DecMetrics: Structured Claim Decomposition Scoring for Factually Consistent LLM Outputs

Anonymous ACL submission

Abstract

Claim decomposition plays a crucial role in the fact-checking process by breaking down complex claims into simpler atomic components and identifying their unfactual elements. Despite its importance, current research primarily focuses on generative methods for decomposition, with insufficient emphasis on evaluating the quality of these decomposed atomic claims. To bridge this gap, we introduce **DecMetrics**, which comprises three new metrics: `COMPLETENESS`, `CORRECTNESS`, and `SEMANTIC ENTROPY`, designed to automatically assess the quality of claims produced by decomposition models. Utilizing these metrics, we develop a lightweight claim decomposition model, optimizing its performance through the integration of these metrics as a reward function. Through automatic evaluation, our approach aims to set a benchmark for claim decomposition, enhancing both the reliability and effectiveness of fact-checking systems.[1]

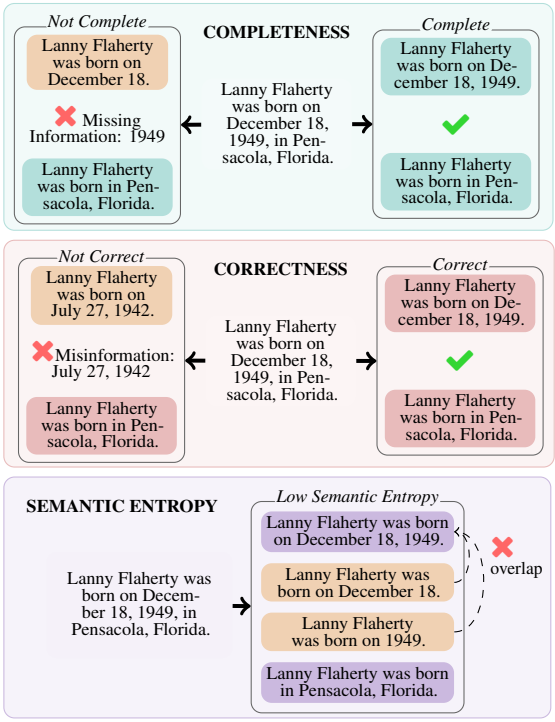

Figure 1: Characteristics of Claim Decomposition: *High Completeness, High Correctness, and High Semantic Entropy.* The process involves breaking down long-form text into atomic claims. High-quality decompositions are indicated by green, red, and purple boxes, whereas orange boxes denote claims that may be deficient in essential information, introduce misinformation, or exhibit redundant semantic overlap.

1 Introduction

The utilization of long-form text generated by large language models (LLMs) has become increasingly common. However, these texts often contain factual inaccuracies, necessitating the development of methodologies to evaluate their factuality. Existing fact-checking approaches, such as SAFE (Wei et al., 2024), FActScore (Min et al., 2023), and FacTool (Chern et al., 2023), conceptualize generated text as claims and employ claim decomposition as a foundational component of the fact-checking process.

Claim decomposition entails breaking down a claim, whether it be an entire text or a segment, into individual atomic claims (Wanner et al., 2024; Hu et al., 2024). Terms such as "atomic fact" and "atomic proposition" are frequently used interchangeably with atomic claims. The traditional fact-checking procedure involves decomposing generated text into atomic claims, verifying each claim's factual accuracy against external knowledge sources, and then compiling these results to determine the text's overall factuality (Iqbal et al., 2024; Li et al., 2024). Despite this, existing methodologies often overlook the variability in claim decomposition quality.

To illustrate, if two models decompose the same

[1] Our anonymous code and dataset available at: https://anonymous.4open.science/r/DecMetrics-64BE

claim into different numbers of atomic claims, the resulting fact scores will differ. Additionally, if decomposition models alter the claim during the decomposition process, the atomic claims may be factual but fail to accurately represent the original claim. As depicted in the second sub-figure in Figure 1, the claim within the orange box is not factual. However, if decomposition models transform it to state "Lanny Flaherty was born on December 18, 1949", the fact score would misleadingly reflect a high accuracy, misrepresenting the true factuality of the LLM's response. Given the critical role of claim decomposition in the fact-checking process, which affects the number and scope of evaluated claims, the results and metrics are inherently influenced by the decomposition method (Hu et al., 2024). Therefore, it is essential to conduct a comprehensive evaluation of decomposed claims to ensure accuracy and reliability.

DecMetrics: Measures for Claim Decomposition Quality

To enhance the fact-checking process's accuracy, it is important to accurately decompose claims into atomic claims. Thus, defining a high-quality atomic claim becomes critical. We propose that an ideal atomic claim should exhibit the following characteristics, as demonstrated in Figure 1: high completeness (Kamoi et al., 2023), high correctness (Wanner et al., 2024), and high semantic entropy (Farquhar et al., 2024).

- **High Completeness**: The decomposed claims should encompass all necessary facets of the original text.
- **High Correctness**: Each atomic claim must be factually correct, substantiated by the original text.
- **High Semantic Entropy**: Atomic claims should not exhibit repetitive paraphrasing or semantic overlap; instead, they should be independent and distinct, ensuring high semantic entropy.

Leveraging these characteristics, we introduce three metrics: `COMPLETENESS`, `CORRECTNESS`, and `SEMANTIC ENTROPY`, to automate the evaluation of atomic claims.

DecModel: A Standardized Claim Decomposition Model

Building on our established metrics, we propose the development of a claim decomposition model aimed at generating atomic claims that align with these quality standards. We specifically employ a reinforcement learning framework, as outlined in prior studies (Schulman et al., 2017; Rafailov et al., 2024), using these metrics as reward signals to enhance the generation of high-quality atomic claims.

