GProofT: A Multi-dimension Multi-round Fact Checking Framework Based on Claim Fact Extraction

Jiayu Liu*, **Junhao Tang***, **Hanwen Wang***,
Baixuan Xu, Haochen Shi, Weiqi Wang, Yangqiu Song
Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
{jliufv, jtangay, hwangfs}@connect.ust.hk

Abstract

In the information era, the vast proliferation of online content poses significant challenges, particularly concerning the trustworthiness of these digital statements, which can have profound societal implications. Although it is possible to manually annotate and verify the authenticity of such content, the sheer volume and rapid pace of information generation render this approach impractical, both in terms of time and cost. Therefore, it is imperative to develop automated systems capable of validating online claims, ensuring that users can use the wealth of information available on the Internet effectively and reliably. Using primarily ChatGPT and the Google search API, GProofT fact checking framework generates question-answer pairs to systematically extract and verify the facts within claims. Based on the outcomes of these QA pairs, claims are subsequently labeled as *Supported*, *Conflicted Evidence/Cherry-Picking*, or *Refuted*. Shown by extensive experiments, GProofT Retrieval generally performs effectively in fact-checking and makes a substantial contribution to the task.

1 Introduction

With the chaotic nature of information on the Internet, it appears to be challenging to determine the credibility of claims on the web. This poses difficulties on LLMs such as ChatGPT (OpenAI, 2023) to conduct fact checking as the hallucination (Huang et al., 2023; Ji et al., 2022) of them can produce seemingly feasible but fake information. Though time-consuming and tedious when performed manually, fact-checking is rather crucial to ensure the trustworthiness of information, especially for the fact-sensitive industry such as journalism and science. In the explosion of information, it's far from adequate to solely rely on manual check to eliminate the rumor and misinformation. Therefore, it's pivotal to develop a trustworthy automatic process to complete fact-checking efficiently and accurately. Recent advancements in LLMs have showcased remarkable performance in tasks involving text comprehension and generation (OpenAI et al., 2024; Wang et al., 2023a; He et al., 2023). However, the application of LLMs in automatic fact-checking has remained a persistent challenge, undergoing continuous exploration and development (Hang et al., 2024; Kim et al., 2024). Current LLMs can only memorize the knowledge embedded in their pretrain data, which makes them struggle with fact-checking when the event is out of their pretrain corpus, namely, out of their knowledge domain. Under this circumstance, it is necessary and crucial to incorporate real-time online search engine to provide LLMs with real-time facts information to assist its reasoning. However, the chaotic nature of internet could imply that the knowledge provided from the search engine could result in misinformation to the LLMs, hindering its reasoning process. Hence, a consistent framework for multi-dimension, multi-round fact checking needs to be proposed to generate stable and trustworthy fact checking result.

To solve the limitation, we propose GProofT fact checking framework to crawl and analyze web evidence based on Google Search API and ChatGPT. As demonstrated in Figure 1, For each given textual claim, we incorporate three stages to retrieve the pertinent evidence from the Internet and a final step to attribute a label based on the retrieved evidence. As suggested in the shared task, our retrieved evidence is in the format of QA pair. The retrieval procedure includes Claim Split, Question Generation, Answer Generation and Expansion. More information could be found in Section 3.1. Overall, our framework could be decomposed into 3 stages:

1. Claim Split: It focuses on the decomposition of the claim for the following question generation.

2. Question Generation: Based on the resulted

*First three authors make equal contribution to this paper.

