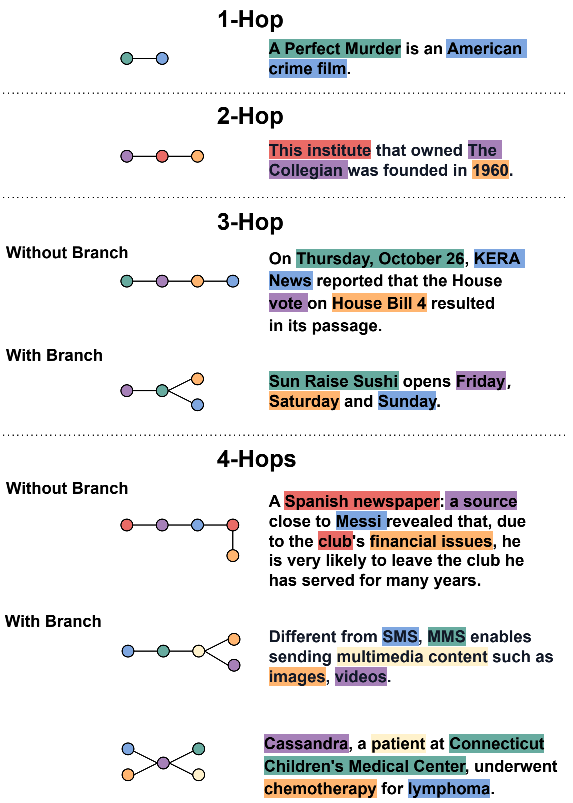
FactCG: Enhancing Fact Checkers with Graph-Based Multi-Hop Data

Deren Lei* **Yaxi Li*** **Siyao Li*** **Mengya Hu** **Rui Xu**
Ken Archer **Mingyu Wang** **Emily Ching** **Alex Deng**

Microsoft Responsible AI
{***derenlei, yaxi.li, lisiyao****, yuetc, alex.deng*}*@microsoft.com*

Abstract

Prior research on training grounded factuality classification models to detect hallucinations in large language models (LLMs) has relied on public natural language inference (NLI) data and synthetic data. However, conventional NLI datasets are not well-suited for document-level reasoning, which is critical for detecting LLM hallucinations. Recent approaches to document-level synthetic data generation involve iteratively removing sentences from documents and annotating factuality using LLM-based prompts. While effective, this method is computationally expensive for long documents and limited by the LLM's capabilities. In this work, we analyze the differences between existing synthetic training data used in state-of-the-art models and real LLM output claims. Based on our findings, we propose a novel approach for synthetic data generation, CG2C, that leverages multi-hop reasoning on context graphs extracted from documents. Our fact checker model, FactCG, demonstrates improved performance with more connected reasoning, using the same backbone models. Experiments show it even outperforms GPT-4-o on the LLM-AGGREFACT benchmark with much smaller model size. [1]



1-Hop
A Perfect Murder is an American crime film.
2-Hop
This institute that owned The Collegian was founded in 1960.
3-Hop
Without Branch
On Thursday, October 26, KERA News reported that the House vote on House Bill 4 resulted in its passage.
With Branch
Sun Raise Sushi opens Friday, Saturday and Sunday.
4-Hops
Without Branch
A Spanish newspaper: a source close to Messi revealed that, due to the club's financial issues, he is very likely to leave the club he has served for many years.
With Branch
Different from SMS, MMS enables sending multimedia content such as images, videos.
Cassandra, a patient at Connecticut Children's Medical Center, underwent chemotherapy for lymphoma.

Figure 1: **Context Graphs in LLM Responses**. Red nodes represent contextualized entities that requires further bridging for coreference resolution.

1 Introduction

Large language models (LLMs) have shown remarkable progress in generating fluent and coherent text across domains. They are capable of generating text conditioned on documents, leading to various applications, such as summarization (Zhang et al., 2024), retrieval augmented generation (Lewis et al., 2020), code completion (Roziere et al., 2023), etc. However, one of the major challenges faced by these models is the tendency to generate hallucinated statements that are factually incorrect or not grounded in the provided document (Maynez et al., 2020; Huang et al., 2023; Zhang et al., 2023). Detecting these hallucinations is critical for providing trustworthy and responsible LLM services.

To identify hallucinations in LLM outputs (claims), recent efforts focuses on LLM-based approaches and training grounded factuality classification models. LLM-based approaches (Manakul et al., 2023; Liu et al., 2023b; Lei et al., 2023; Dhuliawala et al., 2023) leveraging prompts are easy to use and benefit from the strong generalization capabilities of LLMs. However, they are not cost-effective for real application use as they often require asking LLM to generate claims multiple times (Manakul et al., 2023), sending each entity of

[*] Equal contributions.
[1] https://github.com/derenlei/FactCG

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5002–5020
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

interest iteratively to identify faithfulness (Lei et al., 2023), or leveraging chain-of-thought to generate long sequence of reasoning chain per claim judgement (Liu et al., 2023b; Dhuliawala et al., 2023). Therefore, they are mainly used as annotation tools to guide the training of smaller grounded factuality classifiers or treated as strong baselines for benchmark comparison purposes. To make online hallucinations detectors, various studies focuses on training grounded factuality classification models. Their training process usually leverages natural language inference (NLI) datasets (Laban et al., 2022), adapting other reasoning tasks to the NLI framework (Zha et al., 2023) or synthetic data generation techniques (Tang et al., 2024a). Our approach lies under this category with a focus on adapting and generating more complex data.

In this work, we investigate the patterns of existing training data and compare it to real LLM-generated claims. We notice there is a non-negligible difference in the complexity of claims and the reasoning process required for claim verification on grounding documents. We propose context graph to claim (CG2C), a synthetic data generation approach that is able to generate complex multi-hop claims in a more fine-grained and cost-effective way than past studies without any human annotation effort. It also does not leverage LLM-based approaches to generate training data labels, hence it is not bounded by LLM-based approaches' performances. We verify its effectiveness on top of state-of-the-art models by teaching them to do complex multi-hop reasoning. Experiments show that our model, FactCG, achieves state-of-the-art performance on the LLM-AGGREFACT (Tang et al., 2024a) grounded factuality benchmark compared to trained models under a billion parameters. Its performance is not bounded by GPT that is used to construct CG2C. Moreover, FactCG does more connected reasoning than leveraging dataset artifacts.

Our contributions are three-fold:

- We analyze the existing synthetic data generation for training grounded factuality models and identify the gap in claim complexity.
- We study the effectiveness of CG2C, a data generation approach requires no human supervision. It is able to control the complexity of generated claims with high granularity.
- We propose a fact checker: FactCG that achieves state-of-the-art average performance on the LLM-AGGREFACT benchmark compared to other models of comparable size and outperforms GPT-4-o.

2 Related Work

Fact-Checking Methods. There are two main approaches to detect hallucinations without altering LLM architectures. One relies on LLMs evaluating their own outputs, like SelfCheckGPT (Manakul et al., 2023), or leveraging Chain-of-Thought strategies, as seen in G-Eval (Liu et al., 2023b), CoVe (Dhuliawala et al., 2023), and CoNLI (Lei et al., 2023). The other focuses on training cost-efficient factuality classifiers. SummaC (Laban et al., 2022) adapts NLI models for document-level factual checks, while QAFactEval (Fabbri et al., 2022) uses Question Answering (QA) metrics. FactCC (Kryscinski et al., 2020) predicts summary consistency, and AlignScore (Zha et al., 2023) trains on large-scale unified NLI data. MiniCheck (Tang et al., 2024a) generates training data using LLMs to label and outperforms previous work on human-annotated real LLM hallucination benchmark LLM-AGGREFACT (Tang et al., 2024a). Our work continue the study on improving fact-checking models by enhancing LLM-generated data complexity and controllability.

