model, which we refer as secondary LLM. Our reasoning behind using secondary LLM was that a model tend to over-focus on the initial prompt/question even if it is asked to summarize the provided answer. Based on our experiments, we observe that the primary LLM proactively engaged in correcting or classifying the original claim, even

[4]https://github.com/MaartenGr/KeyBERT

though it wasn't explicitly instructed to undertake such actions. By employing a secondary model to perform the summarizing, we aimed to achieve a more balanced representation of the entire text, avoiding the bias towards the initial prompt.

Selection of best filtered prompts: Building on our previously established selection of the best prompts, we introduced three distinct filtering methods to refine the outputs of these prompts further. For each of the top prompts, we identified the most effective filter by applying them to a specific set of 150 claims. Total of three human evaluators were used to select the best filtering method for each of our top 3 prompts based on mutual agreement. As previously mentioned in Section , the annotators focus on **conciseness**, **usefulness** and **clarity** to rank the outputs. The final set of prompts and their best filters are as follows:

- QA-Prompt + Summary request to a secondary LLM
- Relation-Prompt + Summary request to a secondary LLM
- CE-prompt + Summary request to the primary LLM

Phase III - Claim verification

In the verdict prediction phase, our objective is to categorize claims into two distinct classes: "Supports" and "Refutes". To achieve this, we implement a classification model that takes as input the concatenation of the original claim and the generated-filtered evidence, and outputs the class of the claim. This approach allows the model to leverage both the inherent information in the claim and the relevant evidence produced during the evidence generation and filtering stages. We fine-tune several state-of-the-art models, including BERT-base, RoBERTa-base, RoBERTa-large, DeBERTa-v3-base and OPT to assess their performance in capturing the nuanced relationships between claims and evidence. We evaluate the model performance on our beforementioned prepared test set. We create neural networks consisting of the transformer followed by a linear layer, a dropout layer, and finally, a linear layer to obtain the final output. We utilize cross entropy as the loss function and fine-tune the models for 10 epochs, with train batch size 8 and initial learning rate 10^{-5} with Adam optimizer. In addition, we also assess generative models such as GPT-3.5-turbo-0125 and GPT-4-1106-preview for the same task with and without the previously generated evidence as an ablation study.

Phase IV - Claim correction

We employ the T5 claim correction procedure as introduced by Thorne and Vlachos (2021). We select the best performing solution, where the training masks were random and test masks were heuristic (masking tokens which are not in common between the claim and the evidence). We evaluate the model performance on our prepared test set. We fine-tune several models including T5-small, T5-base, T5-large, T5-3B, Flan T5-small, Flan T5-base, Flan T5-large for our purpose. The input to the language models is structured in the following way: "corrector: claim: (masked claim) evidence: (evidence)", for example, if the claim is "Ketogenic diet is incapable of containing carbohydrates" and the generated evidence is "The ketogenic diet is a low-carbohydrate, high-fat diet that...", the prompt to the language models will be "corrector: claim: Ketogenic diet is incapable of containing carbohydrates evidence: The ketogenic diet is a low-carbohydrate, high-fat diet that...". We utilize AdamW optimizer with an initial learning rate $3 * 10^{-5}$. To fine-tune large and xl models, we employ Low-Rank Adaptation of Language Models and Quantization techniques discussed by Hu et al. (2021) with rank 32, LoRA alpha 32, and LoRA dropout 0.05. In addition, we also assess generative models such as GPT-3.5-turbo-0125 and GPT-4-1106-preview for the same task with and without the previously generated evidence as an ablation study.

Results

In this Section, we present comprehensive results of our experimental evaluations on Claim Verification and Claim Correction phases. Our analysis encompasses a range of models, including DeBERTa-v3-base, RoBERTa-base, RoBERTa-large, OPT, BERT-base, GPT-3.5-turbo-0125 and GPT-4-1106-preview for Claim Verification. For Claim Correction we employ GPT-3.5-turbo-0125, GPT-4-1106-preview and T5 variants in small, base, large, and XL sizes. We compare the performance for Claim Verification on the basis of F1 score, precision, recall, and accuracy and Claim Correction on the basis of ROUGE (Lin 2004), BLEU (Papineni et al. 2002), and SARI (Xu et al. 2016), respectively. The following subsections detail the performance of each model in both tasks.

Claim Verfication

In this Subsection, we present the comparative results of the verification models along with their performance with respect to different prompting techniques in detail.