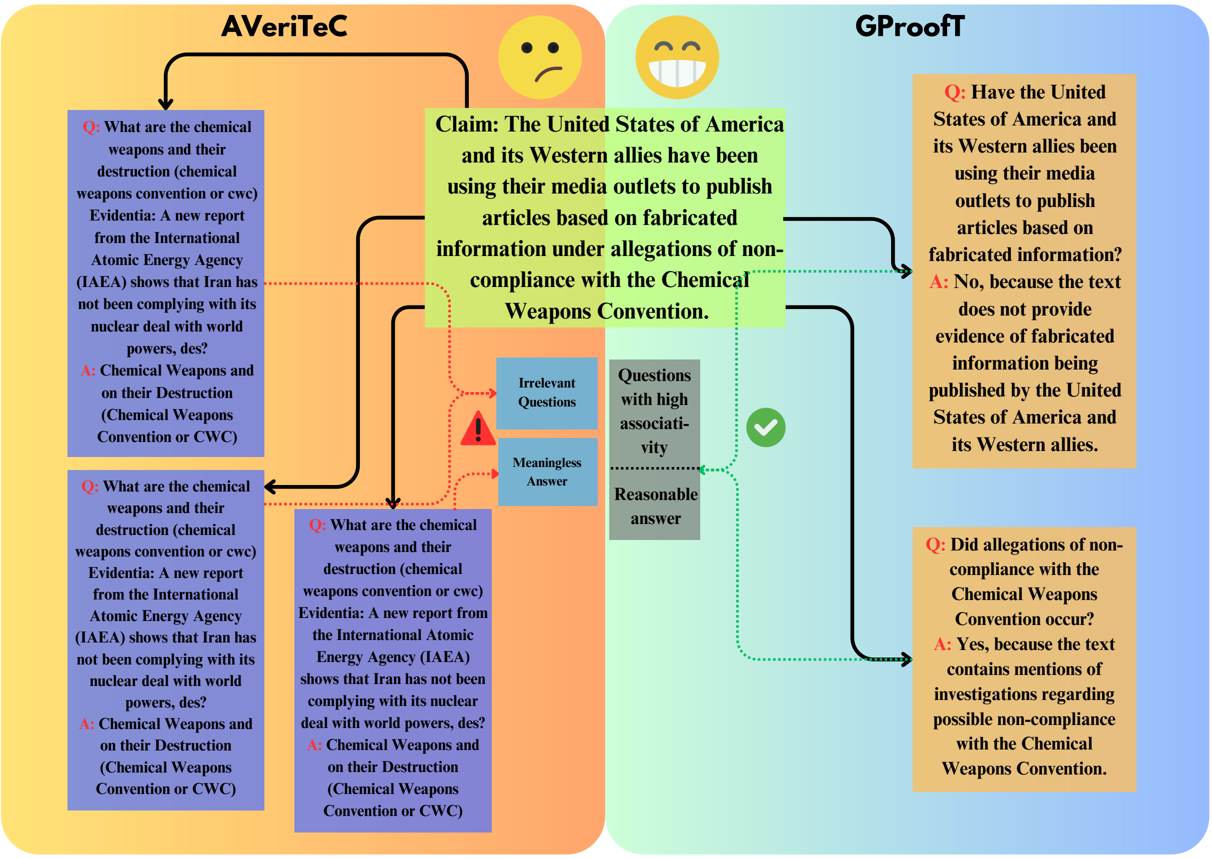

Figure 1: An overview structure of GProofT retrieval

subclaims in Claim Split, binary questions are generated respectively to validate the claim.

3. Answer Generation: Google Search API is employed to search for the questions in Question Generation and 9 relevant snippets are saved in the search results. ChatGPT is then adopted to determine whether they are supporting or refuting the original claim and generate the rationale.