Fact-Checking Data. Several studies focus on creating datasets for hallucination detection, either for training or benchmarking. AGGREFACT (Tang et al., 2023) evaluates factual consistency in summarization (CNN/DM, XSum), while ToFuEval (Tang et al., 2024b) targets dialogue summarization. WiCE (Kamoi et al., 2023) is a textual entailment dataset based on Wikipedia claims. REVEAL (Jacovi et al., 2024) benchmarks Chain-of-Thought reasoning in open-domain QA, and ClaimVerify (Liu et al., 2023a) evaluates correctness and groundedness in generative search engines' responses. FactCheck (Wang et al., 2023) constructs a document-level factuality benchmark, while EXPERTQA (Malaviya et al., 2024) focuses on long-form QA with verified attributions. LFQA (Chen et al., 2024) assesses LLM-generated sentences for factual support by the documents that are either retrieved by humans, models, or randomly selected. RAGTruth (Niu et al., 2024) analyzes word-level hallucinations in various domains and tasks within the standard retrieval-augmented generation frameworks for LLM applications. HaluEval (Li et al., 2023) is an LLM evaluation benchmark with synthetically generated hallucination data.

Multi-hop Reasoning. Multi-hop reasoning refers to inference over multiple steps. Various studies focused on generating QA datasets where multi-hop reasoning on documents is required to get the answer, such as HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), Musique (Trivedi et al., 2022). However, various studies show that models often exploit dataset artifacts to produce correct answers without connecting information across multiple supporting facts even when multi-fact reasoning is required (Jiang and Bansal, 2019; Chen and Durrett, 2019; Min et al., 2019), namely disconnected reasoning (DiRe) (Trivedi et al., 2020).

LLM-based Graph Construction. Recent literatures have shown the effectiveness of LLM in graph construction tasks. Trajanoska et al. leverages LLM to enhance knowledge graph construction from unstructured text. KG-LLM (Yao et al., 2023) explored the LLM-base approach to complete the knowledge graph. Graph RAG (Edge et al., 2024) utilized LLM generated graphs to improve the RAG system on global questions that target the entire text corpus.

3 Problem Formulation

Given a document doc and a claim c, we consider c to be grounded in doc if a generic reader would affirm the statement "According to doc, c is true". Conversely, c is ungrounded with respect to doc if it conflicts with or cannot be verified against doc. Our objective is to model a function to detect ungrounded hallucinations in LLM-generated claims against documents at the sentence-level.

We define grounded as 1 and ungrounded as 0. $\mathcal{M}$ is a classifier that first maps each $\langle doc, c\rangle$ pair into a confidence score of judging c is grounded in doc, formally defined as $\mathcal{M}(doc, c) \in [0, 1]$. We then use a threshold to convert confidence score into a binary prediction

$$y = P(\mathcal{M}(doc, c) \geq \theta \mid doc) \qquad (1)$$

where θ is the threshold we can select.

As most recent grounded factuality detection industrial applications now support dynamic threshold tuning (Google, 2024; Amazon, 2024), we consider the threshold to be dynamically adjustable to its optimal level for per-scenario detection and follow the past work (Laban et al., 2022; Zha et al., 2023; Tang et al., 2023, 2024b).

4 Analysis on Multi-Hops in Claims

In this section, we conduct a proof of concept experiment on the multi-hop in LLM-generated claims. We try to answer the following two research questions (RQ):

RQ1: To what extent do LLM-generated claims, based on grounding documents, require multi-hop reasoning for fact verification?

RQ2: To what extent do current state-of-the-art synthetic claims, based on grounding documents, require multi-hop reasoning for fact verification?

4.1 Analysis Setup

LLM Claims. To answer RQ1, we select the RAGTruth (Niu et al., 2024) validation dataset from the LLM-AGGREFACT benchmark (Tang et al., 2024a) for analysis, as it contains three recognized tasks with RAG settings: Question Answering (QA), Data-to-text Writing (Data2Text), and News Summarization (Summ). It also leverages six different LLMs to generate completion responses. We analyze each generation task separately. To preserve the original LLM completion format, we do not further decompose the claim into atomic claims. For each task, we deduplicate the documents in the entire set and randomly select 500 documents and their corresponding document-claim pairs for analysis to reduce the computational cost.

Synthetic Claims. To answer RQ2, we select MiniCheck's D2C synthetic claims (Tang et al., 2024a) for analysis as the classifier model trained on it achieves the best result on LLM-AGGREFACT. Same as RAGTruth claims, we do not further process the D2C claims and analyze them on a sentence-level basis as provided. We conduct analysis on the entire dataset without random sampling.

Multi-hop Context Sub-graph Extraction. As the LLMs generate claims conditioned on grounding documents, they are not always decontextualized and may contain implicit reasoning chains. To extract the complete reasoning chain of each claim for both datasets, we leverage LLMs in the following steps:

Firstly, we construct the context graph $\mathcal{G}$ from the grounding document doc. We follow GraphRag's (Edge et al., 2024) prompt-based approach to extract $\langle entity, entity, relation\rangle$ triples from doc, where the $relation$ is a short sentence that describes how the two entities are connected based on the context of the document, which is non-directional. We extract all triples and further group them so that each triple shares at least one entity within its cluster.

Secondly, we find the context sub-graph $\mathcal{G}_c$ for each claim c by selecting the triples related to it. The $\mathcal{G}_c$ contains the information in c that requires verification against doc.

Thirdly, we count the number of connected triples in each $\mathcal{G}_c$ as the hop number for c. If $\mathcal{G}_c$ has multiple sub-graphs, we count the hops of the largest sub-graph as the hop number.

We calculate the distribution of hops in the reasoning chains extracted for each dataset.

4.2 Experiment Settings

We use GPT-4-o for context graph extraction and context sub-graph mapping with claims. We remove data with ill-formatted GPT outputs in both steps to simplify the post-processing. This experiment is only conducted on positive samples in every dataset because ungrounded claims may take any form and their reasoning chain may not exist in the source document's graph.

4.3 Results and Analysis

We show multi-hop extraction results in Table 1. Synthetic data in MiniCheck's D2C dataset lean towards claims with fewer hops, while LLM-generated claims in RAGTruth tend to have higher ratios of 2-hop and 3-hop claims. We even found non-negligible amounts of claims with 4 or more reasoning hops in RAGTruth across all tasks, which are very limited in D2C. We believe bridging this gap is a non-trivial task and we hypothesize that teaching the model to learn multi-hop inference may improve the model's overall capability for hallucination detection.

Hops	Data2Text	Summ	QA	D2C
1-hop	42.6%	42.7%	51.1%	71.0%
2-hop	32.1%	36.6%	32.9%	22.6%
3-hop	17.3%	15.2%	10.9%	5.3%
4-hop	6.5%	3.4%	3.3%	1.1%
$\geq$ 5-hop	1.4%	2.0%	1.9%	0.0%

Table 1: **Claim Hop Analysis.** Hop distribution of real LLM-generated claims in RAGTruth data2text, summarization and question-answering split versus MiniCheck D2C synthetic claims.