Table 1 presents the evaluation results for various models on the testing data across three different prompts and their corresponding optimal filters. The model checkpoint selected for evaluation is determined by the minimum loss value observed during fine-tuning.

For the QA-Prompt + Requesting a summary from a secondary LLM method, DeBERTa-v3-base with GPT-4-1106-preview evidence exhibits the highest accuracy 88.90% and F1 score 0.890 among the models. DeBERTa-v3-base with GPT-3.5-turbo-0125 evidence exhibits comparable performance, an accuracy of 87.65% and F1 score of 0.878, while OPT with GPT-3.5-turbo-0125 evidence performs worst amongst the evaluated models with an accuracy of 79.99% and F1 score of 0.810. Notably, while GPT-4-1106-preview showcases superior precision (0.911), DeBERTa-v3-base gives best recall (0.899), emphasizing its robust performance across various metrics.

In the case of the Relation-Prompt + Requesting a summary from a secondary LLM method, DeBERTa-v3-base with GPT-4-1106-preview evidence again stands out with an accuracy of 89.00% and an F1 score of 0.892. The second best-evaluated model is RoBERTa-large with GPT-4-1106-preview evidence resulting in an accuracy of 88.70% and

Evidence Model	Method	Model	Accuracy	F1 Score	Precision	Recall
GPT-3.5-turbo-0125	QA-Prompt + Summary request from a secondary LLM	DeBERTa-v3-base	87.65	0.878	0.868	0.888
		RoBERTa-base	83.55	0.840	0.819	0.862
		BERT-base	82.80	0.828	0.829	0.826
		RoBERTa-large	86.05	0.863	0.848	0.878
		OPT	79.65	0.795	0.801	0.789
		GPT-3.5-Turbo-0125	79.99	0.810	0.829	0.784
	Relation-Prompt + Summary request from a secondary LLM	DeBERTa-v3-base	86.00	0.867	0.826	0.912
		RoBERTa-base	83.55	0.841	0.803	0.883
		BERT-base	81.55	0.818	0.806	0.831
		RoBERTa-large	85.70	0.863	0.828	0.902
		OPT	81.35	0.818	0.798	0.840
		GPT-3.5-turbo-0125	78.33	0.810	0.721	0.925
	CE-prompt + Summary request from the primary LLM	DeBERTa-v3-base	89.10	0.890	0.895	0.886
		RoBERTa-base	86.40	0.864	0.864	0.864
		BERT-base	85.25	0.849	0.867	0.833
		RoBERTa-large	88.25	0.882	0.884	0.880
		OPT	84.05	0.838	0.845	0.827
		GPT-3.5-turbo-0125	81.01	0.814	0.799	0.829
GPT-4-1106-preview	QA-Prompt + Summary request from a secondary LLM	DeBERTa-v3-base	88.90	0.890	0.881	0.899
		RoBERTa-base	86.05	0.863	0.848	0.879
		BERT-base	82.70	0.823	0.841	0.807
		RoBERTa-large	86.85	0.872	0.852	0.892
		OPT	81.80	0.830	0.779	0.888
		GPT-4-1106-preview	85.66	0.846	0.911	0.790
	Relation-Prompt + Summary request from a secondary LLM	DeBERTa-v3-base	89.00	0.892	0.874	0.911
		RoBERTa-base	86.25	0.864	0.852	0.877
		BERT-base	82.90	0.835	0.806	0.866
		RoBERTa-large	88.70	0.890	0.867	0.914
		OPT	82.45	0.821	0.838	0.805
		GPT-4-1106-preview	86.80	0.861	0.912	0.815
	CE-prompt + Summary request from the primary LLM	DeBERTa-v3-base	**91.00**	**0.912**	0.891	**0.934**
		RoBERTa-base	87.60	0.878	0.864	0.892
		BERT-base	85.20	0.854	0.844	0.864
		RoBERTa-large	90.00	0.900	0.899	0.901
		OPT	84.15	0.841	0.845	0.836
		GPT-4-1106-preview	90.03	0.897	**0.925**	0.872
No Evidence		GPT-3.5-turbo-0125	79.87	0.800	0.794	0.807
		GPT-4-1106-preview	87.41	0.869	0.905	0.836

Table 1: Results for the various models used in the Claim Verification Phase.

an F1 score of 0.890. Again the worst-performing model is OPT with GPT-3.5-turbo-0125 evidence with an accuracy of 79.65% and F1 score of 0.795.