Claim2Atom: A Comprehensive Claim Decomposition Benchmark

In parallel, we propose the creation of **Claim2Atom**, a comprehensive evaluation benchmark for structuring claims into atomic components. This benchmark aggregates and filters existing public datasets, FActScore (Min et al., 2023) and WICE (Kamoi et al., 2023), along with our newly curated decomposition dataset, DecData. The benchmark is explicitly designed to facilitate rigorous evaluations across several dimensions. It enables a critical assessment of the capabilities of various LLMs, provides a robust framework for evaluating the effectiveness of our specialized decomposition models, and serves as a foundational tool for future research in the field of claim decomposition.

2 Related Works

Many existing works employ a pipeline approach to decompose text into atomic claims for evaluating the factuality of long-form text. Recent advancements in claim decomposition mainly rely on prompted large language model (LLM)-based methods, often incorporating in-context example decompositions (Min et al., 2023; Kamoi et al., 2023; Chern et al., 2023; Wei et al., 2024; Iqbal et al., 2024). To leverage propositions, akin to atomic claims, as retrieval units, (Chen et al., 2024) developed a Propositionizer that decomposes text into simple propositions using data generated by GPT-4. (Song et al., 2024) introduced VeriScore to check whether each decomposed atomic claims are verifiable. (Gunjal and Durrett, 2024) decontext molecular facts to make each decomposed atomic claims can be self-contained. However, there has been limited attention to fully evaluate the quality of these decomposed atomic claims. (Wanner et al., 2024) highlighted that downstream fact-checking methods are sensitive to the decomposition approach and proposed ClaimScore to assess the number of supported atomic claims as a measure of decomposition quality. Nonetheless, this approach overlooks the semantic overlap between atomic

claims and does not assess the completeness of the decomposed claims. If a language model repeatedly paraphrases factual claims, it may achieve a very high ClaimScore. Consequently, we plan to define metrics for automatically evaluating the quality of decomposed atomic claims and use these metrics as rewards to train a reinforcement-learning-based claim decomposition model. Our aim is to generate atomic claims that are complete (Kamoi et al., 2023), correct (Wanner et al., 2024), and exhibit high semantic entropy (Farquhar et al., 2024).

3 DecMetrics: Evaluation Metrics for Claim Decomposition Quality

We treat a text as a claim, and then decompose it into atomic claims using a claim decomposition model. Let $d(c) = \{ac_1, \ldots, ac_n\}$ denote the automatic decomposition of a claim c into atomic claims via the decomposition model $d(\cdot)$.

3.1 DecMetrics Overview

DecMetrics comprises three evaluation metrics: `COMPLETENESS`, `CORRECTNESS`, and `SEMANTIC ENTROPY`, each framed as an entailment classification task. Following prior work (Kamoi et al., 2023; Tang et al., 2024), we annotate entailment labels as *supported* or *unsupported*.

3.1.1 COMPLETENESS Metric

This metric evaluates whether the decomposed atomic claims $\{ac_1, \ldots, ac_n\}$ collectively cover all necessary aspects of the original claim c. Unlike traditional NLI tasks (Laban et al., 2022), which classify claims as *supported* or *unsupported* in a binary manner, `COMPLETENESS` measures fine-grained factuality by computing the entailment probability between the merged atomic claims and the original claim. Formally:

$$\text{cp}(c, ac) = P(y = \textit{supported} \mid c, ac),$$

where $P(y = \textit{supported} \mid c, ac)$ is derived using a natural language inference (NLI) model. The granularity of this metric is at the *claim-claim level*.

3.1.2 CORRECTNESS Metric

This metric evaluates whether each decomposed atomic claim ac_i is factually faithful to the original claim c, ensuring no fabrication or hallucination. Formally, `CORRECTNESS` computes the fraction of atomic claims entailed by c:

$$\text{cr}(c, ac) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\big(\text{NLI}(c, ac_i) = \textit{supported}\big),$$

where $\mathbb{I}(\cdot)$ is the indicator function (1 if ac_i is entailed by c, 0 otherwise) and $\text{NLI}(c, ac_i)$ denotes the entailment score from a natural language inference model. A high `CORRECTNESS` score indicates minimal factual deviation during decomposition.

3.1.3 SEMANTIC ENTROPY Metric

To assess redundancy among atomic claims, we cluster them using an NLI model: if $\text{NLI}(ac_i, ac_j) = \textit{supported}$ or vice versa, ac_i and ac_j belong to the same cluster C. The `SEMANTIC ENTROPY` metric, operating at the *atomic claim-atomic claim level*, quantifies the diversity of atomic claims via:

$$\text{se}(\{ac_1, \ldots, ac_n\}) = -\sum_{C} P(C) \log P(C),$$

where $P(C)$ is the probability of a decomposed atomic claim ac_i belonging to cluster C. This encourages non-redundant (low clustering) and highly atomic (numerous) decompositions.

These metrics function similarly to recall, precision, and accuracy in their capacity to reveal the strengths and weaknesses of a model. High scores across all three metrics are required for a comprehensive assessment of the model's overall performance.

For instance, if a claim is notably long, `SEMANTIC ENTROPY` may increase, but may exist potential incomplete decomposition. In such cases, `COMPLETENESS` can reveal decomposition completion. Conversely, a decomposition model returning the original claim will score high on `COMPLETENESS` and `CORRECTNESS`, but zero on `SEMANTIC ENTROPY`, exposing shortcomings. Similarly, arbitrary atomic claims unrelated to the original claim might result in high `SEMANTIC ENTROPY`, while `COMPLETENESS` and `CORRECTNESS` will highlight discrepancies.

While these metrics leverage existing NLI models (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020; Liu et al., 2022), fine-tuning on task-specific data is essential for optimal performance.