5 Improving Model with Context Graph to Multi-hop Claim (CG2C)

In this section, we study whether we can improve the state-of-the-art grounded factuality models with synthetically generated data via context graph to multi-hop claims (CG2C). Specifically, the study follows two steps, each addressing a research question:

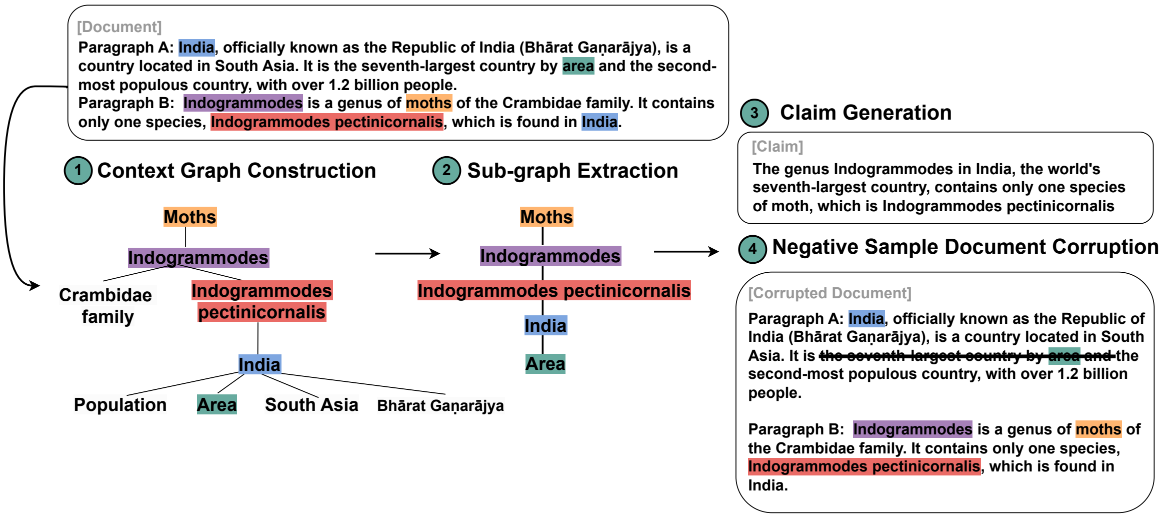
RQ3: Can we improve grounded factuality models by generating synthetic data leveraging large-scale public multi-hop datasets?

RQ4: Can we improve grounded factuality models by generating synthetic multi-hop data leveraging documents only?

5.1 CG2C from Multi-hop QA

In this section, we study RQ3. We leverage publicly available multi-hop question answering (MHQA) datasets to synthetically construct grounded factuality data for model training, namely CG2C-MHQA. Specifically, we use HotpotQA (Yang et al., 2018) and Musique (Trivedi et al., 2022), where each data sample contains a source document doc, a question q, an answer ans and supporting sentences $doc_{support}$, which is a subset of doc that provides evidence for the answer. We leverage LLMs to convert $\langle ans, q\rangle$ to a single declarative statement. We define the converted statement as claim c.

HotpotQA. We leverage the train-hard split in HotpotQA to ensure each data sample requires complex multi-hop reasoning across multiple paragraphs. We treat the $\langle doc, c\rangle$ pair as a positive sample. We then randomly select positive samples to corrupt them into negative samples by following the steps below. Firstly, since $doc_{support}$ is provided, we extract $\mathcal{G}_s$ from $doc_{support}$ and then leverage the $\langle \mathcal{G}_s, c\rangle$ pair to get a context sub-graph $\mathcal{G}_c$, using the same strategy described in Section 4.1. Secondly, from the $\mathcal{G}_c$, we randomly select a triple and leverage LLMs to remove the relation between entities in the triple from doc to form doc_{neg}. Finally, the $\langle doc_{neg}, c\rangle$ pair forms a negative sample. Inspired by Nie et al. (2020), we further reduce the dataset size by leveraging RoBERTa-large trained on the MultiNLI dataset, (Williams et al., 2018) to select difficult samples. As the NLI model predicts each sample from $\{\text{Entailment}, \text{Contradiction}, \text{Neutral}\}$, we define a mapping function $f(\cdot)$ for the model output y as:



[Document]

Paragraph A: India, officially known as the Republic of India (Bhārat Gaṇarājya), is a country located in South Asia. It is the seventh-largest country by area and the second-most populous country, with over 1.2 billion people.

Paragraph B: Indogrammodes is a genus of moths of the Crambidae family. It contains only one species, Indogrammodes pectinicornalis, which is found in India.

[Claim]

The genus Indogrammodes in India, the world's seventh-largest country, contains only one species of moth, which is Indogrammodes pectinicornalis

[Corrupted Document]

Paragraph A: India, officially known as the Republic of India (Bhārat Gaṇarājya), is a country located in South Asia. It is ~~the seventh-largest country by area and~~ the second-most populous country, with over 1.2 billion people.

Paragraph B: Indogrammodes is a genus of moths of the Crambidae family. It contains only one species, Indogrammodes pectinicornalis, which is found in India.

Figure 2: **CG2C from Document.** To generate synthetic data with documents only, first we construct context graph $\mathcal{G}$ from document doc. Second we extract sub-graph $\mathcal{G}_c$ from the context graph as multi-hop. Third we generate claim c with context $\mathcal{G}_c$. To corrupt the document we remove a random relation between entities within $\mathcal{G}_c$ to get doc_{neg}. Finally, we get positive sample $\langle doc, c\rangle$ and negative sample $\langle doc_{neg}, c\rangle$. For CG2C-MHQA, we get c from $\langle q, ans\rangle$ and use $\langle \mathcal{G}, c\rangle$ to get $\mathcal{G}_c$ instead of step 2 and 3 mentioned above.

$$f(y) = \begin{cases} \text{Grounded} & \text{if } y = \text{Entailment}, \\ \text{Ungrounded} & \text{if } y \in \{\text{Contradiction, Neutral}\} \end{cases}$$

We remove data that RoBERTa$_{\text{MNLI}}$ [2] predicts correctly.

Musique. We leverage 3-hop and 4-hop data from the Musique dataset to diversify from HotpotQA where most of the data is 2-hop. The $\langle doc, c\rangle$ pairs with c converted from unanswerable $\langle q, ans\rangle$ pairs are treated as negative samples and answerable pairs are treated as positive samples. Similar to HotpotQA, we remove data that RoBERTa$_{\text{MNLI}}$ predicts incorrectly.

Merge. We merge synthetically constructed data from HotpotQA and Musique into CG2C-MHQA.

5.2 CG2C from Documents

In this section, we study RQ4. We further investigate the possibility of leveraging any document to directly generate synthetic training data, namely, CG2C-Doc. We propose the following steps to generate harder and more diverse data with fine-grained control:

Firstly, we extract a context graph from the given document doc following the same approach as in Section 4.1. Secondly, after filtering out cyclic graphs, we programmatically extract sub-graphs with a specific number of hops and shapes. The types of sub-graphs we extract are shown in Figure 1. Thirdly, for each generated sub-graph, we leverage LLMs to generate a claim c which includes all graph nodes. The $\langle doc, c\rangle$ pair forms a positive data sample. Lastly, we randomly extract a relation from the sub-graph, and request LLMs to remove it from the doc to create the document for negative sample $\langle doc_{neg}, c\rangle$.