The CE-prompt + Requesting a summary from the primary LLM method demonstrates DeBERTa-v3-base with GPT-4-1106-preview evidence as the top performer, achieving the highest accuracy 91.00% and F1 score 0.912. It also maintains a high recall 0.934, highlighting its overall effectiveness. Suprisingly, the second best model GPT-4-1106-preview with GPT-4-1106-preview evidence gives the best precision 0.925. BERT-base and OPT exhibit lower performance across all metrics, around $5-7\%$ lower compared to the best performing model in terms of accuracy.

Therefore, we observe that DeBERTa-v3-base performs consistently better by $1-8\%$ in F1 score than BERT-base, RoBERTa-base, RoBERTa-large, OPT, GPT-3.5-turbo-0125 or GPT-4-1106-preview for verification irrespective of different prompts, summarization strategies or evidence generation models. Additionally, we observe that CE-prompt consistently outperforms the Relation-Prompt and QA-Prompt across F1 score by around 2% and 3% respectively. Our results also highlight the efficiency (2-3% higher accuracy) of evidence generation methods compared to running GPT-3.5-turbo-0125 or GPT-4-1106-preview without the evidence.

Claim Correction

Here we present a detailed analysis of the results of claim correction tasks across GPT-3.5-turbo-0125, GPT-4-1106-preview and T5 model variants, including small, base, large, and XL sizes. In the automated assessment of the Claim Correction phase, we employ SARI, i.e., a metric specifically designed for evaluating sentence simplification. SARI com-

Method Name	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	SARI Keep	SARI Delete	SARI Add	SARI Final
QA-Prompt + Summary request from a secondary LLM	T5-small	0.739	0.583	0.733	0.394	0.562	0.472	0.115	0.383
	Flan T5-small	0.743	0.585	0.737	0.400	0.560	0.468	0.115	0.381
	T5-base	0.768	0.627	0.764	0.452	0.594	0.485	0.125	0.401
	Flan T5-base	0.774	0.633	0.769	0.452	0.591	0.488	0.133	0.404
	T5-large	0.707	0.550	0.697	0.364	0.534	0.468	0.114	0.372
	Flan T5-large	0.765	0.621	0.759	0.446	0.587	0.497	0.140	0.408
	T5-3B	0.775	0.640	0.770	0.465	0.606	0.506	0.151	0.421
	GPT-3.5-turbo-0125	0.697	0.553	0.690	0.332	0.635	0.552	0.078	0.422
Relation-Prompt + Summary request from a secondary LLM	T5-small	0.787	0.654	0.782	0.489	0.610	0.431	0.123	0.388
	Flan T5-small	0.787	0.654	0.783	0.488	0.608	0.439	0.127	0.392
	T5-base	0.810	0.691	0.807	0.537	0.640	0.463	0.140	0.414
	Flan T5-base	0.811	0.692	0.808	0.537	0.641	0.472	0.146	0.420
	T5-large	0.808	0.687	0.803	0.530	0.639	0.469	0.150	0.419
	Flan T5-large	0.801	0.678	0.796	0.521	0.632	0.472	0.154	0.419
	T5-3B	0.814	0.699	0.809	0.549	0.645	0.473	0.158	0.426
	GPT-3.5-turbo-0125	0.715	0.567	0.708	0.343	0.631	0.533	0.076	0.413
CE-prompt + Summary request from the primary LLM	T5-small	0.836	0.738	0.832	0.589	0.687	0.459	0.151	0.432
	Flan T5-small	0.840	0.746	0.837	0.604	0.694	0.479	0.160	0.444
	T5-base	0.857	0.773	0.854	0.640	0.711	0.493	0.188	0.464
	Flan T5-base	0.863	0.779	**0.862**	0.646	0.711	0.502	0.196	0.470
	T5-large	0.858	0.773	0.854	0.638	0.713	0.517	0.197	0.476
	Flan T5-large	0.854	0.765	0.850	0.627	0.710	0.528	0.198	0.479
	T5-3B	**0.865**	**0.782**	**0.862**	**0.650**	**0.721**	0.521	**0.202**	**0.482**
	GPT-3.5-turbo-0125	0.720	0.571	0.714	0.346	0.640	0.554	0.080	0.425
No Evidence	GPT-3.5-turbo-0125	0.778	0.618	0.773	0.384	0.660	**0.575**	0.0845	0.440

Table 2: Results for the various models used in the Claim Correction Phase using evidence generated using GPT-3.5-turbo-0125. The last row presents the ablation results of using GPT-3.5-turbo-0125 directly without evidence.

prises of four key components:

SARI Keep: This component measures the F1 score of n-grams that are retained from the source sentence in the output. It evaluates how well the simplified sentence maintains important content from the original.