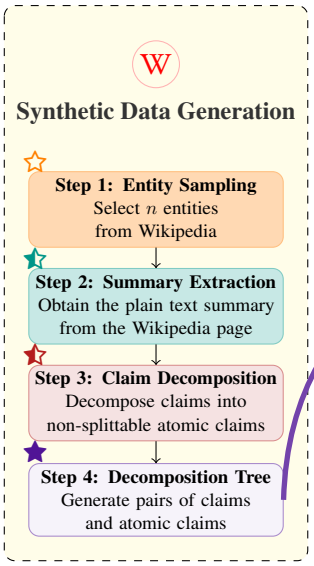

Figure 2: **Synthetic Data Generation Process**. The sequential process includes topic sampling from Wikipedia, extracting textual summaries as original claims, decomposing claims into atomic claims while adjusting granularity, and constructing a decomposition tree that pairs claims with atomic claims.

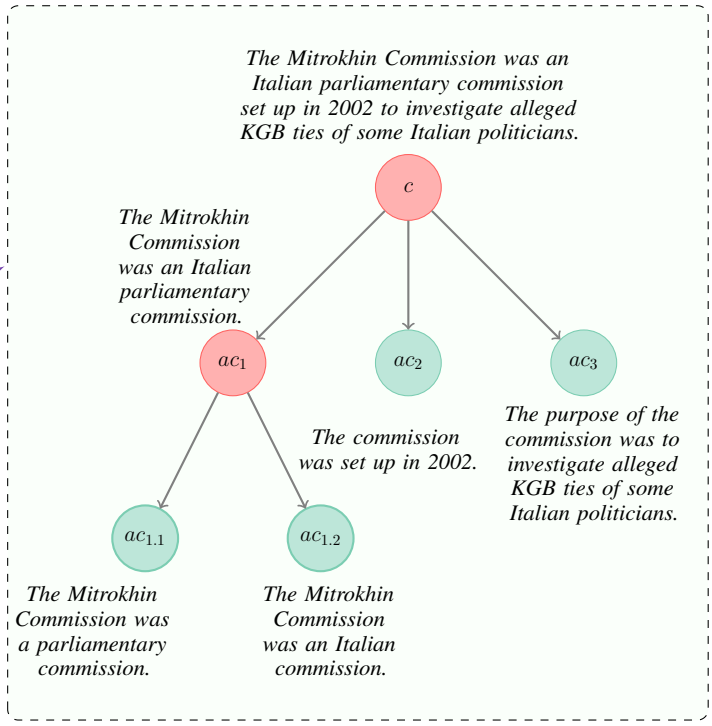

Figure 3: **Synthetic Data Visualation**. Decomposition tree with claims and atomic claims.

3.2 Synthetic Data Generation

To facilitate the calculation of evaluation metrics, it is important to develop models that rely on synthetic training data for effective training. The process of generating synthetic data, depicted in Figure 2, involves several key steps:

Step 1: Entity Sampling from Wikipedia The first step involves selecting entities from Wikipedia, which forms the basis for further processes. Specifically, we sample 200 entities from Wikipedia, spanning from personal entities like 'Floyd D. Rose' to event-related entities such as the '2018 NBA Awards'.

Step 2: Extracting Wikipedia Page Summaries During this phase, we obtain the plain text summary from the Wikipedia page corresponding to each entity using the Python library `wikipedia`[2]. We exclude entities that do not produce satisfactory summary text. After this filtration, 183 entities are retained for subsequent processing.

Step 3: Claim Decomposition The Wikipedia summaries are used as potential claims, which are then decomposed using the large language model (LLM) guided by the claim decomposition prompt described in Appendix Figure 6. A critical aspect is ensuring claims are non-splittable; claims are iteratively split if the number of decomposed atomic claims exceeds one, as detailed in Appendix Algorithm 1. Entities with Wikipedia summaries that cannot be decomposed into non-splittable claims after 10 iterations are filtered out. After this filtration process, 180 entities are retained for subsequent processing.

Step 4: Decomposition Tree Generation This step involves the creation of a decomposition tree, as illustrated in Figure 3. The tree displays claims and their corresponding atomic claims at varying levels of granularity. From this decomposition tree, we extract all subtrees, which serve as the foundation for generating synthetic training data. After extracting the subtrees, we conduct a verification process to ensure each subtree meets specific criteria. This process involves a reverse check using a LLM, guided by the atomic claim checking prompt detailed in Appendix Figure 7. This process verifies that the decomposed atomic claims adhere to our criteria. Given that the forward decomposition guarantees that atomic claims are non-splittable, the reverse check focuses on verifying the following: (1) the aggregate of the decomposed atomic claims accurately represents the original claim's meaning, (2) each atomic claim is correct and fully

[2] https://pypi.org/project/wikipedia/

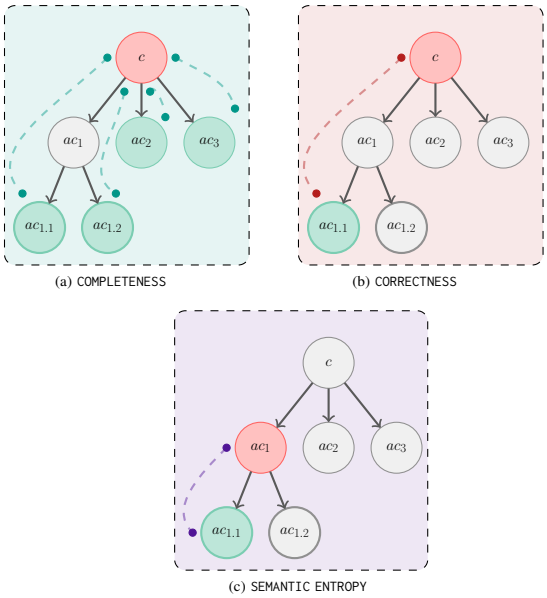

Figure 4: Illustrations for different concepts: (a) Example for COMPLETENESS: Demonstrates leaf node combinations (nodes in green) and the node claim (node in red). (b) Example for CORRECTNESS: Shows the node claim (node in red) and each decomposed atomic claim (nodes in green). (c) Example for SEMANTIC ENTROPY: Illustrates the node claim (node in red) and decomposed atomic claims (nodes in green) ensuring independence among claims.

expressive of the original claim's content, and (3) there is no semantic overlap between atomic claims. Any subtrees failing to meet these conditions are filtered out during the reverse check.