After the retrieved evidence is obtained through our GProofT framework, we adopt LLMs to predict the label and benchmark our system based on the evaluation metrics proposed in AVeriTeC (Schlichtkrull et al., 2023). In-Context Learning (Agrawal et al., 2023; Hu et al., 2022b; Levy et al., 2023) and fine-tuning (Hu et al., 2022a; Xu et al., 2024) are employed for gpt-3.5-turbo and Llama-3 (Huang et al., 2024) respectively to improve the accuracy of prediction. Subsequently, extensive experiments are conducted to further investigate both the strengths and weaknesses of our framework. As our Question-answer score is lower than the Question-only score, we suspect that our binary answer with a subsequent rationale is not sufficient for language models to make more accurate predictions. In this case, future study could focus on instructing LLMs to generate more informative responses based on the retrieved snippets, which could subsequently assist the fact checking process of LLMs. Overall, our work could be summarized in three main aspects:

- We design claim fact extraction to divide claims into informative subclaims which could be beneficial for its downstream fact checking.
- We propose GProofT framework, a multi-dimension, multi-round fact checking framework which can conduct fact checking without heavy human intervention.
- We benchmark a series of LLMs with different techiniques incorporated to demonstrate a comprehensive LLMs evaluation on fact checking task.

2 Problem Definition

2.1 Dataset Description

We leverage the dataset proposed by Schlichtkrull et al. (2023) for training and benchmarking. The

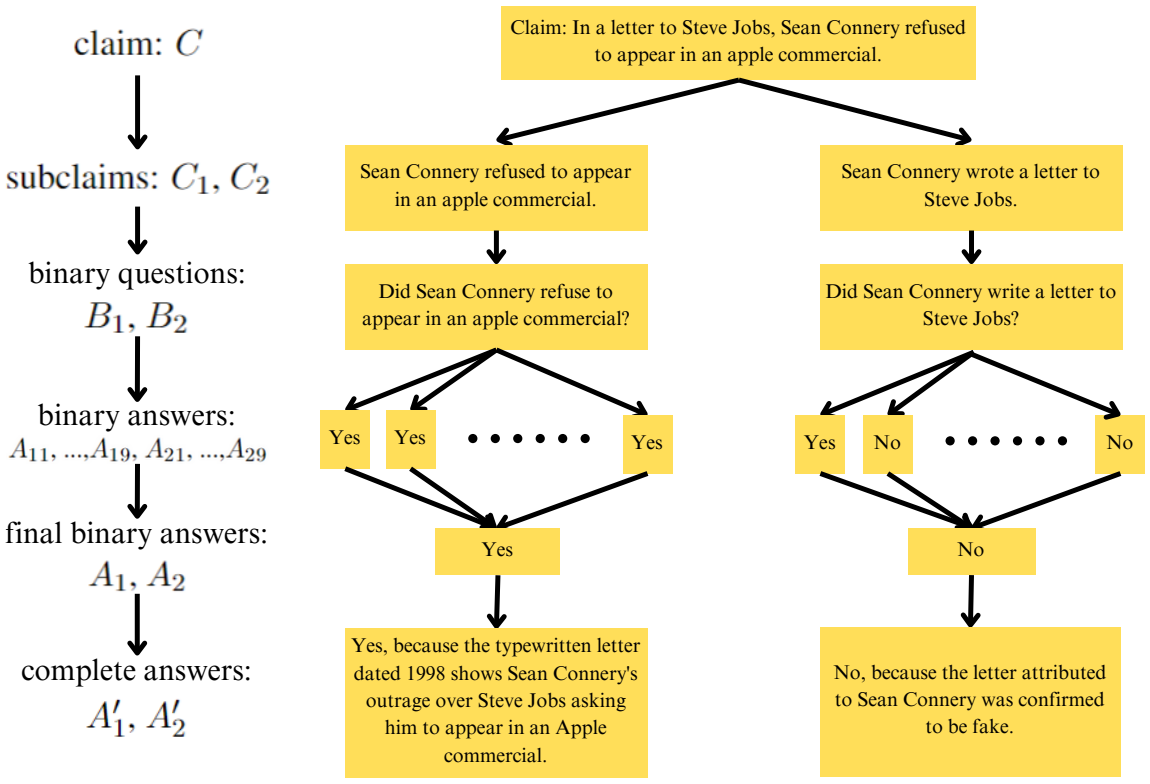

Figure 2: A comprehensive example of the GProofT retrieval process is provided by analyzing the claim, "In a letter to Steve Jobs, Sean Connery refused to appear in an Apple commercial." This example traces the progression from the original claim through to the final question-answer (QA) pair.