SARI Delete: SARI Delete assesses the n-grams that are present in the source sentence but are deleted in the output. It helps capture the information loss or deletion of unnecessary details during the simplification process.

SARI Add: This component focuses on the n-grams that are added to the output sentence but were not present in the source. SARI Add evaluates the introduction of new information or details that were not in the original sentence.

SARI Final: SARI Final is a composite metric that combines the scores from SARI Keep, SARI Delete, and SARI Add. It provides an overall evaluation of the sentence simplification output, taking into account both retention and modification of n-grams.

Additionally, we include BLEU and ROUGE in our reporting to signify the precision and recall of the correction process. As Thorne and Vlachos (2021) mention SARI Keep scores have the highest correlation to manual evaluation, we compare our models primarily on SARI Keep scores. Tables 2 and 3 present these metrics for the heuristic masking testing dataset for GPT-3.5-turbo-0125 and GPT-4-1106-preview respectively.

For the QA-Prompt + Requesting a summary from a secondary LLM method, the newer models, such as GPT-3.5-turbo-0125 and GPT-4-1106-preview demonstrate higher performance across most metrics, about a 30% increment in SARI Final scores (0.476 compared to 0.359) compared to smaller T5 variants. Notably, Flan T5-large outperforms its counterpart T5-large, showcasing improved scores in ROUGE-1 ($\approx$ 7%, 0.765 compared to 0.707), ROUGE-2 (0.621 compared to 0.550), and ROUGE-L (0.759 compared to 0.697). The SARI metrics for Keep (0.674), Delete (0.612), and Add (0.141) operations indicate best performance with GPT-4-1106-preview, underlining the effectiveness of GPT models for this prompt.

Relation-Prompt + Requesting a summary from a secondary LLM method, similar trends emerge. Larger and newer models consistently outperform smaller and older ones, with GPT-4-1106-preview exhibiting the highest scores, SARI Keep score of 0.682. The Flan-T5 variants again outperform their counterpart T5 models. The SARI metrics further reinforce this observation, indicating improvements in content retention (Keep), deletion accuracy (Delete), and new information integration (Add) for the generative models, highlighting an improvement of up to 10% in SARI scores.

For the CE-prompt + Requesting a summary from the primary LLM method T5-3B, demonstrates superior performance and SARI Keep score of 0.721 compared to 0.640 and 0.706 for GPT-3.5-turbo-0125 and GPT-4-1106-preview, respectively. The SARI metrics are however vary on the best model as the best SARI Delete score is given by GPT-4-1106-preview (0.655 compared to 0.459), while best SARI Add score (0.202 compared to 0.080) is given by T5-3B.

Therefore, we observe that T5-3B with GPT-3.5-turbo-0125 evidence performs better in SARI Keep score than T5-small, T5-base, T5-large, Flan T5-small, Flan T5-base, Flan T5-large, GPT-3.5-turbo-0125 and GPT-4-1106-preview. Additionally, we observe CE-prompt + Requesting a summary from the primary LLM generally outperforms the other methods across all model sizes and metrics. This method

