Progress Report: Subclaim Generation and Generative Retriever for Efficient Multi-Hop Fact Verification

Saeyoung Choi saeyoung.choi@snu.ac.kr
Jinwha Jang gdejin@snu.ac.kr
Yan GuangJing mysnuguangjing@snu.ac.kr

1 Introduction

As the cost of generating and delivering texts decreases, identifying misinformation and fact-checking becomes more important issues. Automatically generated misinformation is threatening to pollute the Internet. However, it is almost impossible to manually check the factuality of a massive number of texts generated by bots or language models. Hallucinations, plausible but incorrect text generation by language models, are also a difficult challenge to solve. Thus, the need for automated fact verification is urgent. With advances in natural language processing, attempts are being made to use various machine learning models for automatic fact verification [1, 2, 3].

Recently, retrieval-augmented generation (RAG) [4, 5] is drawing attention for its potential in improving factuality of the language models [6, 7]. It can not only improve factuality but also provide explainability of the generated results, by utilizing both LLM's prior knowledge (parameters) and external knowledge (retrieved document). However, it is difficult to verify complex multi-hop claims. Also, constructing a reliable knowledge base and searching through it is expensive.

In this study, we propose a generative approach to enhance the model performance and efficiency, by utilizing generative capability of the language model for not only claim verification, but also querying and retrieval. This approach includes (1) subclaim generation to verify the complex factuality within multi-hop claims (like chain-of-thought approach) and (2) generative retrieval to dynamically obtain relevant evidence for each claim. We will integrate these two approaches into our baseline model, and evaluate the performance of the integrated model.

2 Related Work

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

QACHECK [8] emerges as a groundbreaking model in the realm of automated fact verification, particularly within natural language processing (NLP). It innovatively adopts a question-guided multi-hop reasoning framework, significantly enhancing the clarity and effectiveness of fact-checking processes. Unlike traditional models that rely on linear evidence assessment, QACHECK dissects complex claims into simpler components through a series of contextually relevant questions. This methodology not only increases the interpretability of the verification process but also aligns closely with human cognitive patterns in assessing veracity.

The architecture of QACHECK is a composite of five key modules: a claim verifier, a question generator, a question-answering module, a QA validator, and a reasoner. Each module plays a distinct yet interdependent role in the fact-checking pipeline. The claim verifier initially assesses the sufficiency of available context in substantiating a claim. Sequentially, the question generator and the QA module collaboratively work to explore and fill informational gaps. This synergy ensures a comprehensive examination of claims, thereby enhancing the accuracy of verification.

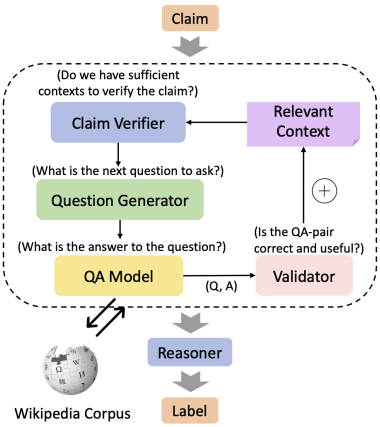

Figure 1: The Architecture of the QACHECK System

QACHECK's methodological innovation lies in its unique use of question-answer pairs for evidence validation, a significant deviation from conventional fact-checking models. By iteratively generating and validating these pairs, the model effectively accumulates a robust evidence base for each claim. This approach is particularly adept at tackling claims requiring deep, multi-faceted reasoning, a common challenge in NLP tasks focused on misinformation and fact verification. The integration of these components into a cohesive system offers a more transparent, user-friendly, and dynamic fact-checking process, addressing the ever-growing complexities in discerning factual accuracy in digital content.

2.2 Generative Retrieval Models

In previous works, most of the generative models applied to the information retrieval domain obey the "index-retrive-then-rank" principle. Such as the re-implementation of BERT for query-based passage re-ranking proposed by Nogueira et al. [9] as well as the DPR [10] method used in the RAG model's retriever. However, this kind of classical approach requires a large document index and a complicated search process, which leads to considerable memory and computational overhead. Additionally, independent scoring and ranking paradigms lack dependent information among documents and sentences. A fixed number of selected sentences are not precise in most cases and can result in lower accuracy.

To mitigate these limitations, Chen et al. [11] proposed Generative Evidence Retrieval for Fact Verification (GERE) to adapt a pre-trained sequence-to-sequence model to dynamically select a precise set of relevant evidence for each claim by modeling the dependency among documents and sentences.

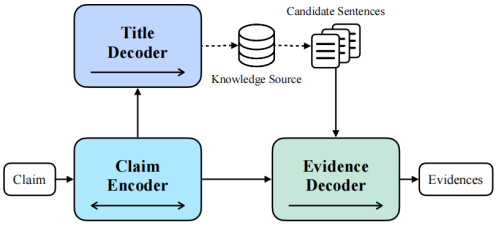

Figure 2: The Overview of the GERE Framework

GERE consists of 3 independent components, including claim encoder, title decoder and evidence decoder. It first encodes the claim into a compact vector with BART to capture its essential topics. Then the title decoder predicts a sequence of document titles based on both the claim and previous generated titles, here Beam Search is adopted for token prediction to save computational resources.

Generated titles are used to retrieve relevant documents which contribute to a candidate sentence set. The sentence set are used to predict sentence identifiers, which generates a dynamic set of evidence.

The experimental results on FEVER dataset [12] gives the best performance in terms of precision and F1 score, but fails in recall for sentence retrieval. This indicates that GERE provides a more compact but more preciser evidence set. Besides, GERE has a significant reduction of memory footprint and inference time of document retrieval, suggesting that it is much more resource-saving than other baseline models. By combining GERE with LLM, we hypothesize that our model can generate a more consistent output with less resource demand.

3 Model Description

3.1 Model Overview

Figure 3 and Figure 4 shows the architecture of our baseline model and proposed model, respectively. While the baseline model has conventional retriever and claim verifier, our proposed model consists of the following modules: (1) subclaim generator, (2) generative retriever (evidence generator), and (3) claim verifier.

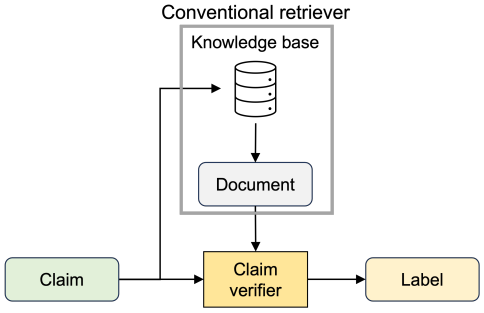

Figure 3: Baseline Model Architecture

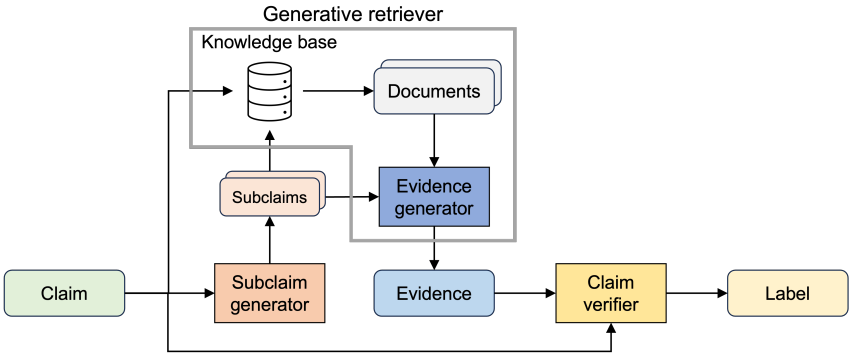

Figure 4: Proposed Model Architecture

3.2 Baseline model

Baseline model consists of retriever and claim verifier. The retriever first retrieves the documents that are similar to given claim, by querying to the knowledge base. Claim and retrieved documents are put into the claim verifier, which then returns the factuality label (True, False, or Not Verifiable).

Experiment with baseline model

Our baseline model is constructed based on LlamaIndex [13], a framework for LLM-based application. Wikipedia documents from FEVER dataset (5,416,537 documents) are stored in vector database

(pgvector on PostgresSQL), after transformed to embedding vectors. BAAI's embedding model [14] is used as sentence transformer. When a claim is given, it is transformed to a query embedding. Then, the documents are retrieved from the knowledge base based on the similarity with the query embedding. Claim and contexts (retrieved documents) are fed to Llama 2 [15] model with fact-checking prompt. The baseline model is tested on claim samples from FEVER and HOVER datasets. It should be noted that HOVER dataset has only two labels (Supported and Not supported) because of ambiguity between False and Not Verifiable in multi-hop claims. It is clearly shown that the conventional retrieval model fails to correctly verify the claims with multi-hop.

	FEVER	HOVER		
	1-hop	2-hop	3-hop	4-hop
EM	0.49	0.44	0.44	0.41
F1 score	0.47	0.55	0.43	0.39

Table 1: Baseline model performance on FEVER and HOVER datasets

3.3 Subclaim generator

The subclaim generator in QACHECK plays a crucial role in breaking down complex claims into contextually relevant subclaims, using InstructGPT. This process generates essential initial and subsequent follow-up questions, which are integral to the system's multi-hop reasoning approach. This method ensures a comprehensive and thorough fact-checking process.

3.4 Generative retriever

The generative retriever consists of 3 independent components, including claim encoder, title decoder and evidence decoder. It first encodes the claim into a compact vector to capture its essential topics. Then the title decoder predicts a sequence of document titles based on both the claim and previous generated titles. Generated titles are used to retrieve relevant documents which contributes candidate sentence set. The sentence set are used to predict sentence identifier, which generates a dynamic set of evidence.

3.5 Evaluation

To evaluate the fact verification performance of the proposed model, we will use the fact verification datasets: FEVER, HOVER [16], and FEVEROUS [17]. They are open-domain fact verification dataset based on Wikipedia articles. While FEVER mostly has single-hop claims (related to a single document), HOVER and FEVEROUS have multi-hop claims (related to multiple documents). F1 score will be used as evaluation metric. Other sub-tasks and evaluation metrics will be proposed in further study: e.g., performance of subclaim generation, performance of generative retrieval, memory efficiency in knowledge base storage, and time-efficiency in inference.

4 Future Work and Improvement

4.1 Integrating Subclaim Generation from QACHECK into baseline model

Our primary objective is to enhance the capability of our existing NLP system especially in the claims generators parts, through the integration of QACHECK. This integration aims to leverage the sophisticated, question-guided multi-hop reasoning mechanism of QACHECK for more effective and interpretable fact-checking processes within our NLP tasks.

We will align current retrieval component with QACHECK's subclaim generation module. This will involve modifying the retrieval process to utilize the output of the subclaim generator to get more effective information retrieval. The integration will require modifying the data flow process so that the input to the subclaim generator comes directly from the LlamaIndex's output. Then, subclaims generated by QACHECK will be fed back into LlamaIndex's retrieval and embedding

modules. This can create a recursive retrieval process, enhancing the overall relevance and accuracy of the information gathered.

While Instruct GPT is effective in generating contextually relevant questions, but integrating transformer models like BERT or GPT-3 could provide additional improvements. These have advanced capabilities in understanding nuances and complex language structures, which can lead more precise and varied subquestions.

4.2 Integrating GERE as a retriever module to our baseline model

In our pursuit of generating evidence that is not only more flexible and aligned with factual accuracy but also ensuring improved time-efficiency and memory-efficiency, we are strategically incorporating the GERE into our processing system as a dedicated retriever module.While GERE can be easily adapted to our claim verification model, there also remain some space for future improvement. By incorporating additional contextual information during the sentence prediction step, GERE could capture a more comprehensive set of relevant evidence. By seamlessly integrating GERE, we aim to enhance the speed, and resource utilization of our fact-validation system, thereby advancing its effectiveness in delivering precise and reliable results in a timely and resource-conscious manner.

References

[1] Sahil Chopra, Saachi Jain, and John Merriman Sholar. Towards automatic identification of fake news: Headline-article stance detection with lstm attention models. *Proc. Stanford CS224d Deep Learn. NLP Final Project*, pages 1–15, 2017.

[2] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. Reasoning over semantic-level graph for fact checking. *arXiv preprint arXiv:1909.03745*, 2019.

[3] Yixin Nie, Haonan Chen, and Mohit Bansal. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6859–6866, 2019.

[4] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[5] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

[6] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.

[7] Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*, 2023.

[8] Liangming Pan, Xinyuan Lu, Min-Yen Kan, and Preslav Nakov. Qacheck: A demonstration system for question-guided multi-hop fact-checking. *arXiv preprint arXiv:2310.07609*, 2023.

[9] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[10] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[11] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. Gere: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2184–2189, 2022.

[12] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[13] LlamaIndex. https://www.llamaindex.ai/.

[14] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[15] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[16] Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. Hover: A dataset for many-hop fact extraction and claim verification. *arXiv preprint arXiv:2011.03088*, 2020.

[17] Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*, 2021.