3.2.1 Data Synthesis for DecMetrics

In Step 4, we get decomposition trees that meet the conditions, which are labeled as *supported* examples. The process is visually illustrated in Figure 4. To generate *unsupported* examples, we employ a method that utilizes claims as root nodes and their decomposed atomic claims as leaf nodes within the decomposition trees and their subtrees. This structured approach allows us to systematically dissect complex claims into fundamental components, ensuring precise and detailed training and evaluating datasets.

COMPLETENESS Figure 4(a) illustrates the example of COMPLETENESS. Positive samples are generated by shuffling and pairing leaf node combinations with the node claim, and these are labeled as *supported*. For instance, one such combination in the figure is ($ac_{1.1}$, $ac_{1.2}$, ac_2, ac_3) with the node claim c. In contrast, negative samples are formed by arbitrarily discarding some leaf nodes and pairing the remaining combinations with the node claim, labeled as *unsupported*.

CORRECTNESS Figure 4(b) illustrates the example of CORRECTNESS. A random node is selected from the decomposition tree as the claim at level k. Its child nodes, extending from level $k+1$ to the leaf nodes, act as decomposed atomic claims. Pairs are formed between the node claim and each decomposed atomic claim, resulting in CORRECTNESS-positive samples labeled as *supported*. An example shown involves node claim c and a correctly decomposed atomic claim $ac_{1.1}$. Nodes from subtrees not under node are used to create negative samples labeled as *unsupported*.

SEMANTIC ENTROPY SEMANTIC ENTROPY is depicted to ensure the independence of decomposed atomic claims. The model aims to identify overlaps among these claims. Atomic claims with overlaps are labeled as *supported*; for example, in Figure 4(c), ac_1 overlaps with $ac_{1.1}$. Conversely, atomic claims without overlaps are labeled as *unsupported*, such as ac_1 not overlapping with ac_2. Achieving claim independence is facilitated through the identification of non-overlapping nodes.

We get 621, 2063 and 2063 *supported* examples for COMPLETENESS, CORRECTNESS, and SEMANTIC ENTROPY specifically. In summary, the synthetic data generation process leverages structured methodologies for creating high-quality datasets through iterative refinement. The use of diverse assessment criteria ensures data integrity and accuracy, which are crucial for dependable model training.

3.2.2 Training and Evaluation Synthesized Data for DecMetrics

We partition the decomposition trees into training and evaluation sets, allocating 80% for training and reserving the remaining 20% for evaluation. For training, examples that satisfy the conditions outlined in Step 4 are labeled as *supported*. An dynamic number of *unsupported* examples are sampled to meet the requirement of the training process. In the evaluation data, examples meeting the conditions specified in Step 4 are labeled as *supported*, while an equivalent number of *unsupported* examples are generated using the same synthetic process described in Section 3.2.1.

3.3 Experiment Settings of DecMetrics

To decompose claims into atomic claims in Step 3, we conducted experiments using multiple large language models (LLMs), including LLaMA-3.1-405B-Intruct (AI@Meta, 2024), Mixtral-8x22B (Jiang et al., 2024), GPT-4 (OpenAI et al., 2024), DeepSeek V3 (DeepSeek-AI et al., 2025), and Qwen 3-32B (Yang et al., 2025). We found that Qwen 3-32B was particularly efficient and effective at generating atomic claims that could be processed using regex functions. Therefore, we used Qwen 3-32B in Step 3 to generate synthetic training data.

After Step 3, we obtained a total of 4,889 decomposed claims, derived from text summaries of 180 entities as original claims. In Step 4, these atomic claims were categorized into 3,637 leaf nodes and 1,252 general child nodes. After reverse checking process in Step 4, we kept 778 decomposition trees, the detail of data are shown in Appendix Table 6.

Using these claims and decomposed claims, we created training examples for three metrics: `COMPLETENESS`, `CORRECTNESS`, and `SEMANTIC ENTROPY`. With these synthetic training data, we fine-tune DeBERTa-v3-large (He et al., 2021) for DecMetrics with Localized Contrastive Estimation loss (Gao et al., 2021).