training set includes 3068 claims, while the development set and test set include 500 and 2215 claims respectively. The dataset contains claims accompanied by their gold evidence and labels prepared by a hired annotator as well as their metadata including speaker, publisher, date, and location. The claims are collected from 6661 fact-checking articles with duplicated and dead articles removed. The extraction of claims and metadata, question and answer generation, verdict prediction are completed by annotators. For each instance, the label, either *Supported*, *Refuted*, *Not Enough Evidence* or *Conflicted Evidence/CherryPicking* is assigned based on retrieved evidence. Specifically, *Supported* and *Refuted* indicate that the authenticity of the claim can be identified based on the evidence recovered. In case of insufficient evidence or conflicted retrieved evidence, *Not Enough Evidence* or *Conflicted Evidence/CherryPicking* will be attributed to the specific claim.

2.2 Task Definition

We follow the task definition formulated by Schlichtkrull et al. (2023). Formally, for each claim C, one or multiple QA pairs $\{Q_i, A'_i\}$ (i=1, 2, ..., n) are served as evidence, in which Q_i is a fact-checking question, and A_i is its complete answer. The objective is to predict the validity of the fact by leveraging the evidence retrieved. Specifically, we utilize LLMs to label each QA pair as supported, refuted, or irrelevant. Then we predict the label with the label of each QA pair.

3 System Overview

In this section, we would introduce the GProofT fact checking framework and elaborate our benchmark setup.

3.1 GProofT Fact Checking Framework

The GProofT fact-checking framework is a multi-dimension, multi-round fact checking framework. It decompose the original claims into several sub-claims, enabling a comprehensive evaluation from various dimensions and multiple rounds. The GProofT fact-checking framework consist of three stages: Claim Decomposition, Question Generation and Answer Generation. The overall framework pipeline is exhibited in Figure 2. We would

like to explain them in detail in the following paragraphs.

3.1.1 Claim Decomposition

By examining the instances, we observe that claims could be consisted of multiple opinions. Addressing these complex claims as singular entities can pose significant challenges for LLMs. Consequently, we decompose each claim C into several subclaims $C_1, C_2, ..., C_n$, ensuring that each resultant subclaim encompasses 1 to 2 facts.

To conduct decomposition, we employ a set of heuristic rules designed to guide ChatGPT (OpenAI, 2023) in effectively implementing this approach. The following rules outline this methodology:

- The overall mission involves instructing the LLMs to divide a claim into multiple subclaims based on the number of distinct facts it contains.
- Return only the subclaims, separated by periods rather than numbers.
- Avoid generating duplicated subclaims.
- Answers should be specific and avoid unnecessary pronouns to maintain clarity and conciseness.
- Limit subclaims to 15 words or less, ensuring they are shorter than the original claim.
- Capture the opinions or facts already present within it.
- Extract multiple subclaims, unless the claim is confined to a single fact.

Upon receiving the response from ChatGPT, we utilize SpaCy (Honnibal and Montani, 2017) to systematically split the subclaims. This process ensures that each subclaim is individually extracted, thereby deriving the subclaims from the original claim.

3.1.2 Question Generation

Subsequently, we proceed to generate the question of the QA pairs. We transform the subclaims into binary questions, which are structured to elicit yes or no responses. The heuristic rules adopted in this stage are as followed:

- Recognize the factual statement within the claim and formulate a binary question that can be used to verify this fact.
- Output the question directly without any rationales.
- Answers should be specific and avoid unnecessary pronouns to maintain clarity and conciseness.