Method Name	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	SARI Keep	SARI Delete	SARI Add	SARI Final
QA-Prompt + Summary request from a secondary LLM	T5-small	0.690	0.531	0.687	0.345	0.518	0.471	0.089	0.359
	Flan T5-small	0.697	0.538	0.693	0.352	0.520	0.463	0.091	0.358
	T5-base	0.722	0.578	0.719	0.401	0.551	0.487	0.114	0.384
	Flan T5-base	0.728	0.583	0.725	0.404	0.542	0.477	0.111	0.377
	T5-large	0.718	0.572	0.713	0.397	0.548	0.490	0.116	0.385
	Flan T5-large	0.720	0.574	0.715	0.398	0.552	0.500	0.116	0.390
	T5-3B	0.728	0.591	0.724	0.422	0.562	0.499	0.129	0.400
	GPT-4-1106-preview	0.770	0.647	0.766	0.432	0.674	0.612	0.141	0.476
Relation-Prompt + Summary request from a secondary LLM	T5-small	0.715	0.564	0.710	0.393	0.558	0.304	0.091	0.317
	Flan T5-small	0.721	0.568	0.716	0.400	0.533	0.460	0.114	0.369
	T5-base	0.742	0.610	0.738	0.455	0.568	0.476	0.132	0.392
	Flan T5-base	0.744	0.607	0.740	0.453	0.600	0.348	0.122	0.356
	T5-large	0.739	0.602	0.735	0.444	0.563	0.496	0.141	0.400
	Flan T5-large	0.738	0.599	0.734	0.440	0.563	0.491	0.134	0.396
	T5-3B	0.746	0.614	0.742	0.459	0.578	0.500	0.144	0.407
	GPT-4-1106-preview	0.784	0.665	0.780	0.454	0.682	0.600	0.148	0.477
CE-prompt + Summary request from the primary LLM	T5-small	0.794	0.670	0.791	0.491	0.646	0.497	0.131	0.425
	Flan T5-small	0.798	0.673	0.793	0.501	0.645	0.490	0.143	0.426
	T5-base	0.820	0.712	0.818	0.553	0.676	0.528	0.173	0.459
	Flan T5-base	0.825	0.714	0.822	0.552	0.672	0.528	0.172	0.458
	T5-large	0.818	0.704	0.812	0.539	0.670	0.535	0.178	0.461
	Flan T5-large	0.816	0.701	0.811	0.535	0.673	0.546	0.174	0.465
	T5-3B	**0.829**	**0.722**	**0.824**	**0.560**	0.681	0.537	0.183	0.467
	GPT-4-1106-preview	0.799	0.682	0.796	0.478	0.706	0.631	0.166	0.501
No Evidence	GPT-4-1106-preview	0.802	0.685	0.799	0.477	**0.707**	**0.655**	**0.187**	**0.516**

Table 3: Results for the various models used in the Claim Correction Phase using evidence generated using GPT-4-1106-preview. The last row presents the ablation results of using GPT-4-1106-preview directly without evidence

achieves higher scores in terms of ROUGE ($\approx 5-15\%$), BLEU ($\approx 15-30\%$), and SARI ($\approx 10-15\%$), indicating its efficacy in generating more accurate and semantically similar corrections. Suprisingly, GPT-4-1106-preview tend to get higher SARI Delete scores than the other models, indicating that it is prone to not use words from the original claim when correcting. We additionally highlight that for GPT-4-1106-preview running the correction without evidence yielded the best SARI Keep scores, which can indicate that evidence is not as important for correcting sentences as it is for verifing them.

Summary of Results

In our study, we have demonstrated the competitive capabilities of *EviGenerate* in both claim verification and claim correction tasks, as evidenced by the performance results presented in Tables 1, 2 and 3. For both tasks, the highest scores were given using the CE-prompt + Summary request from the primary LLM. This highlights the efficiency of breaking large challenges into smaller tasks to improve overall efficiency of LLM problem solving.

Claim Verification: We have successfully established *EviGenerate* as a working solution. Our best model achieved an accuracy of 91.00% and F1 score of 0.912, compared to our best results without evidence generation with accuracy of 87.41 and 0.869.

Claim Correction: Our assessment using the SARI metric yielded insightful results:
SARI Keep: We achieved the highest score with T5-3B with GPT-3.5-turbo-0125 generated evidence, scoring 0.721, which shows our model's effectiveness in retaining essential content during correction.
SARI Delete: Suprisingly GPT-4-1106-preview without evidence gave the best score of 0.521. This indicates the model's cautious approach in deletion, prioritizing context preservation when given additional evidence.
SARI Add: With the best score of 0.202, T5-3B with GPT-3.5-turbo-0125 generated evidence model demonstrated a balanced addition of information, crucial for maintaining the accuracy and relevance of corrections.
SARI Final: The best overall SARI Final score stood at 0.516 using GPT-4-1106-preview without evidence. This score reflects the model's comprehensive and balanced capabilities in text modification.

As SARI Keep is the closest to human evaluation for claim correction tasks, we highlight that we consider T5-3B with GPT-3.5-turbo-0125 generated evidence the best for this task. However its important to note that there were marginal differences in results when it came to using or not using evidence generation. This highlights the difficulty of the claim correction task and shows that generative evidence does not provide a large increase in performance in this challenge.