3.4 Performance Evaluation of DecMetrics

We assess the effectiveness of automated evaluation models on both the DecMetrics test dataset and established NLI datasets, specifically SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). Table1 provides a comparative analysis of model performance using standard classification metrics. Our findings demonstrate that existing NLI models, such as **bart-large-mnli**(Lewis et al., 2020), trained on MultiNLI, and cross-encoder models (Reimers and Gurevych, 2019) (**nli-deberta-v3-large**, **nli-roberta-base**, and **nli-MiniLM2-L6-H768**) trained on SNLI and MultiNLI, perform well on traditional NLI datasets. However, these models encounter difficulties with the DecMetrics test data, as exhibited by the disparity between high recall and low precision scores in some models, and high precision but low recall in others. This indicates a need for models that perform consistently well across all classification metrics. Therefore, it is essential to develop models specifically adapted and trained on our synthetic dataset for DecMetrics evaluation.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
DecMetrics	bart-large-mnli	89.50%	82.93%	**99.48%**	90.45%
	nli-deberta-v3-large	89.97%	84.91%	97.23%	90.65%
	nli-roberta-base	93.86%	97.21%	90.32%	93.64%
	nli-MiniLM2-L6-H768	94.34%	97.68%	90.84%	94.13%
	DecMetrics	**99.14%**	**99.65%**	98.62%	**99.13%**
SNLI	bart-large-mnli	79.71%	63.30%	**97.15%**	76.65%
	nli-deberta-v3-large	**94.94%**	**92.89%**	92.31%	**92.60%**
	nli-roberta-base	94.62%	92.69%	91.51%	92.10%
	nli-MiniLM2-L6-H768	94.57%	92.02%	92.16%	92.09%
	DecMetrics	83.34%	73.30%	80.85%	76.89%
MultiNLI	bart-large-mnli	81.40%	65.99%	**97.37%**	78.67%
	nli-deberta-v3-large	**93.98%**	**93.51%**	89.08%	**91.25%**
	nli-roberta-base	92.39%	90.15%	88.02%	89.07%
	nli-MiniLM2-L6-H768	92.17%	89.87%	87.64%	88.74%
	DecMetrics	71.95%	58.81%	67.92%	63.04%

Table 1: Evaluation Metrics for Models across Different Datasets.

4 DecModel: A Standardized Claim Decomposition Model

In this section, we investigate the application of evaluation metrics during model training to create a standardized decomposition model, **DecModel**. This model is designed to effectively decompose claims into atomic components that meet to the criteria of `COMPLETENESS`, `CORRECTNESS`, and `SEMANTIC ENTROPY`.

4.1 Training Pipeline of DecModel

Optimizing a supervised claim decomposition model for the three metrics presents substantial challenges. However, these metrics are advantageous when used as reward signals within a reinforcement learning framework. We adopt the reinforcement learning paradigm, implementing a three-stage training pipeline:

1. **Supervised Fine-Tuning (SFT)**: Initially, we employ supervised learning with high-quality claim decomposition annotations to establish baseline decomposition capabilities for our model.

2. **Reward Modeling**: We define a composite reward function:

$$R(c, ac) = \alpha \cdot \text{cp}(c, ac) + \beta \cdot \text{cr}(c, ac) + \gamma \cdot \text{se}(ac)$$

   where α, β, and γ are tunable hyperparameters determining the relative importance of each metric dimension.

3. **Reinforcement Learning Optimization**: The model is further refined through Proximal Policy Optimization (PPO) (Schulman et al., 2017) using the learned reward function, enhancing its decomposition capabilities while preserving generation diversity.

4.2 Configuration of DecModel

In our experiments, we utilized three T5-based model architectures (Raffel et al., 2020): T5-small with 80 million parameters, T5-base with 250 million parameters, and T5-large with 780 million parameters, as the foundational architectures for our claim decomposition tasks. To isolate the contribution of each reward component, we set α, β, and γ to 1 by default and performed ablations by zeroing out individual coefficients.

5 Claim2Atom: Claim Decomposition Benchmark

Follow the creation of LLM-AGGREFACT (Tang et al., 2024), we constructed **Claim2Atom**, which is a comprehensive evaluation benchmark structuring claims into atomic claims. This benchmark integrates both existing public datasets, FActScore and WICE, and newly curated decomposition examples from DecData.

5.1 FActScore and WICE: Public Datasets for Claim Decomposition Evaluation

The FActScore and WICE datasets do not guarantee the quality of atomic claims. To address this, we filtered out examples with atomic claims that were incomplete, incorrect, or exhibited semantic overlap.

5.2 DecData: A New Dataset for Claim Decomposition Evaluation

From the structured decomposition trees and their associated subtrees, we construct the dataset **DecData**, consisting of pairs in the form of (claim, atomic claims). The **DecData** dataset is partitioned into training and test sets to facilitate the training and evaluation of claim decomposition models.

5.3 Benchmark Details

This multifaceted approach aims to improve both the robustness and the breadth of the benchmark. Table 2 presents the statistics of the resulting claim decomposition benchmark.

The newly constructed benchmark is specifically designed to facilitate a rigorous evaluation across several dimensions. Firstly, it enables a critical assessment of various LLM capabilities. Additionally, it provides a robust framework to evaluate the effectiveness of our specialized decomposition models. Finally, this benchmark is intended to serve as a foundational tool for future research endeavors in the realm of claim decomposition.

Split	Dataset	Claims	Atomic Claims	Max	Avg
FActScore	train	7558	12038	22	1.59
	test	1880	2992	22	1.59
WICE	train	3194	4668	6	1.46
	test	1854	2702	6	1.46
DecData	train	2684	4126	42	1.54
	test	656	1002	19	1.53
ALL	**train**	**13436**	**20832**	**42**	**1.55**
	test	**4390**	**6696**	**22**	**1.53**

Table 2: Statistics of the Claim Decomposition Benchmark.

Model	Size	COMP.	CORR.	SEM.
Qwen3-32B	32B	**95.67%**	94.77%	0.3240
Mistral-NeMo-12B-Instruct	12B	80.08%	80.14%	**0.4420**
Llama-3.2-3B-Instruct	3B	24.37%	10.81%	2.6175
DecModel$_{large}$	0.78B	93.70%	98.96%	0.2823
DecModel$_{base}$	0.25B	92.67%	**99.70%**	0.2620
DecModel$_{small}$	0.08B	91.37%	98.90%	0.2343

Table 3: Evaluation for Claim Decomposition in test data of Claim2Atom.

6 Experimental Setup

In our experiments, we compare decomposition quality against contemporary large language models (LLMs). This comparison aims to highlight the effectiveness of our approach relative to existing models.

For downstream evaluation, we integrate multiple state-of-the-art fact-checking systems to assess the practical utility of the decomposed atomic claims generated by **DecModel**$_{large}$.

Additionally, we conduct an ablation study with **DecModel**$_{large}$ to analyze the individual contribution of each metric in DecMetrics. By isolating each component in the reward structure, we evaluate their impact on overall performance, offering insights into the effectiveness of the metrics in enhancing claim decomposition outcomes.

6.1 Metric Performance Comparison

Table 3 provides a detailed evaluation of our T5-based models, available in various configurations, in comparison to baseline LLMs such as Qwen3-32B, Mistral-NeMo-12B-Instruct and Llama-3.2-3B-Instruct. This table also includes parameter counts of models for efficiency analysis. It can be observed that both Qwen3-32B, Mistral-NeMo-12B-Instruct and Llama-3.2-3B-Instruct achieve high `SEMANTIC ENTROPY` scores but exhibit low `CORRECTNESS` metrics. This suggests that while

Decomposition	Model	BAcc
Not Decompose	SummaC-Conv	**62.1%**
	SummaC-ZS	67.9%
	QAFactEval	**66.5%**
	AlignScore	70.4%
	MiniCheck-RBTA	72.6%
	MiniCheck-DBTA	**72.7%**
	MiniCheck-FT5	**74.7%**
GPT-4	SummaC-Conv	58.8%
	SummaC-ZS	**69.1%**
	QAFactEval	64.6%
	AlignScore	**71.5%**
	MiniCheck-RBTA	**73.2%**
	MiniCheck-DBTA	72.7%
	MiniCheck-FT5	73.3%
DecModel$_{large}$	SummaC-Conv	55.5%
	SummaC-ZS	67.3%
	QAFactEval	60.9%
	AlignScore	69.7%
	MiniCheck-RBTA	72.5%
	MiniCheck-DBTA	71.5%
	MiniCheck-FT5	72.4%

Table 4: Evaluation for Claim Decomposition on the test set of LLM-AGGREFACT.

these LLMs are proficient in generating numerous atomic claims, they often fall short in maintaining factual accuracy with respect to the original claims. Our models, in contrast, exhibit competitive performance levels comparable to those of the LLMs and demonstrate superior parameter efficiency.

6.2 Downstream Fact-Checking Performance

In this analysis, we evaluate the effectiveness of atomic claim decomposition within downstream fact-checking systems. Claims are broken down into atomic facts, with results from non-decomposed and GPT-4 models (Tang et al., 2024) serving as baselines for comparison. Fact-checking systems assess the factuality of each atomic claim; a claim is supported if all its atomic facts are corroborated by the source document, otherwise it is considered unsupported. Note that FactCheck-GPT datasets inherently comprise atomic facts.

Table 4 shows results across a range of fact-checking systems. A slight decrease in performance is observed, attributed to our aggregation approach. This method, demanding support for all atomic claims, can enhance precision but may negatively impact recall, as it includes non-essential claims that are challenging to verify, thereby potentially diminishing recall. To better integrate atomic claims with fact-checking systems, more sophisticated aggregation strategies, such as those employed by SummaC models (Laban et al., 2022), could be explored to optimize performance.

DecModel$_{large}$	COMP.	CORR.	SEM.
w/o DecMetrics	93.70%	98.94%	0.2534
w/o COMPLETENESS	93.12%	99.21%	0.3036
w/o CORRECTNESS	94.08%	98.76%	0.2894
w/o SEMANTIC ENTROPY	93.64%	98.77%	0.2409

Table 5: Ablation study results on the test data of Claim2Atom.

6.3 Ablation Study

Table 5 presents a systematic evaluation of individual reward components by isolating each metric: COMPLETENESS, CORRECTNESS, and SEMANTIC ENTROPY.

The results in Table 5 indicate that removing the DecMetrics as a reward in the SFT without the RL stage results in decreased COMPLETENESS and SEMANTIC ENTROPY. This suggests that omitting DecMetrics tends to produce primarily correct outputs while avoiding uncertain ones, thereby generating fewer atomic claims. Alternatively, employing pairwise combinations of DecMetrics reward components enables the Claim Decomposer to generate more atomic claims, resulting in increased COMPLETENESS, CORRECTNESS, and SEMANTIC ENTROPY scores. Specifically, excluding COMPLETENESS reduces the COMPLETENESS score; omitting CORRECTNESS lowers the CORRECTNESS score; and excluding SEMANTIC ENTROPY diminishes the SEMANTIC ENTROPY score. These observations underscore the synergistic contribution of all three metrics to the overall performance.

7 Conclusion

In this work, we introduce **DecMetrics**, a new set of metrics: COMPLETENESS, CORRECTNESS, and SEMANTIC ENTROPY, designed to improve the evaluation of claim decomposition in fact-checking. We also present **DecModel**, a lightweight model that uses reinforcement learning with DecMetrics to set new standards for generating accurate atomic claims. To further aid research, we propose **Claim2Atom**, a comprehensive benchmark that integrates existing datasets with our new DecData, enabling diverse evaluations in claim decomposition.

8 Limitations

Despite the structured approach to synthetic data generation via Wikipedia summaries and decomposition steps, certain limitations persist:

Generalization Limitations Synthetic data might not fully capture real-world complexity and nuances, limiting model generalizability. Moreover, entity selection may introduce bias, as the chosen 200 subjects do not represent all topics. Specific claim decomposition prompts might perpetuate biases in large language models (LLMs).

Scalability Issues The decomposition and verification process is resource-intensive, limiting scalability and experimentation across larger datasets or different domains. This computational demand can hinder refinement and adaptation efforts.

In summary, these limitations point to essential areas for further improvement to enhance the robustness, generalizability, and efficiency of the synthetic data generation process.

References

AI@Meta. 2024. Llama 3 model card.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X Retrieval: What Retrieval Granularity Should We Use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

I.-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. FacTool: Factuality Detection in Generative AI – A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. *arXiv preprint*. ArXiv:2307.13528 [cs].

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025. DeepSeek-V3 Technical Report. *arXiv preprint*. ArXiv:2412.19437.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline. In Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani, editors, *Advances in Information Retrieval*, volume 12657, pages 280–286. Springer International Publishing, Cham.

Anisha Gunjal and Greg Durrett. 2024. Molecular Facts: Desiderata for Decontextualization in LLM Fact Verification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768, Miami, Florida, USA. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2024. Decomposition Dilemmas: Does Claim Decomposition Boost or Burden Fact-Checking Performance? *arXiv preprint*. ArXiv:2411.02400 [cs].

Hasan Iqbal, Yuxia Wang, Minghan Wang, Georgi Nenkov Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2024. OpenFactCheck: A Unified Framework for Factuality Evaluation of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 219–229, Miami, Florida, USA. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *arXiv preprint*.

Ryo Kamoi, Tanya Goyal, Juan 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.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Haonan Li, Xudong Han, Hao Wang, Yuxia Wang, Minghan Wang, Rui Xing, Yilin Geng, Zenan Zhai, Preslav Nakov, and Timothy Baldwin. 2024. Loki: An Open-Source Tool for Fact Verification. *arXiv preprint*. ArXiv:2410.01794 [cs].

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

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.

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, C. J. 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. *arXiv preprint*. ArXiv:2303.08774.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, Accurate and Easy to Use Coreference Resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 53728–53741, Red Hook, NY, USA. Curran Associates Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint*. ArXiv:1707.06347 [cs].

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024. A Closer Look at Claim Decomposition. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 153–175, Mexico City, Mexico. Association for Computational Linguistics.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. *arXiv preprint*. ArXiv:2403.18802 [cs].

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao

Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. *arXiv preprint*. ArXiv:2505.09388.

A Claim Decomposition Algorithm

The algorithm to decompose claims into atomic claims is shown in Appendix Algorithm 1.

Algorithm 1 Claim Decomposition Algorithm

Require: $claim$: A claim to be decomposed
Ensure: $atomic_claims$: A list of atomic claims

$atomic_claims \leftarrow$ empty list
function DecomposeRecursively($claim$, $result$)
  $sub_claims \leftarrow decompose_model(claim)$ {Decompose the claim into sub-claims}
if length(sub_claims) $= 1$ **then**
  append $claim$ to $result$ {If the claim is atomic, add it to the result}
  return
end if
for sub_claim **in** sub_claims **do**
  $sub_result \leftarrow$ empty list
  DecomposeRecursively(sub_claim, sub_result)
  append $\{sub_claim : sub_result\}$ to $result$
end for
end function
DecomposeRecursively($claim$, $atomic_claims$)

B Data of Reverse Checking Process

In this section, we present the detailed data involved in the reverse checking process, categorized into *supported* and *unsupported* labels. Appendix Table 6 categorizes the metrics used for analysis along with the corresponding data statistics, such as the total number of claims and the average count of atomic claims. Our observations reveal that claims comprising a higher number of atomic statements tend to exhibit higher failure rates concerning COMPLETENESS, CORRECTNESS, and SEMANTIC ENTROPY.

Label	Metric	Number of Claims	Avg of Atomic Claims
supported	COMPLETENESS	1235	6.06
	CORRECTNESS	1273	5.84
	SEMANTIC ENTROPY	805	3.3
unsupported	COMPLETENESS	197	10.09
	CORRECTNESS	159	12.79
	SEMANTIC ENTROPY	627	10.86

Table 6: Detailed Data of Reverse Checking Process.

In Appendix Figure 5, we provide samples of the claims and the corresponding atomic claims identified as *unsupported* during the reverse checking process. These samples highlight the typical issues encountered with unsupported data and offer insights into the complexity of atomic claims.

C Post-Processing Strategies

We explore two such approaches: claim decontextualization and verifiability.

C.1 Decontextualization

Decontextualization refines atomic claims by clarifying or modifying context that may obscure their true meaning. Techniques such as coreference resolution, exemplified by FactCoef(Otmazgin et al., 2022), process pairs of (c, ac_i) to support decontextualization. Alternatively, large language model (LLM) approaches, like MOLECULAR-DECONTEXT(Gunjal and Durrett, 2024), enhance ac_i with additional context for clarity and precision. In this study, we employ SIMPLE-DECONTEXT (Gunjal and Durrett, 2024) to decontextualize decomposed atomic claims.

C.2 Verifiability

Verifiability involves distinguishing between content that can be substantiated and that which cannot, particularly in long-form generation tasks. Assuming all claims are verifiable may lead to extracting entire texts, including unverifiable components such as examples or hypotheticals, potentially unfairly penalizing models during evaluation. In this study, we use VERISCORE (Song et al., 2024) to evaluate the verifiability of each decomposed atomic claim.

C.3 Results

Appendix Table 7 presents the number of atomic claims edited through decontextualization and removed due to verifiability concerns. Our dataset, derived from Wikipedia entities, frequently includes incomplete names or pronouns, necessitating decontextualization for standalone clarity. However, decontextualization may not significantly

Type: not complete
Claim: Sweden is increasingly looking at waste-to-energy generation as an energy diversification strategy.
Atomic Claims: ['Sweden is looking at waste-to-energy generation.', 'Waste-to-energy generation is considered as an energy diversification strategy.']

Type: not correct
Claim: She held a role under John Major.
Atomic Claims: ['She held a role in government.', 'The role was under John Major.']

Type: not independent
Claim: Àlex Corretja was the defending champion, but did not participate. Nicolás Lapentti won the title, defeating Vince Spadea in the final 4–6, 6–4, 6–4.
Atomic Claims: ['Àlex Corretja was the defending champion.', 'Àlex Corretja did not participate.', 'Nicolás Lapentti won the title.', 'Vince Spadea was in the final match.', 'Nicolás Lapentti defeated Vince Spadea in the final.', 'The final score was 4–6, 6–4, 6–4.']