Unlike previous work (Tang et al., 2024a), where LLMs are used directly to decide the synthetic samples' labels, our proposed data generation strategy avoids involving LLMs in label generation, hence is less bounded by the LLM prompt-based approach's performance. We can also easily generate more complex data (3-hop or 4-hop samples) for each document by controlling the sub-graph we extract.

For a fair comparison with the current state-of-the-art MiniCheck (Tang et al., 2024a), we use the same documents from its D2C dataset to generate CG2C-Doc. We only use data from 3-hop and 4-hop sub-graphs without branches for training because we find them empirically better. We analyze this result in the Ablation Study Section 6.4.

5.3 Proposed Fack Checker: FactCG

Model Backbones and Synthetic Data Baseline. We follow MiniCheck (Tang et al., 2024a) training data and the two-stage training process as the synthetic data training baseline. We also select the same model backbones as MiniCheck for easier

[2] https://huggingface.co/FacebookAI/roberta-large-mnli/tree/main

experiment comparison, namely RoBERTa-large (RBT) (Liu, 2019), DeBERTa-v3-large (DBT) (He et al., 2023) and Flan-T5-large (FT5) (Chung et al., 2024). We add our synthetically generated data into the MiniCheck training pipeline and perform an apple-to-apple comparison.

Model Input and Output. We do conventional sequence pair classification for FactCG-RBT. For FactCG-FT5, we use the FLAN instruction (Chung et al., 2024) for the ANLI (Nie et al., 2020) dataset as the input template. At inference time, we leverage "Yes" and "No" logits from the language model head as the final classification logits. For FactCG-DBT, we use the same FLAN input template as FactCG-FT5 and perform sentence classification training. We find it to be empirically effective. We apply the softmax function to transform the model's raw logits then use threshold θ to get final binary output y.

Chunking at Inference Time. Following recent works (Tang et al., 2024a; Zha et al., 2023), we chunk the document into sequential chunks at sentence boundaries during inference time. We set the target chunk size for FactCG-RBT, FactCG-DBT, and FactCG-FT5 to 400, 550, and 550 tokens respectively. The highest score across chunks will be treated as the final score.

6 Experiments

In this section, we compare FactCG models with other approaches on grounded factuality datasets.

6.1 Datasets

We leverage LLM-AGGREFACT benchmark (Tang et al., 2024a), which consists of real hallucinations generated from recent language models. It includes 11 datasets, namely AGGREFACT-CNN, AGGREFACT-XSum (Tang et al., 2023), TofuEval-MediaS, TofuEval-MeetB(Tang et al., 2024b), WiCE (Kamoi et al., 2023), REVEAL (Jacovi et al., 2024), ClaimVerify (Liu et al., 2023a), FactCheck (Wang et al., 2023), EXPERTQA (Malaviya et al., 2024), LFQA (Chen et al., 2024) and RAGTruth (Niu et al., 2024).

6.2 Experiment Setup

Synthetic Data Construction. We use GPT-4-o for the CG2C data generation pipeline described in Section 5.1 and Section 5.2. The number of synthetic data we generated from each source is shown in Table 2.

Dataset	CG2C-MHQA		CG2C-Doc
	HotpotQA	Musique	
Positive	6734	5385	1034
Negative	1479	1048	1034
Total	8213	6433	2068

Table 2: **CG2C Dataset.** Number of synthetic data generated from different sources.

Training Details. In this section, we provide the training and hyperparameter details. We employed standard cross-entropy loss to train all the models. Unless otherwise specified, we utilized the default parameters of the backbone models.

For **FactCG-RBT**, following Tang et al.(2024a), we begin by fine-tuning the RoBERTa-large from AlignScore (Zha et al., 2023). We load both the base model and the binary classification layer. For the first stage, we used a batch size of 2, an accumulated step of 8, and a learning-rate of 1e-5 to fine-tune the model for 1 epoch on the MiniCheck's C2D data along with our CG2C-MHQA data.

For **FactCG-DBT** and **FactCG-FT5**, we start from DeBERTa-v3-large[3] and Flan-T5-large[4] respectively. For the first stage, we used a batch size of 2 and an accumulated step of 8 to fine-tune the models for 1 epoch on MiniCheck's ANLI subset and C2D along with our CG2C-MHQA. We used learning rates of 1e-5 and 5e-5 respectively.

Same as MiniCheck, we then fine-tune these models for a second stage for 1 epoch on MiniCheck's D2C, along with our CG2C-Doc datasets. Each model's training settings remain the same as in the first stage.

Dynamic Threshold. We adjust the threshold for each LLM-AGGREFACT dataset to two decimal places, optimizing for the best performance on the validation set. We investigate how models behave when they are at their best performance on each dataset. Unlike (Tang et al., 2024a), we observe that building a unified threshold system leads to different results (shown in Appendix A.3), potentially underestimating the true potential of the model. We leave data calibration for a unified threshold as future work.

Evaluation Metrics. Following previous works, we evaluate performance with balanced accuracy (BAcc) (Brodersen et al., 2010; Laban et al., 2022). $\text{BAcc} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}}\right)$, where TP, TN,

[3] https://huggingface.co/microsoft/deberta-v3-large

[4] https://huggingface.co/google/flan-t5-large

Table 3: **LLM-AGGREFACT Results with Threshold Tuning.** Evaluation using BAcc (%). GPT results are added for reference. Highest score (except GPT) for each dataset is highlighted with dark green and second highest score highlighted with light green.

Model	AGGREFACT		TofuEval		WiCE	REVEAL	Claim	Fact	Expert	LFQA	RAG	AVG
	CNN	XSum	MediaS	MeetB			Verify	Check	QA		Truth	
GPT-4o-2024-05-13	68.1	76.8	71.4	79.8	78.5	86.5	69.0	77.5	59.6	83.6	84.3	75.9
SummaC-ZS	62.7	67.6	68.5	71.1	63.9	86.3	70.0	75.2	58.1	81.9	65.1	70.0
SummaC-Conv	67.7	64.6	68.4	69.2	65.2	85.2	71.3	74.2	57.8	79.0	65.1	69.8
AlignScore	63.5	69.4	71.8	71.8	66.2	85.3	72.9	76.6	59.2	86.0	74.3	72.5
AlignScore-DBT	65.4	72.6	70.2	76.1	69.9	86.4	74.1	76.9	57.8	83.7	74.3	73.4
MiniCheck-RBT	65.4	70.1	71.1	74.9	73.7	88.0	77.1	77.5	59.0	84.1	77.3	74.4
MiniCheck-DBT	63.7	74.6	68.9	70.6	76.5	87.4	75.6	75.9	58.8	84.2	78.9	74.1
MiniCheck-FT5	70.1	74.7	73.1	76.2	76.0	86.6	74.7	76.8	58.6	85.7	78.5	75.5
FactCG-RBT	70.6	72.2	71.2	75.3	72.1	89.9	78.2	76.7	58.7	86.6	78.1	75.4
FactCG-DBT	73.8	74.6	73.7	73.9	80.9	90.0	78.3	75.1	59.4	86.6	82.6	77.2
FactCG-FT5	69.8	73.8	73.7	76.5	77.8	89.6	79.3	77.6	59.1	87.7	79.3	76.7

Table 4: **Ablation Study on CG2C.** Evaluation using BAcc (%). Removing both CG2C-Doc and CG2C-MHQA results in the lowest performance. Removing either one while retaining the other leads to a smaller performance drop compared to removing both datasets. Consistent observations on three backbone models.