Conclusions

This study introduces *EviGenerate*, an automated fact-checking system that integrates advanced language models for improved evidence generation. The system's notable features include a novel evidence generation pipeline and sophisticated prompting strategies, which facilitate the accurate verification and correction of information across varied domains. *EviGenerate*'s performance, as indicated by its F1 score of 0.912 in claim verification using DeBERTa and a SARI Keep score of 0.721 in claim correction with T5-3B, reflects its capability to handle complex online information.

Acknowledgments

This work has been funded from the EU H2020 program under the SoBigData++ project (grant agreement No. 871042), ETAg (grant No. SLTAT21096), HAMISON project (PCI2022-135026-2), and PSG grant (PSG855).

Appendix

The appendix is structured into two sections: (A) Limitations and (B) Ethics Statement.

Limitations

In this section we highlight the various remaining challenges in the domain of using large language models for generating evidence.

Justification production

One important part of automated fact-checking is justification production, which allows users to evaluate the machine learning output reasoning. We note that with black-box language models such as the ones used in our work, the task of justification production is difficult if not impossible. This is due to the fact that language models interpret language differently than humans and analysing biases and weights is thus not reasonable. Additionally continously trained models, such as GPT-3.5 or GPT-4 are not open-source, thus any analysis of the weights is also not possible.

Limited knowledge base

Our proposed solution relies on the knowledge bases of language models to generate relevant evidence. Large language models are trained on billions of texts and for some models (like GPT-3.5 and GPT-4) the training is continuous based on user input and feedback (Ouyang et al. 2022). However, even then the training data is only a small fraction of the complete knowledge contained over the internet. Future works could improve the system by including external knowledge into the prompts themselves. This could allow the model to better understand the underlying context of the claim and fill its gaps in knowledge.

LLM uncertainty

We only run our full pipeline on the binary classes of "Refutes" and "Supports", disregarding the class of "Not Enough Information" in the process. We decided to take this route due to the way language models work. As generative language models are trained to give the most probable answer, they tend to be overconfident in their answers (Xiong et al. 2023). We noticed that this aspect hindered their ability to create evidence where the model would confess to not knowing the answer, thus making classifying "Not Enough Information" claims impossible with our solution. Future works could solve this by exploring LLM uncertainty quantification methods to evaluate the output generated before it is passed on to the classification stage.

Ethics statement

Our work uses only the FEVER dataset (released under Apache-2.0 license), which consists of collected and modified claims from Wikipedia (Thorne and Vlachos 2021). These claims have been human annotated and carefully curated to not include any personal information or offensive content not suitable for public consumption. The evaluation of our prompts were done by anonymous volunteer evaluators. Any personal data of the evaluators was anonymized and not stored after the end of the project. The evaluators were not revealed to any harmful content during evaluation. We ensure full access to our code repository to ensure reproducibility. Our solution can be misleading in some cases due to implicit LLM bias and hallucinations. We explain all limitations (see Section) in order to mitigate any misuse of our results. Given the nature of our work and the aforementioned reasons, we did not require any permissions from our institution's Ethical Board Committee.

References

Ahmadi, N.; Lee, J.; Papotti, P.; and Saeed, M. 2019. Explainable Fact Checking with Probabilistic Answer Set Programming. arXiv:1906.09198.

Alhindi, T.; Petridis, S.; and Muresan, S. 2018. Where is Your Evidence: Improving Fact-checking by Justification Modeling. In Thorne, J.; Vlachos, A.; Cocarascu, O.; Christodoulopoulos, C.; and Mittal, A., eds., *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 85–90. Brussels, Belgium: Association for Computational Linguistics.

Atanasova, P.; Simonsen, J. G.; Lioma, C.; and Augenstein, I. 2020. A Diagnostic Study of Explainability Techniques for Text Classification. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3256–3274. Online: Association for Computational Linguistics.

Augenstein, I.; Lioma, C.; Wang, D.; Chaves Lima, L.; Hansen, C.; Hansen, C.; and Simonsen, J. G. 2019. MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4685–4697. Hong Kong, China: Association for Computational Linguistics.

Barikeri, S.; Lauscher, A.; Vulić, I.; and Glavaš, G. 2021. RedditBias: A Real-World Resource for Bias Evaluation and Debiasing of Conversational Language Models. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1941–1955. Online: Association for Computational Linguistics.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan,

T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165.