3.1.3 Answer Generation

After preparing the binary questions, we employ the Google API to retrieve relevant evidence from online sources. For each question, 9 relevant snippets from 9 different websites are returned in the search results. Our pipeline then prompts ChatGPT to determine the binary answer for each search result, given both the question and the corresponding snippet. Following the resulted answers, we apply majority voting to determine the final binary response to the question. To give more insight on the rationale between the claim and each question, we expand the binary answer into a complete sentence that includes the binary response and the rationale derived from the snippet. Formally, given the question Q_i, the complete answer A_i is formulated as $\{$**Binary answer**, **Rationale**$\}$. The following heuristic rules are employed in this approach:

- Extend the initial binary answer into a comprehensive sentence.
- Answer the question directly without additional information.
- The answer should be formatted as "Yes, because ..." or "No, because ...".
- If the rationale for the answer is not provided in the snippet, respond with "Information not enough to judge the question".
- The word "snippet" should be avoided in the answer.

3.2 Label Prediction

3.2.1 Zero-shot learning

We benchmark the performance of different models under zero-shot setting. The evidence generated in previous stages is cohesively incorporated into the input-prompted sentence. For each claim, we obtain $\{C, \{Q_i, A_i\}\}$ from the retrieval process.

	Gold evidence		Baseline evidence		GProofT evidence	
Q only score	1.000		0.241		0.331	
Q+A score	1.000		0.185		0.204	
	macro F1	AVeriTeC score	macro F1	AVeriTeC score	macro F1	AVeriTeC score
Baseline Model	-	-	0.321	0.092	-	-
Zero-shot model						
GPT-3.5-turbo	0.387	0.472	0.166	0.076	0.180	0.096
Llama-3-8B-Instruct	0.341	0.640	0.263	0.108	0.288	0.166
Llama-3.1-8B-Instruct	0.404	0.730	0.327	0.114	0.288	0.174
falcon-7b-instruct	0.335	0.550	0.290	0.096	0.299	0.172
Gemma-2-2b-it	0.324	0.528	0.303	0.098	0.266	0.146
Gemma-2-9b-it	0.453	0.694	0.351	0.092	0.332	0.170
Mistral-7B-Instruct-v0.3	0.365	0.642	0.301	0.106	0.295	0.174
Mistral-Nemo-Instruct	0.383	0.632	0.297	0.086	0.333	0.172
Qwen2-7B-Instruct	0.417	0.654	0.317	0.090	0.311	0.166
Finetuned model						
GPT-3.5-turbo (one-shot)	0.532	0.656	0.243	0.080	0.347	0.166
Llama3-8B	0.607	0.806	0.361	0.112	**0.347**	**0.186**
Llama-3-8B-Instruct	0.629	0.786	0.332	0.114	0.321	0.162
Llama-3.1-8B	0.627	0.782	0.342	0.122	0.329	0.180
Llama-3.1-8B-Instruct	0.684	0.800	0.320	0.108	0.330	0.186
Mistral-7B-Instruct-v0.1	0.639	0.748	0.332	0.106	0.332	0.184
Mistral-7B-Instruct-v0.2	0.675	0.770	0.337	0.110	0.334	0.185
Mistral-7B-Instruct-v0.3	0.623	0.780	0.357	0.114	0.339	0.178
Qwen2-7B-Instruct	0.653	0.758	0.345	0.106	0.338	0.170

Table 1: Evaluation results on development set of AVeriTeC. The best performances are **bold-faced**. "Q only" and "Q+A" refer to Hungarian METEOR score (Schlichtkrull et al., 2023). "AVeriTeC" indicates using accuracy at $\lambda = 0.25$. We present the results of three distinct versions: utilizing gold evidence (Gold evidence), employing evidence from baseline (Baseline search), and utilizing evidence procured through GProofT (GProofT evidence).

To instruct the model to predict the label based on given evidence, we formulated the prompt as follows: Determine one most possible verdict for the claim "{C}", based on the given question and answer pairs Q: {Q_i} A: {A_i} (i=1, 2, ..., n).