Figure 5: Samples for *unsupported* data in reverse checking process: claim and atomic claims.

enhance fact-checking systems as these models inherently infer context-dependent information, particularly when retrieval stages provide context alignment. While decontextualization can improve readability by eliminating context dependence, merely replacing "he" with "Lanny Flaherty" may not enhance fact-checking performance. Additionally, errors introduced during decontextualization could actually reduce fact-checking effectiveness, as demonstrated in (Tang et al., 2024).

Furthermore, some decomposed atomic claims are unverifiable, as they cannot be definitively proven true or false, such as "Betacyanin is like a superhero cape". Despite their correctness and relevance to the original text, these atomic claims must be generated when decomposing a claim but can be disregarded or labeled as unverifiable during verification. These results indicate a significant potential for quality improvement through these post-processing strategies.

Strategy	Filter Num of Atomic Claims
Decontextualization	14515
Verifiability	5566

Table 7: Statistics of the post-processing strategies.

D Prompt for Claim Decomposition and Reverse Checking

The prompt for claim decomposition is shown in Appendix Figure 6, and the prompt for for reverse checking is shown in Appendix Figure 7.

Claim Decomposition Prompt

Break down the Claim into simple Atomic Claims.
Steps:

1. Check if the Claim is simple. If yes, return it as is.
2. If complex, identify key components and break into distinct Atomic Claims.
3. Ensure Atomic Claims cover the full meaning without redundancy.
4. Preserve Named Entities and replace pronouns with full names.
5. Verify each fact is clear, distinct, and self-contained.

Example Walkthrough:
Claim: Ash is known for her comedic timing and her ability to play a wide range of characters, from quirky and offbeat to more serious and dramatic roles.
Steps:

1. Claim is complex.
2. Break into:
   - Ash is known for her comedic timing.
   - Ash is known for her ability to play a wide range of characters.
   - The range includes quirky and offbeat roles.
   - The range includes more serious and dramatic roles.
3. Atomic Claims cover all aspects.
4. Named Entities (Ash) preserved.
5. Atomic Claims are clear and distinct.

Atomic Claims:

```
<answer>
  - Ash is known for her comedic timing.
  - Ash is known for her ability to play a wide range of characters.
  - The range includes quirky and offbeat roles.
  - The range includes more serious and dramatic roles.
</answer>
```

Figure 6: Claim Decomposition Prompt with Chain of Thought

Claim Decomposition Prompt (Continued)

Claim: 38 artistic roller skating competitors, from 11 nations, participated in the tournament.
Steps:

1. Claim is complex.
2. Break into:
   - 38 artistic roller skating competitors participated.
   - Competitors came from 11 nations.
3. Atomic Claims cover all aspects.
4. Named Entities preserved.
5. Atomic Claims are clear and distinct.

Atomic Claims:

```
<answer>
  - 38 artistic roller skating competitors participated.
  - Competitors came from 11 nations.
</answer>
```

Claim: The artistic roller skating tournaments took place between 22 and 23 July.
Steps:

1. Claim is simple.
2. Return as is.

Atomic Claims:

```
<answer>
  - The artistic roller skating tournaments took place between 22 and 23 July.
</answer>
```

Claim: [Claim]

Atomic Claims Verifying Prompt

Verify the Atomic Claims against the Original Claim for completeness, correctness, and independence.
Steps:

1. Aggregate the Atomic Claims to ensure they fully represent the meaning of the Original Claim.
2. Check that each Atomic Claim is factually correct and contains no information absent from the Original Claim.
3. Identify any semantic overlap between Atomic Claims, ensuring each is distinct and independent.

Example Walkthrough:
Original Claim: Ash is known for her comedic timing and her ability to play a wide range of characters, from quirky and offbeat to more serious and dramatic roles.
Atomic Claims:

- Ash is known for her comedic timing.
- Ash is known for her ability to play a wide range of characters.
- The range includes quirky and offbeat roles.
- The range includes more serious and dramatic roles.

Analysis:

1. The Atomic Claims collectively represent the complete Original Claim.
2. Each Atomic Claim is factually accurate and sourced directly from the Original Claim.
3. There is no semantic overlap; each Atomic Claim is distinct.

Conclusion:

```
<answer>
  - complete
  - correct
  - independent
</answer>
```

Figure 7: Atomic Claims Verifying Prompt with Chain of Thought

Atomic Claims Verifying Prompt (Continued)

Original Claim: 38 artistic roller skating competitors, from 11 nations, participated in the tournament.
Atomic Claims:

- 36 artistic roller skating competitors participated.

Analysis:

1. The Atomic Claims do not fully represent the Original Claim.
2. An Atomic Claim is factually inaccurate, diverging from the Original Claim.
3. There is no semantic overlap; each Atomic Claim is distinct.

Conclusion:

```
<answer>
  - not complete
  - not correct
  - independent
</answer>
```

Original Claim: The artistic roller skating tournaments took place between 22 and 23 July.
Atomic Claims:

- The artistic roller skating tournaments took place between 22 and 23 July.
- The artistic roller skating tournaments took place in July.

Analysis:

1. The Atomic Claims represent the complete Original Claim.
2. Each Atomic Claim is factually consistent with the Original Claim.
3. There is semantic overlap.

Conclusion:

```
<answer>
  - complete
  - correct
  - not independent
</answer>
```

Original Claim: [Original Claim]
Atomic Claims: [Atomic Claims]
Analyze and conclude whether the Atomic Claims meet the criteria or require adjustments.

Figure 8: Atomic Claims Verifying Prompt with Chain of Thought (Continued)