Cohen, R.; Hamri, M.; Geva, M.; and Globerson, A. 2023. LM vs LM: Detecting Factual Errors via Cross Examination. arXiv:2305.13281.

Dong, Y.; Wang, S.; Gan, Z.; Cheng, Y.; Cheung, J. C. K.; and Liu, J. 2020. Multi-Fact Correction in Abstractive Text Summarization. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9320–9331. Online: Association for Computational Linguistics.

Ferreira Caceres, M. M.; Sosa, J. P.; Lawrence, J. A.; Sestacovschi, C.; Tidd-Johnson, A.; Rasool, M. H. U.; Gadamidi, V. K.; Ozair, S.; Pandav, K.; Cuevas-Lou, C.; Parrish, M.; Rodriguez, I.; and Fernandez, J. P. 2022. The impact of misinformation on the COVID-19 pandemic. *AIMS public health*, 9(2): 262–277.

Gad-Elrab, M. H.; Stepanova, D.; Urbani, J.; and Weikum, G. 2019. ExFaKT: A Framework for Explaining Facts over Knowledge Graphs and Text. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 87–95. New York, NY, USA: Association for Computing Machinery. ISBN 9781450359405.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; and Wang, H. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997.

Gorrell, G.; Kochkina, E.; Liakata, M.; Aker, A.; Zubiaga, A.; Bontcheva, K.; and Derczynski, L. 2019. SemEval-2019 Task 7: RumourEval, Determining Rumour Veracity and Support for Rumours. In May, J.; Shutova, E.; Herbelot, A.; Zhu, X.; Apidianaki, M.; and Mohammad, S. M., eds., *Proceedings of the 13th International Workshop on Semantic Evaluation*, 845–854. Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Graves, L. 2018. Understanding the Promise and Limits of Automated Fact-Checking.

Guo, Z.; Schlichtkrull, M.; and Vlachos, A. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10: 178–206.

Hao, S.; Gu, Y.; Ma, H.; Hong, J. J.; Wang, Z.; Wang, D. Z.; and Hu, Z. 2023. Reasoning with Language Model is Planning with World Model. arXiv:2305.14992.

Hassan, N.; Li, C.; and Tremayne, M. 2015. Detecting Check-Worthy Factual Claims in Presidential Debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, 1835–1838. New York, NY, USA: Association for Computing Machinery. ISBN 9781450337946.

Heine, J. 2021. The Attack on the US Capitol: An American Kristallnacht. *Protest*, 1(1): 126 – 141.

Hu, B.; Sheng, Q.; Cao, J.; Shi, Y.; Li, Y.; Wang, D.; and Qi, P. 2023. Bad Actor, Good Advisor: Exploring the Role of Large Language Models in Fake News Detection. arXiv:2309.12247.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Kangur, U.; Chakraborty, R.; and Sharma, R. 2024. Who Checks the Checkers? Exploring Source Credibility in Twitter's Community Notes. arXiv:2406.12444.

Kong, A.; Zhao, S.; Chen, H.; Li, Q.; Qin, Y.; Sun, R.; and Zhou, X. 2023. Better Zero-Shot Reasoning with Role-Play Prompting. arXiv:2308.07702.

Lee, N.; Bang, Y.; Madotto, A.; and Fung, P. 2020a. Misinformation Has High Perplexity. arXiv:2006.04666.

Lee, N.; Li, B. Z.; Wang, S.; Yih, W.-t.; Ma, H.; and Khabsa, M. 2020b. Language Models as Fact Checkers? In Christodoulopoulos, C.; Thorne, J.; Vlachos, A.; Cocarascu, O.; and Mittal, A., eds., *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, 36–41. Online: Association for Computational Linguistics.

Li, Y.; Gao, J.; Meng, C.; Li, Q.; Su, L.; Zhao, B.; Fan, W.; and Han, J. 2015. A Survey on Truth Discovery. arXiv:1505.02463.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Lipton, Z. C. 2018. The Mythos of Model Interpretability. *Commun. ACM*, 61(10): 36–43.

Ma, J.; Gao, W.; Mitra, P.; Kwon, S.; Jansen, B. J.; Wong, K.-F.; and Cha, M. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 3818–3824. AAAI Press. ISBN 9781577357704.

Mayank, M.; Sharma, S.; and Sharma, R. 2022. DEAP-FAKED: Knowledge Graph based Approach for Fake News Detection. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 47–51.