3.3 Fine-tuning

To assess the effectiveness of GProofT across various settings, we fine-tune LLMs and evaluate them on the development set. Formally, for each instance {C, {Q_i, A_i}}, we integrate the claim with all QA pairs and fine-tune the model to predict the final label using the cross-entropy loss. Detailed settings and implementation of the fine-tuning process are discussed in 4.2.2.

4 Experiments

In this section, we will elaborate the data processing flow and the evaluation setting we adopted in the experiments.

4.1 Data processing

To construct comprehensive experiments, we preprocess three versions of the development set:

Gold Evidence: Gold evidence provided by the organizer is annotated manually. This evidence is considered highly reliable and has been meticulously curated for accuracy.

Baseline Evidence: The second type of evidence is retrieved by the organizer using a baseline model. This evidence serves as a comparison point to evaluate the performance of different systems.

GProofT Evidence: The last type of evidence is retrieved using our GProofT framework. This system has been optimized to improve the accuracy and relevance of the retrieved evidence.

We employ different LLMs to make verdicts on claims based on these different types of evidence, allowing us to assess system performance under various conditions.

4.2 Evaluation

We will introduce the evaluation experiments setup and analyze the experiment results in the following paragraphs.

4.2.1 Zero-shot

For the evaluation under zero-shot setting, we employ COT (Wei et al., 2022) and COT with self-

consistency (Wang et al., 2023b) prompting to generate the label for combined QA pairs of each claim. For ChatGPT, the temperature τ is set to 0.1 for non-Self-Consistency decoding and 0.7 otherwise. Specifically, for claims whose content is blocked by OpenAI filtering regulation, we set the label as *Not Answerable*. For other models under zero-shot setting, we adhere to the default configurations provided by HuggingFace. We benchmark different versions of LLAMA-3 (Huang et al., 2024), Mistral (Jiang et al., 2023), Falcon (Almazrouei et al., 2023), Gemma (Team et al., 2024), and Qwen2 (Yang et al., 2024).

4.2.2 Fine-tuning

We fine-tune the model using the label of claim in training set. Specifically, we input all QA pairs of one claim simultaneously into the LLMs and fine-tune it using the final label. During the evaluation phases, we maintain consistency with the training settings, distinguishing our approach from zero-shot learning.

For fine-tuning LLMs, we use the open-sourced library LLaMA-Factory (Zheng et al., 2024) to train all models with cross-entropy loss. All hyper-parameters follow the default settings, and a LoRA rank (Hu et al., 2022a) of $\alpha = 64$ is used. We fine-tune different versions of LLAMA-3, Mistral, and Qwen2. We conduct all experiments on a Linux machine with eight NVIDIA V100 GPUs.

4.3 Experiment results

The main results are demonstrated in Table 1. The evaluation metrics are consistent with the setting in Schlichtkrull et al. (2023), where we involve the Hungarian METEOR score, macro F1, and AVeriTeC at $\lambda = 0.25$. The evaluation results are obtained with the script. Our GProofT appendix checking framework achieves a Question Hu-meteor score(Banerjee and Lavie, 2005) of 0.331 and a Question+Answer Hu-meteor score of 0.204 on the development set of this shared task, encompassing the baseline. We observed that performance on our GProofT evidence generally surpasses that of the baseline, and fine-tuning significantly enhances model performance. The fine-tuned Llama3-8B model demonstrates the most outstanding performance on GProofT evidence, achieving the AVeriTeC score of 0.186, outperforming the baseline model. In the zero-shot setting, the Gemma-2-9b model consistently outperforms other models across three distinct datasets.

5 Analysis

In this section, we conduct error analysis and case study to further investigate the strengths and potential drawbacks of our framework. Furthermore, a imbalance prediction analysis is attached in appendix A to serve as a analysis on prediction distribution of our framework.

5.1 Error Analysis

The section analyzes the failure cases arise with GProofT framework. The issues could be categorized into two types: duplicated subclaims and biased claim split.

5.1.1 Duplicated Subclaims

When we processed the claim "Tanzania has not been affected by COVID-19." using our pipeline for subclaim generation, GPT initially produced two identical subclaims: "Tanzania was not affected by COVID-19." This occurred despite explicit instructions in the prompt to avoid generating duplicate subclaims. Similar problems have been observed with claims containing fewer than 15 words, as demonstrated in Table 2. We hypothesize that the phrasing of our prompt might incline GPT to generate more than one subclaim, leading to instances where the claim is unnecessarily split into multiple subclaims that are highly similar or identical.

5.1.2 Biased Claim Split

Occasionally, splitting claims may introduce inaccurate or irrelevant subclaims. For instance, in the development dataset, there is a claim that states "Over thirty million people... in the last several months had to file for unemployment." When processed by our pipeline, this claim was split into a subclaim stating "The last several months have passed." This subclaim clearly originates from the phrase "in the last several months" within the original claim. However, it represents a trivial fact rather than a meaningful assertion relevant to the original context. More examples are shown in Table 2.

5.2 Case Study

Generally, our framework is proved to be successful in most cases. In Table 3, We present several successful examples of the claim split process, which effectively avoided issues such as duplicate subclaims or biased splits. These instances correctly identified the key facts within the claims and generated corresponding subclaims. For instance,

Error Type	Claim	Subclaims one	Subclaims two
Duplicated subclaims	Viral video purportedly shows Pakistani opposition MPs chanted 2018Modi, Modi2019 (India's PM) inside the Pakistani Parliament.	Pakistani opposition MPs chanted 'Modi, Modi' inside the Pakistani Parliament.	Pakistani opposition MPs chanted 'Modi, Modi' inside the Pakistani Parliament.
Duplicated subclaims	All USA Ballots Arriving After Election Day Will Be Thrown Out.	All USA ballots arriving after Election Day will be thrown out.	Ballots arriving after Election Day will be thrown out.
Duplicated subclaims	Olive Garden prohibits its employees from wearing face masks depicting the American flag.	Olive Garden prohibits its employees from wearing face masks depicting the American flag.	Olive Garden prohibits employees from wearing face masks with American flag.
Biased claim split	Over thirty million people... in the last several months had to file for unemployment.	The last several months have passed.	Over thirty million people filed for unemployment.
Biased claim split	Edo State in Nigeria has a 19% unemployment rate.	The unemployment rate in Edo State is 19%.	Unemployment rate is a significant issue in Edo State.

Table 2: Failure examples of claim split part in GProofT Framework. The table categorizes two types of issues encountered in claim splitting: duplicated subclaims and biased claim splits.

Claim	Subclaim One	Subclaim Two
Congress MP Rahul Gandhi has been listed as the seventh most educated leader by Forbes.	Rahul Gandhi has been listed as the seventh most educated leader.	It is Forbes that has listed Rahul Gandhi as the seventh most educated leader.
Ulysses S. Grant, commander of the Union Army during the American Civil War, was a slave owner.	Ulysses S. Grant was the commander of the Union Army during American Civil War.	Ulysses S. Grant owned slaves.
Joe Biden proposed a US wide 2% property tax increase.	Joe Biden proposed a 2% property tax increase.	The tax increase that Joe Biden proposed apply to the entire US.

Table 3: Successful examples of claim split in GProofT Framework.In the majority of cases, GProofT Framework effectively identifies the facts within claims and splits them appropriately.

in the case of "Congress MP Rahul Gandhi has been listed as the seventh most educated leader by Forbes", the process not only accurately extracted the primary facts that Gandhi was listed as the seventh most educated leader and was featured by Forbes, but also leveraged the emphatic sentence structure to underscore these facts within the subclaims. This approach enhanced the effectiveness of the subsequent claim split process.

6 Conclusion

In this paper, we introduced GProofT, a multi-dimension, multi-round fact-checking framework designed to improve the efficacy and accuracy of validating online claims by leveraging LLMs and web evidence retrieval. Through extensive experiments, our approach demonstrated superior performance compared to baseline models, particularly in the critical task of evidence retrieval. Moreover, our framework requires less human labor involved in evidence checking which means it could be easily scale up when there is a huge amount of fact checking workload, improving the efficiency. Apart from such advantages, our framework also encounter challenges such as duplicated subclaims and biased claim splits, indicating areas for further improvement. Furthermore, refining the claim decomposition process and enhancing the handling of conflicting evidence will be crucial steps in advancing automated fact-checking systems. Our work contributes to the ongoing efforts to develop reliable, scalable, and automated tools for ensuring the trustworthiness of online information.

Acknowledgments

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20), and the GRF (16211520 and 16205322) from RGC of Hong Kong. We also thank the support from the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

Limitation

In our research pipeline, we employed GProofT Retrieval, incorporating the Google Search API and ChatGPT to generate question-answer (QA) pairs, which were subsequently utilized to inform predictions in conjunction with the Llama model for the labeling of numerous claims. Throughout this process, the API of Large Language Models was invoked multiple times. On average, the processing of each claim necessitated approximately 30 API calls to ChatGPT, leading to considerable computational overhead. Moreover, the heightened frequency of API calls led to a reduction in program execution speed, thereby impeding the efficient processing of large-scale datasets.

Ethics statement

All models and datasets accessed are freely accessible for research purposes and we do not create any harmful contents that would yield negative impact. The authors thus believe that this paper does not raise additional ethics concerns.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8857–8873. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Ching Nam Hang, Pei-Duo Yu, and Chee Wei Tan. 2024. Trumorgpt: Query optimization and semantic reasoning over networks for automated fact-checking. In *58th Annual Conference on Information Sciences and Systems, CISS 2024, Princeton, NJ, USA, March 13-15, 2024*, pages 1–6. IEEE.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022b. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2627–2643. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ArXiv*, abs/2311.05232.

Wei Huang, Xingyu Zheng, Xudong Ma, Haotong Qin, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. 2024. An empirical study of llama3 quantization: From llms to mllms. *Preprint*, arXiv:2404.14047.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1 – 38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *CoRR*, abs/2402.07401.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1401–1422. Association for Computational Linguistics.

OpenAI. 2023. Gpt-3.5 turbo.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin

Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Weiqi Wang, Tianqing Fang, Baixuan Xu, Chun Yi Louis Bo, Yangqiu Song, and Lei Chen. 2023a. CAT: A contextualized conceptualization and instantiation framework for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13111–13140. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2024. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models.

A Imbalanced Prediction

Model	S	R	C	N	Macro
baseline	.41	.69	.10	.16	.23
gpt-3.5 turbo	.57	.59	.08	.16	.34
llama3	.54	.74	.04	.06	.35
mistral	.55	.74	.00	.11	.35

Table 4: Performance of models on different categories of claim.

As demonstrated in Table 4, our model exhibits better performance on the "Supported" (S) and "Refuted" (R) labels but struggles with "Conflicting Evidence/Cherrypicking" (C) and "Not Enough Evidence" (N) labels. This performance discrepancy suggests a few potential reasons:

1. **Evidence Retrieval Challenges:** For Supported and Refuted labels, the evidence is clear and directly relevant, making it easier for the model to make accurate predictions. For Conflicting Evidence/Cherrypicking, the model struggles with retrieving or interpreting evidence that is contradictory or only partially relevant. If the model fails to retrieve diverse or contradictory evidence, it default to classifying the claim as either supported or refuted, missing the nuance required for the conflicting/cherrypicking evidence label.

2. **Training Data Imbalance:** The training data had more examples of claims with verdict supported or refuted, leading the model to be better at these tasks. Fewer examples of conflicting evidence or cherrypicking cases leads the model not have learned to handle these as effectively.