Model	AGGREFACT		TofuEval		WiCE	REVEAL	Claim	Fact	Expert	LFQA	RAG	AVG
	CNN	XSum	MediaS	MeetB			Verify	Check	QA		Truth	
FactCG-RBT												
– CG2C-Doc	-1.94 ↓	+0.05 ↑	-1.24 ↓	-1.10 ↓	-3.08 ↓	-0.01 ↓	-0.39 ↓	-0.23 ↓	+0.62 ↑	-1.00 ↓	+0.27 ↑	-0.73 ↓
– CG2C-MHQA	-4.72 ↓	+0.90 ↑	-0.34 ↓	-0.22 ↓	+0.46 ↑	-1.45 ↓	-1.41 ↓	+0.44 ↑	-0.22 ↓	+0.04 ↑	-1.17 ↓	-0.70 ↓
– CG2C-Both	-8.06 ↓	-0.96 ↓	-1.85 ↓	+0.10 ↑	-4.37 ↓	-2.29 ↓	-2.36 ↓	+1.35 ↑	+0.09 ↑	-0.78 ↓	-1.23 ↓	-1.85 ↓
FactCG-DBT												
– CG2C-Doc	-1.40↓	+1.14↑	+0.72↑	+2.13↑	-2.74↓	-3.08↓	-1.40↓	+0.12↑	-0.9↓	-0.16↓	-0.35↓	-0.54↓
– CG2C-MHQA	-1.95↓	+1.60↑	-1.33↓	+2.04↑	-4.58↓	-0.30↓	+0.46↑	+2.70↑	-0.35↓	+1.74↑	-1.02↓	-0.09↓
– CG2C-Both	-10.12↓	-0.05↓	-4.79↓	-3.32↓	-4.36↓	-2.67↓	-2.68↓	+0.85↑	-0.60↓	-2.40↓	-3.71↓	-3.08↓
FactCG-FT5												
– CG2C-Doc	-5.79 ↓	-1.91 ↓	-2.39 ↓	+0.86 ↑	+0.91 ↑	-0.66↓	-0.47 ↓	-0.89 ↓	+0.63 ↑	-1.06 ↓	-0.97 ↓	-1.07 ↓
– CG2C-MHQA	-1.67 ↓	-2.13 ↓	-1.88 ↓	-1.57 ↓	-1.82 ↓	-0.38 ↓	-0.99 ↓	+0.39 ↑	-0.07 ↓	+0.50 ↑	-0.31 ↓	-0.90 ↓
– CG2C-Both	-0.15 ↓	-3.99 ↓	-2.69 ↓	+0.47 ↑	-3.44 ↓	-2.40 ↓	-1.72 ↓	-0.15 ↓	+1.06 ↑	-0.52 ↓	-0.34 ↓	-1.26 ↓

FP, and FN stand for true positives, true negatives, false positives and false negatives.

Baseline Comparison. We compare FactCG models with recent state-of-the-art fact-checking models, SummaC-ZS, SummaC-Conv (Laban et al., 2022), AlignScore (Zha et al., 2023), MiniCheck (Tang et al., 2024a) with RoBERTa-large and DeBERTa-v3-large backbones. Since AlignScore was originally trained using the RoBERTa-large backbone, we additionally trained an AlignScore variant with the DeBERTa-v3-large backbone, referred to as AlignScore-DBT, to ensure a fair apples-to-apples comparison during benchmarking.

6.3 Evaluation on Benchmark

We show results on LLM-AGGREFACT in Table 3. Each experiment was ran only once due to its cost. Scores highlighted with dark green and light green are the best and second best performances for each dataset except GPT. We add GPT results for reference because both CG2C and MiniCheck use GPT to synthesize data. On average, our FactCGs outperform previous solutions with the same backbones. Our FactCG-DBT achieves the state-of-the-art, even outperforming GPT-4. We believe it is because our methods do not directly rely on LLMs to produce the label, so it is possible for FactCG to surpass the boundary established by LLMs. Our FactCG-DBT, although much smaller than its FT5 counterpart, reaches similar (or even higher) performance. It may be because RoBERTa is encoder-based and suitable for classification tasks in nature.

6.4 Ablation Study

As we achieve better performance than other state-of-the-art models with similar parameter size on LLM-AGGREFACT with a dynamic threshold, to better answer RQ3 and RQ4, we conduct an ablation study to see the performance difference when dropping CG2c-Doc or CG2C-MHQA or both. If both are dropped, we train stage 1 for 2 epochs same as MiniCheck (Tang et al., 2024a) to ensure better model convergence. Results are shown in Table 4. Dropping both CG2C-Doc and CG2C-

Table 5: **Ablation Study on Sub-graph Shapes.** CG2C without branches achieves higher BACC on AggreFact than those with branches.

Model	Context Sub-Graph Type	AVG
FactCG-RBT	CG2C with Branches	74.7
	CG2C without Branches	**75.4**
FactCG-DBT	CG2C with Branches	77.1
	CG2C without Branches	**77.2**
FactCG-FT5	CG2C with Branches	76.0
	CG2C without Branches	**76.7**

MHQA leads to lowest performances on each backbone model. In contrast, removing either one leads to a less performance drop comparing to remove both, which means adding either CG2C data can improve the model quality. This demonstrates the effectiveness of both synthetic generation approaches and answers RQ3 and RQ4 that both CG2C-MHQA and CG2C-Doc helps improving model performance.

In addition, as shown in Table 5, we observe that leveraging multi-hop data generated from context sub-graphs with branches performs worse than generate all data without branches. This is potentially caused by they can be further decomposed into multiple sub-graph without branches. Hence converting claims to simpler sub-claims with less hops and reduced reasoning complexity.

7 Evaluation on Connected Reasoning

As QA models suffer from disconnected reasoning, they leverage exploited dataset artifacts to produce correct answers instead of connecting information across multi-hops. We conduct a similar connected reasoning (CoRe) analysis on the multi-hop reasoning chain of trained grounded factuality models similar to Trivedi et al. (2020). We would like to answer the research question:

RQ5: Does FactCG has more connected reasoning than models not trained on CG2C datasets?

7.1 CoRe Dataset Construction

We leverage WiCE (Kamoi et al., 2023) in the LLM-AggreFact benchmark as it provides the document-claim-evidence triples in the forms of $(doc_i, c_i, \mathcal{E}_i)$ where $\mathcal{E}_i = \{E_{i1}, E_{i2}, .., E_{ij}, ...\}$. Each evidence E_{ij} is a unique minimum independent sentence set $E_{ij} \subset doc_i$ sufficient to make a grounded factuality judgement. $E_{ij} = \{e_{ij}^1, e_{ij}^2, ..., e_{ij}^k, ...\}$. E_{ij} may share the same evidence sentences. We define E_i^* as the union of all evidence sets such that $E_{ij} \subset E_i^*$.

We filter positive cases in the WiCE validation and test sets that contain more than one evidence sentence in each E_{ij}. We treat these as the positive set that requires connected reasoning across sentences with sufficient context.