Muhammed T, S.; and Mathew, S. K. 2022. The disaster of misinformation: a review of research in social media. *International Journal of Data Science and Analytics*, 13(4): 271–285.

Nakashole, N.; and Mitchell, T. M. 2014. Language-Aware Truth Assessment of Fact Candidates. In Toutanova, K.; and Wu, H., eds., *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1009–1019. Baltimore, Maryland: Association for Computational Linguistics.

Nikopensius, G.; Mayank, M.; Phukan, O. C.; and Sharma, R. 2023. Reinforcement Learning-based Knowledge Graph Reasoning for Explainable Fact-checking. arXiv:2310.07613.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.;

Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Łukasz Kaiser; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Łukasz Kondraciuk; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.; Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155.

Pan, L.; Chen, W.; Xiong, W.; Kan, M.-Y.; and Wang, W. Y. 2021. Zero-shot Fact Verification by Claim Generation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 476–483. Online: Association for Computational Linguistics.

Pan, L.; Wu, X.; Lu, X.; Luu, A. T.; Wang, W. Y.; Kan, M.-Y.; and Nakov, P. 2023. Fact-Checking Complex Claims with Program-Guided Reasoning. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6981–7004. Toronto, Canada: Association for Computational Linguistics.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Peng, B.; Galley, M.; He, P.; Cheng, H.; Xie, Y.; Hu, Y.; Huang, Q.; Liden, L.; Yu, Z.; Chen, W.; and Gao, J. 2023. Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback. arXiv:2302.12813.

Popat, K.; Mukherjee, S.; Strötgen, J.; and Weikum, G. 2016. Credibility Assessment of Textual Claims on the Web. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 2173–2178. New York, NY, USA: Association for Computing Machinery. ISBN 9781450340731.

Popat, K.; Mukherjee, S.; Yates, A.; and Weikum, G. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 22–32. Brussels, Belgium: Association for Computational Linguistics.

Porter, E.; and Wood, T. J. 2021. The global effectiveness of fact-checking: Evidence from simultaneous experiments in Argentina, Nigeria, South Africa, and the United Kingdom. *Proceedings of the National Academy of Sciences of the United States of America*, 118(37): e2104235118.

Potthast, M.; Kiesel, J.; Reinartz, K.; Bevendorff, J.; and Stein, B. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 231–240. Melbourne, Australia: Association for Computational Linguistics.

Shahi, G. K.; and Nandini, D. 2020. *FakeCovid- A Multilingual Cross-domain Fact Check News Dataset for COVID-19*. ICWSM.

Sharma, S.; Datta, A.; and Sharma, R. 2024. AMIR: An Automated Misinformation Rebuttal System—A COVID-19 Vaccination Datasets-Based Exposition. *IEEE Transactions on Computational Social Systems*, 11(6): 7723–7733.

Sharma, S.; Sharma, R.; and Datta, A. 2024. (Mis)leading the COVID-19 Vaccination Discourse on Twitter: An Exploratory Study of Infodemic Around the Pandemic. *IEEE Transactions on Computational Social Systems*, 11(1): 352–362.

Shin, J.; Park, S.-B.; and Song, H.-J. 2023. Token-Level Fact Correction in Abstractive Summarization. *IEEE Access*, 11: 1934–1943.

Shu, K.; Cui, L.; Wang, S.; Lee, D.; and Liu, H. 2019. DEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 395–405. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.

Thorne, J.; and Vlachos, A. 2021. Evidence-based Factual Error Correction. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3298–3309. Online: Association for Computational Linguistics.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819. New Orleans, Louisiana: Association for Computational Linguistics.

Wang, W. Y. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 422–426. Vancouver, Canada: Association for Computational Linguistics.

Xiong, M.; Hu, Z.; Lu, X.; Li, Y.; Fu, J.; He, J.; and Hooi, B. 2023. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. arXiv:2306.13063.

Xu, W.; Napoles, C.; Pavlick, E.; Chen, Q.; and Callison-Burch, C. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4: 401–415.

Zeng, F.; and Gao, W. 2023. Prompt to be Consistent is Better than Self-Consistent? Few-Shot and Zero-Shot Fact Verification with Pre-trained Language Models. arXiv:2306.02569.

Zubiaga, A.; Liakata, M.; Procter, R.; Wong Sak Hoi, G.; and Tolmie, P. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PLOS ONE*, 11(3): e0150989.