To synthetically construct negative cases, for each positive case, we randomly select evidence sentences e_i^- in E_i^* and remove their appearance in the grounding document such that each E_{ij} has one evidence removed. In this case, the claim cannot be verified with the corrupted document through any E_{ij}. We treat this negative set as minimum disconnected reasoning set with insufficient context.

For each c_i we have a pair of positive and negative data $\langle doc_i, c_i \rangle$ and $\langle doc_i \backslash e_i^-, c_i \rangle$, where $i \in I$ and $|I|$ is the total size of pairs in the constructed dataset. For a grounded factuality model $\mathcal{M}$, we define the accuracy via connected reasoning as:

$$\text{Accuracy}_{\text{CoRe}} = \frac{\sum_i (\mathcal{M}(doc_i, c_i) \geq \theta) \cap (\mathcal{M}(doc_i \backslash e_i^-, c_i) < \theta)}{|I|} \quad (2)$$

We also define the proportion of correct predictions that were made via connected reasoning as:

$$\text{Precision}_{\text{CoRe}} = \sum_i \frac{(\mathcal{M}(doc_i, c_i) \geq \theta) \cap (\mathcal{M}(doc_i \backslash e_i^-, c_i) < \theta)}{(\mathcal{M}(doc_i, c_i) \geq \theta)} \quad (3)$$

7.2 Experiment Setting

We run AlignScore, MiniCheck models along with FactCG trained on the same backbone models for comparison. We select the same threshold used for benchmark **LLM-AggreFact** evaluation on the WiCE dataset for each model.

7.3 Results and Analysis

We show the CoRe evaluation results in Table 6. We observe consistent performance across the RBT, DBT and FT5 backbones. For RBT, FactCG achieves superior $\text{Precision}_{\text{CoRe}}$ while maintaining comparable $\text{Accuracy}_{\text{CoRe}}$ relative to MiniCheck. In the cases of FT5 and DBT, FactCG outperforms MiniCheck in terms of both $\text{Accuracy}_{\text{CoRe}}$ and $\text{Precision}_{\text{CoRe}}$. It shows that FactCG leverages more on the effective multi-hop reasoning chain across multi-hops than dataset artifacts. However, we still observe that half of the reasoning can be categorized as disconnected reasoning on both models. It demonstrates room for improvement in future trustworthy factuality models' reasoning process.

Model	Accuracy$_{\text{CoRe}}$	Precision$_{\text{CoRe}}$
AlignScore	13.85	18.22
Minicheck-RBT	**24.42**	34.32
FactCG-RBT	**24.42**	**54.04**
Minicheck-DBT	26.54	37.81
FactCG-DBT	**33.85**	**44.78**
Minicheck-FT5	28.85	37.04
FactCG-FT5	**32.31**	**45.53**

Table 6: **CoRe Evaluation Results.** Definition of two metircs can be found in section 7.1.

8 Conclusion

In this paper, we investigated the difference between state-of-the-art synthetic generated claims and real LLM-generated claims. To fill the gap, we proposed a new synthetic data generation approach, CG2C, that leverages the context graph to generate complex multi-hop claims without relying on LLMs to decide data labels. Our Fact Checker FactCG leveraging this generated data achieves state-of-the-art performance compared with models of similar parameter size and even outperforms GPT-4-o, which we used to construct the CG2C dataset.

9 Limitations

9.1 Graph Extraction Quality

We find it hard to define the granularity of the extracted triples. The simplest solution is to extract the $\langle noun, verb, noun \rangle$ triples from sentences, but information after prepositions and conjunctions may be lost. Long sentences and clauses make the task even more complicated. On the other hand, if we extract too many details, the graph will contain a lot of unnecessary edges and become too large to process efficiently. It is not easy to obtain optimal graphs for our use cases. Future research is needed to improve the graph extraction process. In addition, although the graph extraction method can extract the relations between entities and help downstream models make inference, the extraction process itself cannot handle inference within the source document perfectly. For example, when we extract sub-graph for claims from the source documents' graphs in Section 4, we may observe the following two relations from the source document: "Shop A is open on Monday" and "Shop A is open on Tuesday". However, they cannot be matched exactly with "Shop A is open two days a week" in the claim. In this scenario, we have to rely on LLMs' inference ability to resolve the co-reference.

9.2 Data Generation Quantity

Although our CG2C could support more scalable generation, we mainly extract few samples from one sub-graph to avoid duplication. Exploiting the graph to generate more samples and controlling the ratio of generated samples' hops could be a non-trivial topic to explore.

9.3 Chunking Methodology

We follow previous work (Zha et al., 2023; Tang et al., 2024a) for the chunking methodology and fix the model input to a similar input token length. However, each scenario may have its own optimal chunk size. An area worth exploration for improving generalizable hallucination detection classifiers is developing intelligent methods to segment long documents into chunks based on topic flow.

References

Amazon. 2024. Guardrails for amazon bedrock.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. 2010. The balanced accuracy and its posterior distribution. *2010 20th International Conference on Pattern Recognition*, pages 3121–3124.

Jian Chen, Peilin Zhou, Yining Hua, Loh Xin, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024. FinTextQA: A dataset for long-form financial question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6047, Bangkok, Thailand. Association for Computational Linguistics.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4026–4032.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Google. 2024. Check grounding with rag.

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*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4615–4634, Bangkok, Thailand. Association for Computational Linguistics.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Deren Lei, Yaxi Li, Mengya Hu, Mingyu Wang, Vincent Yun, Emily Ching, and Eslam Kamal. 2023. Chain of natural language inference for reducing large language model ungrounded hallucinations. *ArXiv*, abs/2310.03951.

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

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023a. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024b. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480, Mexico City, Mexico. Association for Computational Linguistics.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models. *arXiv preprint arXiv:2305.04676*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863, Online. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2023. Factcheck-gpt: End-to-end fine-grained document-level fact-checking and correction of llm output. *ArXiv*, abs/2311.09000.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. 2023. Exploring large language models for knowledge graph completion. *arXiv preprint arXiv:2308.13916*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen Mckeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 11:39–57.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

A Appendix

A.1 Benchmark Statistics

The statistics of **LLM-AGGREFACT** benchmark test set can be found in table 7. More details can be found in Tang et al. (2024a).

Dataset	Size
AggreFact-CNN	558
AggreFact-XSum	558
TofuEval-MediaS	726
TofuEval-MeetB	772
Wice	358
Reveal	1710
ClaimVerify	1088
FactCheck-GPT	1566
ExpertQA	3702
Lfqa	1911
RAGTruth	16371

Table 7: **LLM-AGGREFACT benchmark test set size.**

A.2 T-Test Analysis

We conduct a paired bootstrap test with 100 runs and a p-value < 0.05. The bootstrap sample size is set to 150. Our experiment aims to reject the null hypothesis that the observed differences between methods are due to random chance.

We first compare our FactCG-DBT with MiniCheck-DBT on synthetic data generation as they share the same model backbone. We set H_0 as the performance on BAcc of FactCG-DBT is no better than that of MiniCheck-DBT. We show results in Table 8.

On average, FactCG-DBT demonstrates a significant improvement in performance (BAcc) compared to MiniCheck-DBT with p-value less than 0.05. Therefore we reject the null hypothesis that FactCG-DBT performs no better than MiniCheck-DBT. Notably, in datasets where FactCG-DBT exhibits a relative improvement, the p-values are consistently below 0.05 and align with the observed BAcc gains. This demonstrates the effectiveness of our synthetic training data.

We additionally compare FactCG-DBT with MiniCheck-FT5, which is the best-performing fact-checker (<0.8B) on LLM-AggreFact, to prove the significance of our improvement. We set H_0 as the performance on BAcc of FactCG-DBT is no better than that of MiniCheck-FT5. We show results in Table 9.

Similar to the above experiment, the averaged performance shows a statistically significant improvement, providing strong evidence to reject the null hypothesis that FactCG-DBT performs no better than MiniCheck-FT5. In datasets where FactCG-DBT demonstrates improvement, the p-values are consistently below 0.05, confirming the statistical significance of these gains.

In conclusion, the results of both t-tests support our assertion of the statistically significant improvement of FactCG-DBT and align with the BAcc gains in Table 3.

A.3 Results without Threshold Tuning

We use threshold 0.5 for our FactCG model. We shared our results without specific task threshold tuning on Table 10. Our FactCG-DBT achieves the highest balanced accuracy compared to MiniCheck.

A.4 Model Size and Computational Cost

We conduct our experiment on Nvidia Quadro RTX 8000 and train all FactCG model backbones with 4 GPUs and inference with 1 GPU. Training costs and the inference costs (including threshold selection) are shown below:

For our 0.4B size FactCG-RBT, the training takes approximately 7 minutes and the inference takes about about 65 minutes to complete.

For our 0.4B size FactCG-DBT, the training costs approximately 18 minutes and the inference takes about 103 minutes to complete.

For our 0.8B size FactCG-T5, the training takes approximately 32 minutes and the inference takes about 238 minutes to complete.

A.5 Scientific Artifacts

We list the licenses used in this paper: pytorch-lightning (Apache License 2.0), PyTorch (BSD-3), Huggingface Transformers (Apache License 2.0), MiniCheck (Apache License Version 2.0), AlignScore (MIT License), RoBERTa-Large-MNLI (MIT License), hotpotqa (Apache License 2.0), musique (Creative Commons Attribution 4.0 International), OpenAI (Term of use[5]). We adhere to the intended use of all the existing artifacts mentioned in this paper.

A.6 Packages

We employed the following package to conduct our experiment: pytorch-lightning v2.4.0, PyTorch

[5] https://openai.com/policies/row-terms-of-use/

Measures	AGGREFACT		TofuEval		WiCE	REVEAL	Claim Verify	Fact Check	Expert QA	LFQA	RAG Truth	AVG
	CNN	XSum	MediaS	MeetB								
Relative Improvement	10.1	0.0	4.8	3.3	4.4	2.6	2.7	-0.8	0.6	2.4	3.7	3.1
p-value	3.9e-23	1.0e-01	2.0e-13	7.1e-08	3.1e-15	1.4e-12	9.0e-10	9.9e-01	4.5e-02	1.2e-08	2.3e-07	4.43e-40
p-value < 0.05	True	False	True	True	True	True	True	False	True	True	True	True

Table 8: **T-test Comparison for FactCG-DBT and MiniCheck-DBT.** An apples-to-apples comparison with the same DBT backbone. It allows us to validate the significance of the improvement presented in Table 3 and assess the effectiveness of our synthetic training data. Relative Improvement is defined as the BAcc (%) delta between FactCG-DBT and MiniCheck-DBT.

Measures	AGGREFACT		TofuEval		WiCE	REVEAL	Claim Verify	Fact Check	Expert QA	LFQA	RAG Truth	AVG
	CNN	XSum	MediaS	MeetB								
Relative Improvement	3.7	-0.1	0.6	-2.3	4.9	3.4	3.6	-1.7	0.8	0.9	4.1	1.7
p-value	0.6e-03	2.6e-01	3.6e-03	9.9e-01	9.7e-18	4.6e-15	5.9e-11	1.0e-05	4.4e-03	3.2e-03	9.5e-08	8.0e-14
p-value < 0.05	True	False	True	False	True	True	True	False	True	True	True	True

Table 9: **T-test Comparison for FactCG-DBT and MiniCheck-FT5.** We validate the significance of the improvement for FactCG-DBT over the best performing fact-checker on LLM-Aggrefact. Relative Improvement is defined as the BAcc (%) delta, representing the difference between FactCG-DBT and MiniCheck-FT5.

v2.3.0+cu121, and transformers v4.44.2.

A.7 FactCG-DBT and FactCG-RBT Instruction Template

We show FactCG-DBT and FactCG-RBT Instruction Template in Table 11

A.8 Prompts and GPT Usage

We use GPT-4-o to create CG2C datasets. For each positive and negative sample pair generated from HotpotQA, we run four GPT calls in order to generate a claim (Table 15), a context graph (Table 12), a context sub-graph (Table 14), and a negative sample's document (Table 16). For each positive sample in CG2C-MHQA generated from Musique, we run one GPT call to generate a claim (Table 15). To construct CG2C-Doc dataset, we run one GPT call for each unique document (around 1300) in MiniCheck's D2C dataset to extract the context graph (Table 13). In addition, for each positive and negative sample pair we generated, we call GPT two more times to generate the claim (Table 17) and negative sample's document (Table 16).

Table 10: **LLM-AGGREFACT Results without threshold tuning.**

Model	**AGGREFACT**		**TofuEval**		**WiCE**	**REVEAL**	**Claim**	**Fact**	**Expert**	**LFQA**	**RAG**	**AVG**
	CNN	**XSum**	**MediaS**	**MeetB**			**Verify**	**Check**	**QA**		**Truth**	
GPT-4o-2024-05-13	68.1	76.8	71.4	79.8	78.5	86.5	69.0	77.5	59.6	83.6	84.3	75.9
Summac-CV	65.2	54.5	63.7	62.8	54.3	67.7	70.9	53.4	54.9	62.1	61.7	61.0
Summac-ZS	51.1	61.5	69.5	71.0	62.8	85.3	69.7	75.2	55.2	77.6	65.6	67.7
AlignScore	52.4	71.4	69.2	72.6	66.0	85.3	69.6	74.3	58.3	84.5	71.7	70.5
MiniCheck-RBT	63.7	70.8	71.9	75.9	67.6	88.8	77.4	73.3	57.4	84.4	77.2	73.5
MiniCheck-DBT	64.2	71.0	69.3	72.7	69.4	87.3	75.6	73.0	58.9	83.9	78.8	73.1
MiniCheck-FT5	69.9	74.3	73.6	77.3	72.2	86.2	74.6	74.7	59.0	85.2	78.0	75.0
FactCG-RBT	70.1	67.0	70.0	73.4	64.9	89.7	77.8	72.4	57.7	86.5	78.2	73.4
FactCG-DBT	70.2	73.9	72.3	74.3	74.2	88.4	78.5	72.1	59.1	86.7	82.3	75.6
FactCG-FT5	62.8	73.2	74.3	75.5	72.8	89.7	78.6	73.0	58.3	88.2	78.9	75.0

Table 11: FactCG-DBT and FactCG-RBT Instruction Template

<DOCUMENT>.

Choose your answer: based on the paragraph above can we conclude that "<CLAIM>"?

OPTIONS:
- Yes
- No
I think the answer is

Table 12: Prompt: Context Graph Extraction for CG2C-MHQA

You are a helpful assistant.

Extract content graph with sentences in forms of triples (entity, relation, entity) based only on the provided sentences. Provide the triples only.

Examples:

Provided Sentences:

- Scott Derrickson (born July 16, 1966) is an American director, screenwriter and producer.
- Edward Davis Wood Jr. (October 10, 1924 – December 10, 1978) was an American filmmaker, actor, writer, producer, and director.

Triples in Provided Sentences:

- (Scott Derrickson, born on, July 16, 1966)
- (Scott Derrickson, is, American)
- (Scott Derrickson, is, director)
- (Scott Derrickson, is, screenwriter)
- (Scott Derrickson, is, producer)
- (Edward Davis Wood Jr., born on, October 10, 1924)
- (Edward Davis Wood Jr., died on, December 10, 1978)
- (Edward Davis Wood Jr., was, American)
- (Edward Davis Wood Jr., was, filmmaker)
- (Edward Davis Wood Jr., was, actor)
- (Edward Davis Wood Jr., was, writer)
- (Edward Davis Wood Jr., was, producer)
- (Edward Davis Wood Jr., was, director)

Your turn:

Provided Sentences:
<SENTENCES>

Triples in Provided Sentences:

Table 13: Prompt: Context Graph Extraction for CG2C-Doc

You are a helpful assistant.

Given a news article, go over every sentence and extract triples in forms of (entity {tuple_delimiter} entity {tuple_delimiter} a short description about the relation between two entities). Group the triples with the same entity. Separate groups of triples using {group_delimiter}.

Examples:

Provided Sentences:
Hunt also said the government needs to reform the welfare system to get more people back to work. The number of people not in the workforce for physical or mental health reasons has soared since the pandemic. Ken Clarke, a former Conservative Treasury chief, said cutting inheritance tax "might appeal to the Conservative right, but it leaves them open to the most appalling criticisms when inflation and the state of affairs is making poorer people in this country very vulnerable indeed." "I'm not sure that the economic and financial state of the country justifies it."

Groups of Triples in Provided Sentences:

- Hunt {tuple_delimiter} government {tuple_delimiter} Hunt said something about the government.
- government {tuple_delimiter} welfare system {tuple_delimiter} government need to reform welfare system.
- welfare system {tuple_delimiter} people {tuple_delimiter} reformed welfare system can get people back to work.
- people {tuple_delimiter} physical or mental health reasons {tuple_delimiter} people has soared due to physical or mental health reasons
- {group_delimiter}
- Ken Clarke {tuple_delimiter} former Conservative Treasury chief {tuple_delimiter} Ken Clarke is former Conservative Treasury chief
- Ken Clarke {tuple_delimiter} cutting inheritance tax {tuple_delimiter} Ken Clarke said something about cutting inheritance tax
- cutting inheritance tax {tuple_delimiter} Conservative right {tuple_delimiter} cutting inheritance tax appeal to the Conservative right
- cutting inheritance tax {tuple_delimiter} criticisms {tuple_delimiter} cutting inheritance tax leaves open to criticisms
- cutting inheritance tax {tuple_delimiter} inflation and the state of affairs {tuple_delimiter} cutting inheritance tax cause inflation and the state of affairs
- inflation and the state of affairs {tuple_delimiter} poorer people {tuple_delimiter} inflation and the state of affairs make poorer people vulnerable
- economic and financial state of the country {tuple_delimiter} cutting inheritance tax {tuple_delimiter} economic and financial state of the country might not justify cutting inheritance tax.

Your turn:

Provided Sentences:
<SENTENCES>

Groups of Triples in Provided Sentences:

Table 14: Prompt: Multi-hop Reasoning Chain Mapping for Claim Sub-graph Extraction

You are a helpful assistant.

Find the triplets in forms of (entity {tuple_delimiter} entity {tuple_delimiter} a short description about the relation between two entities) that exist in the provided sentence with the provided triplets only. Directly output the Existed Triplets in Provided Sentences.

Examples:
Provided Triples:

- (Scott Derrickson {tuple_delimiter} July 16, 1966 {tuple_delimiter} Scott Derrickson was born on July 16, 1966)
- (Scott Derrickson {tuple_delimiter} American {tuple_delimiter} Scott Derrickson is an American)
- (Scott Derrickson {tuple_delimiter} director {tuple_delimiter} Scott Derrickson is a director)
- (Scott Derrickson {tuple_delimiter} screenwriter {tuple_delimiter} Scott Derrickson is a screenwriter)
- (Scott Derrickson {tuple_delimiter} producer {tuple_delimiter} Scott Derrickson is a producer)
- (Edward Davis Wood Jr. {tuple_delimiter} October 10, 1924 {tuple_delimiter} Edward Davis Wood Jr. was born on October 10, 1924)
- (Edward Davis Wood Jr. {tuple_delimiter} December 10, 1978 {tuple_delimiter} Edward Davis Wood Jr. was died on December 10, 1978)
- (Edward Davis Wood Jr. {tuple_delimiter} American {tuple_delimiter} Edward Davis Wood Jr. was an American)
- (Edward Davis Wood Jr. {tuple_delimiter} filmmaker {tuple_delimiter} Edward Davis Wood Jr. was a filmmaker)
- (Edward Davis Wood Jr. {tuple_delimiter} actor {tuple_delimiter} Edward Davis Wood Jr. was an actor)
- (Edward Davis Wood Jr. {tuple_delimiter} writer {tuple_delimiter} Edward Davis Wood Jr. was a writer)

Provided Sentences:
Scott Derrickson and Ed Wood were both of the same nationality.

Existed Triplets in Provided Sentences:

- (Scott Derrickson {tuple_delimiter} American {tuple_delimiter} Scott Derrickson is an American)
- (Edward Davis Wood Jr. {tuple_delimiter} American {tuple_delimiter} Edward Davis Wood Jr. was an American)

Your turn:

Provided Triples:
<TRIPLES>

Provided Sentences:
<SENTENCES>

Existed Triplets in Provided Sentences:

Table 15: QA to Declarative Statement Prompt

You are a helpful assistant.

Question: <QUESTION>

Answer: <ANSWER>

Convert the question answer pair into a single declarative sentence.

Single declarative sentence:

Table 16: Prompt: Document Relation Removal

You are a helpful assistant.

Remove the relation between two provided entities in below provided sentences with minimal changes. Provide the rewrite sentences only.

Examples:

Provided Entities:
India, Area

Provided Sentences:
Indogrammodes is a genus of moths of the Crambidae family. It contains only one species, Indogrammodes pectinicornalis, which is found in India. India, officially the Republic of India ("Bhārat Gaṇarājya"), is a country in South Asia. It is the seventh-largest country by area, the second-most populous country (with over 1.2 billion people), and the most populous democracy in the world.

Rewrite Sentences with relationship between provided two entites removed:
Indogrammodes is a genus of moths of the Crambidae family. It contains only one species, Indogrammodes pectinicornalis, which is found in India. India, officially the Republic of India ("Bhārat Gaṇarājya"), is a country in South Asia. It is the second-most populous country (with over 1.2 billion people) and the most populous democracy in the world.

Your turn:

Provided Entities:
<ENTITIES>

Provided Sentences:
<SENTENCES>

Rewrite Sentences with Relationship Between Provided Two Entites Removed:

Table 17: Prompt: Claim Generation from Graph

You are a helpful assistant.

Given a news article, write a short sentence, which must include the following entities: <ENTITIES>

Provided Article:
<SENTENCES>

Your Claim: