

# FACT5: A Novel Benchmark and Pipeline for Nuanced Fact-Checking of Complex Statements

Shayan Chowdhury<sup>1</sup>, Sunny Fang<sup>2</sup>, Smaranda Muresan<sup>2</sup>

<sup>1</sup>Department of Computer Science, Columbia University, New York, NY, USA

<sup>2</sup>Department of Computer Science, Barnard College, Columbia University, New York, NY, USA

## Abstract

Fact-checking complex statements is integral to combating misinformation, but manual approaches are time-consuming, while automated approaches often oversimplify truthfulness into binary classifications and rely on resource-intensive models. This paper introduces: (i) FACT5, a curated dataset of 150 real-world statements with five ordinal classes of truthfulness, designed to capture the nuanced nature of factual accuracy and (ii) an open-source end-to-end pipeline using large language models (LLMs) that decomposes statements into atomic claims, generates targeted questions, retrieves evidence from the web, and produces justified verdicts. We evaluate our pipeline on FACT5 using MISTRAL-7B-v0.3 and Google’s GEMINI-1.5-FLASH. Our findings demonstrate significant improvements over baseline LLM performance, with MISTRAL-7B showing a 71.9% reduction in MSE for pass@3 evaluation. The FACT5 dataset, pipeline implementation, and evaluation framework are anonymized and provided at <https://github.com/shyantist/FACT5/>, and a demo of the pipeline can be interacted with at <https://fact5check.streamlit.app/>.

## 1 Introduction

Traditionally, fact-checking relied on time-consuming and resource-intensive work by human experts (e.g., crowd-sourcing). With the widespread dissemination of mis/disinformation, coupled with growing capabilities of large language models (LLMs), recent work has been exploring how to leverage LLMs for automated fact-checking. Wang et al. (2024) proposed Factcheck-GPT, an approach that uses retrieval-augmented methods to detect and correct factual errors in natural language, including text generated by LLMs, which can be prone to hallucinations. Gou et al. (2023) introduced the CRITIC framework for LLM self-

correction, which extends the output through iterations of verification and correction.

When handling claims that require reasoning, existing methods fail to capture the nuances of factuality by relying on binary true/false labels (Vlachos and Riedel, 2014). Factcheck-GPT by Wang et al. (2024) addresses this issue by breaking down complex claims into a series of atomic facts to verify separately, whereas Min et al. (2023) proposed more fine-grained metrics such as FACTSCORE, computing the percentage of atomic facts supported by reliable knowledge sources.

Despite progress, challenges remain, particularly for fact-checking long-form statements that are decomposable and require multi-hop reasoning. Our work addresses these challenges with the following contributions: (i) curated test dataset—named **FACT5: Factual Analysis of Complex Truths (5-label)**—of 150 statements with a five-class ordinal scale for truthfulness classification, (ii) a comprehensive end-to-end pipeline that supports ranked web-based search, multi-hop reasoning, question-answering, and five-way classification, and (iii) enhanced transparency and explainability through source citations and reasoning at each intermediary step. In addition, we perform an error analysis that provides insights into the types of fallacies present in misleading claims (e.g., hasty generalization, causal oversimplification).

## 2 Related Work

In recent years, automated fact-checking has gained traction in the journalistic process, from pioneers such as ClaimBuster (Hassan et al., 2017), to novel approaches such as one that leverages a frame-semantic parser (FSP) (Devasier et al., 2025). Guo et al. (2022) comprehensively reviewed the state of fact-checking research as of 2022. A common theme recognizes that simple true/false labels for factuality fall short in capturing factual correctness

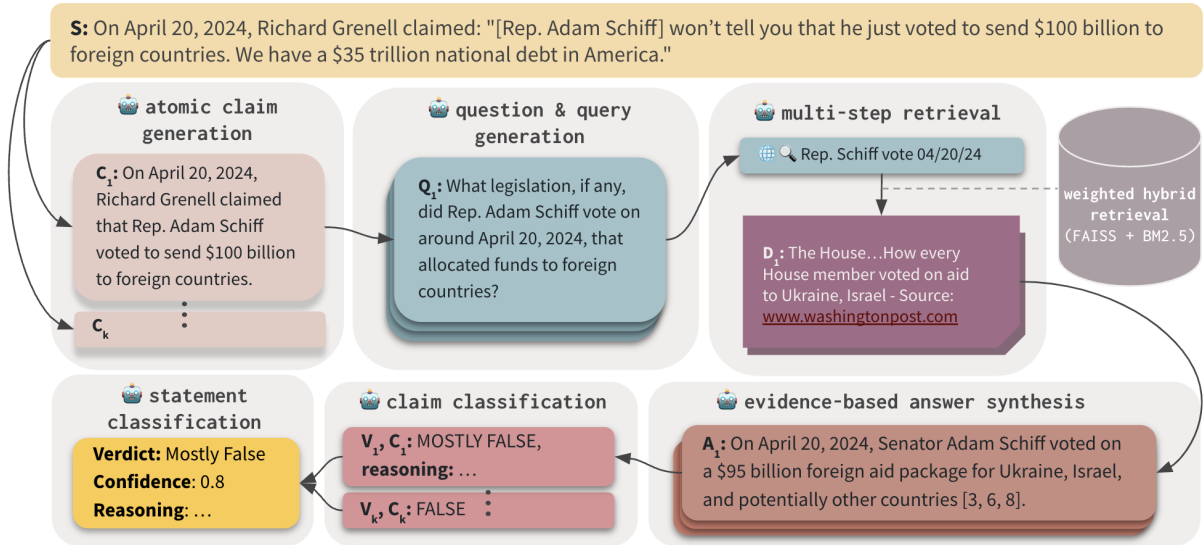


Figure 1: Overview of our proposed pipeline. Each gray box indicates a step detailed in §3.2

in abstractive or complex (e.g., political) settings. Thus, research has developed multi-level typologies or turned to sources such as PolitiFact as potential datasets (Wang, 2017; Ma et al., 2023; Devasier et al., 2025), which informed the design of our evaluation dataset detailed in §3.1. Other benchmarks such as AVERITEC extend slightly the binary labels by adding conflicting evidence/cherry picking (Schlichtkrull et al., 2024). However, challenges remain, as Pagnoni et al. (2021) notes how conventional metrics in natural language processing (NLP), such as METEOR score, fall short in measuring factual correctness of generated reasoning. Factuality classification aside, quality evaluation of textual justifications is an emerging direction for fact-checking (Russo et al., 2023).

Our fact-checking pipeline, detailed in §3.2, is outlined in the following steps: (1) atomic claim generation, (2) question/query generation, (3) retrieval of relevant documents, (4) answer synthesis, (5) claim-wise classification, and (6) overall statement classification and justification. Similar approaches have been seen in past research (Wang et al., 2024; Rothermel et al., 2024; Wei et al., 2024). Other work has focused on investigating intermediary steps, such as atomic claim generation (Gunjal and Durrett, 2024; Wanner et al., 2024a) and found that conducting decomposition and decontextualization in one step yields optimal results (Wanner et al., 2024b), which we incorporated into our pipeline. Steps 2 through 4 are inspired by past work that demonstrates the importance of related information, oftentimes in the form of question-

answer pairs, in improving the accuracy of fact-checking (Fan et al., 2020). Furthermore, we focus on using smaller models that can be run on consumer hardware.

### 3 Methods

#### 3.1 Test Dataset

The need for a new dataset (FACT5) stems from several critical limitations in existing fact-checking datasets. Current datasets predominantly use binary true/false labels, which fail to capture the nuanced nature of factual correctness in complex statements. While AVERITEC represents an important benchmark in fact-checking research, our decision to evaluate primarily on FACT5 was motivated by several key factors, including label granularity and temporal relevance.

Additionally, concerns about data contamination and model memorization necessitate fresh data collection (Balloccu et al., 2024; Carlini et al., 2022). As LLMs may have been trained on existing fact-checking datasets, evaluating their true fact-checking capabilities requires testing on previously unseen claims. Our dataset’s temporal range (January 2024 to January 2025) ensures the evaluation of model performance on genuinely novel information rather than memorized training data.

We developed our **FACT5** dataset as an initial benchmark for evaluating nuanced fact-checking capabilities. We began by collecting candidate statements from recognized fact-checking insti-

Sources	Verdicts				
	F	MF	HT	MT	T
PolitiFact	21	24	22	24	20
Snopes	4	0	1	1	4
WaPo	9	3	2	0	0
CNN	8	0	2	2	3
Total	42	27	27	27	27

Table 1: Summary of Dataset

F = FALSE, MF = MOSTLYFALSE, HT = HALFTTRUE, MT = MOSTLYTRUE, T = TRUE

tutions, including PolitiFact<sup>1</sup>, Snopes.com<sup>2</sup>, The Washington Post’s Fact Checker Section<sup>3</sup>, and CNN’s Facts First<sup>4</sup>, covering the period from January 10th, 2024, to January 31st, 2025. Following a meticulous manual curation process, we prioritized statements necessitating multi-hop reasoning and ensured consistent mapping of original verdict labels to our five-class ordinal scale, resulting in our final FACTS dataset comprising 150 statements. Future work on expanding the size and scope of this benchmark dataset is in §6. This carefully selected set is intended as a focused resource for testing models on this complex, fine-grained classification task. The distribution of sources is summarized in Table 1.

A key methodological decision was to prioritize sources that provide gold label evaluations alongside fact-check analyses, whose labels can be mapped to PolitiFact’s *Truth-O-Meter* labels. This was essential for our classification objective, though it necessarily narrowed the pool of eligible source materials. *Truth-O-Meter* labels are provided in Table 6 in Appendix A.1. Mappings for the labels among data sources are in Appendix A.2, Table 7. A snippet of the dataset can be found in Appendix A.3.

Based on the *Truth-O-Meter*, the five classes used to classify a given statement are TRUE, MOSTLYTRUE, HALFTTRUE, MOSTLYFALSE, and FALSE, representing an ordinal scale of factuality. The verdict UNVERIFIABLE is provided as an option for models to explicitly state when there is insufficient evidence to make a verdict. The dataset contains  $n = 42$  statements labeled FALSE and  $n = 27$  for each of the other four factuality labels, reflecting the real-world importance of identifying

<sup>1</sup>[www.politifact.com](http://www.politifact.com)

<sup>2</sup>[www.snopes.com](http://www.snopes.com)

<sup>3</sup>[www.washingtonpost.com/politics/fact-checker](http://www.washingtonpost.com/politics/fact-checker)

<sup>4</sup>[www.cnn.com/specials/politics/fact-check-politics](http://www.cnn.com/specials/politics/fact-check-politics)

falsehoods.

### 3.2 Fact-Checking Pipeline

This section details each step of our fact-checking pipeline, visualized in Figure 1.<sup>5</sup> For prompting, we leverage the DSPy library (Khattab et al., 2022, 2023), a framework that optimizes language model outputs via a declarative programming approach, replacing manual prompt engineering. Each step also leverages chain-of-thought (CoT) prompting to elicit improved reasoning capabilities (Wei et al., 2022).

**Step 1: Atomic Claim Generation.** Given a statement from §3.1, the model is prompted to decompose and decontextualize the statements into **atomic\_claims**, a list of strings. Each claim should not rely on additional context to be understood and should focus on a single idea or concept (Barsalou, 1982; Wang et al., 2024).

**Step 2: Question & Query Generation.** For each claim in **atomic\_claims**, the model is prompted to generate two key components: (1) questions that break down the claim into verifiable sub-components and (2) search queries optimized for retrieving relevant evidence.

**Step 3: Multi-Stage Retrieval of Relevant Documents.** We implement a custom retrieval-augmented generation (RAG) system (Lewis et al., 2020) that involves fetching information from external sources (i.e., the internet) and a hybrid retrieval approach combining dense and sparse retrieval. For each claim, we iterated through the list of **questions** and conducted two sub-steps for each question:

**Step 3a: Querying.** Using the queries generated for each question from Step 2, we conduct web searches via API calls. We have implemented functionality to use both DuckDuckGo as well as Google Search via Serper’s Search Engine Results Page (SERP) API<sup>6</sup>, which returns a list of **search\_results**, each containing the title, a search engine-provided excerpt, and website metadata.

**Step 3b: Dense-Sparse Hybrid Retrieval.** We first split the **search\_results** retrieved from the

<sup>5</sup>For each, see Appendix B for criteria & DSPy signature

<sup>6</sup><https://serper.dev/>

web into chunks and processed them for retrieval using a dual-index system as demonstrated in Wang et al. (2021). For dense retrieval, we utilize the all-MiniLM-L6-v2 pre-trained sentence embedding model (Wang et al., 2020) using the SentenceTransformers library to calculate vector representations of each text chunk, then store them in a vector database—in this case, Facebook AI Similarity Search (FAISS) (Douze et al., 2024) due to its efficiency and ease of implementation, but any other vector database such as ChromaDB could also be used. Simultaneously, for sparse retrieval, we implement BM25, the keyword text-retrieval algorithm using the BM25Okapi library for traditional lexical matching. To combine these two retrieval methods, we use a weighted combination ( $\alpha * BM25 + (1 - \alpha) * FAISS$ ) to determine final document relevance, ensuring that the retrieved documents are both semantically and lexically similar to the query, similar to how web search engines work. We then retrieve 10 documents with the highest combined relevance scores to help answer each question in the following step.

**Step 4: Evidence-Based Answer Synthesis.** For each question, the pipeline synthesizes answers using the relevant evidence. Since each chunk of evidence retrieved retains metadata regarding the source, we can maintain provenance through explicit source attribution and inline citations. Furthermore, the pipeline also tracks the relevance score of each document to the question to help with the synthesis process to weigh the importance of each document in the final answer. This process is then repeated for every single question-answer pair for each claim.

**Step 5: Claim-Wise Classification.** With a list of question-answer pairs and evidence for each atomic claim, each claim is then evaluated for truthfulness, assigned one of the five ordinal classes or 'UNVERIFIABLE', and accompanied by a justification.

**Step 6: Overall Statement Classification.** Similar to the claim-wise classification step, the overall statement containing all the claims is then evaluated for truthfulness. The final verdict for the statement is determined by considering the atomic claims—and each of their question-answer pairs, verdicts, and confidence scores from step 5—inter-claim relationships, and the original statement.

Since the truthfulness reasoning of each claim contains information pertinent to determining the overall statement’s truthfulness, we harness the reasoning capabilities of the model (Zhang and Gao, 2023). Adopting the same class labels from the claim-wise classification in the previous step, we finally classify the overall statement into one of the six classes (five ordinal plus UNVERIFIABLE).

### 3.3 Language Models Used & Technical Specifications

Models used for our research include GEMINI-1.5-FLASH and MISTRAL-7B-v0.3. MISTRAL-7B is an open-weight model that utilizes Grouped Query Attention (GQA) and Sliding Window Attention (SWA) to improve performance and lower cost (Jiang et al., 2023). MISTRAL-7B-v0.3, an iteration upon previous versions, features a vocabulary of 32,768 tokens, enhancing the model’s language understanding and generation capabilities (Jiang et al., 2024). Google’s GEMINI-1.5-FLASH is designed for high-volume, cost-effective applications. It is online-distilled from GEMINI-1.5-PRO, a sparse mixture of experts (MoE) model; its number of parameters is not disclosed but can be reasonably estimated to be somewhere between 8B and 200B (Team et al., 2024).

We chose these two models specifically due to their cost-effectiveness and performance to maximize accessibility and ease of use: MISTRAL-7B is open-source and can be run locally on many consumer-grade hardware, while GEMINI-1.5-FLASH has a "free tier" with limited rate limits but is still a very powerful, versatile, and fast model. We ran our experiments on a MacBook Pro with an M1 Pro processor and 16GB of RAM using Ollama (Ollama, 2024) to leverage MISTRAL-7B for local inference, requiring roughly 3 GPU hours per pass over the FACT5 dataset. In total, we ran 3 passes through the dataset for each model, taking roughly 9 GPU hours for MISTRAL-7B and 2.5 GPU hours for GEMINI-1.5-FLASH. All models were run with a temperature of 0.3 and a maximum context length of 8192 tokens.

## 4 Evaluation

### 4.1 Ablation Studies

Beyond the full pipeline (§3.2), we evaluated baseline LLM performance where only the statement itself was provided for factuality prediction. The two main methods tested are as follows:

- **Baseline:** Only the statement is given to the model to generate a factuality label.
- **Pipeline:** After iterating through the proposed pipeline (§3.2), the statement, atomic claims, question-answer pairs, and claim assessment are given to the model to generate a factuality label.

If providing relevant information queried from the internet enhances the model’s fact-checking capability, it would demonstrate the model’s ability to effectively synthesize and reason over external knowledge sources, a desirable trait for reliable automated claim verification systems. The baseline condition provides a basis for comparison to see if the model can answer accurately without external information. Since not all language models have an explicit cutoff date, a fair baseline performance makes it challenging to know if the correct answer stems from memorization.

## 4.2 Evaluation Metrics

As mentioned in §3.1, our work treats fact-checking as an **ordinal multi-class classification** task. Our evaluation framework first mapped ordinal verdict classes to numerical values (TRUE = 5, MOSTLYTRUE = 4, HALFTTRUE = 3, MOSTLYFALSE = 2, FALSE = 1). Crucially, although not in the gold dataset, the label UNVERIFIABLE is a possible output for the LLMs at the classification step, erring on the side of caution when there is insufficient evidence. For ordinality-based metrics, we excluded UNVERIFIABLE verdicts from calculations. This decision was motivated by two key factors: the inherent difficulty of quantifying the ‘distance’ between an UNVERIFIABLE verdict and ordinal categories, and the fundamentally different nature of UNVERIFIABLE claims, which indicate insufficient evidence rather than a position on the ordinal truth spectrum.

Drawing lessons from Kulal et al. (2019), we employ the  $\text{pass}@k$  metric when evaluating model outputs. Under this paradigm, we prompted the model  $k$  times for each statement. For the ablation study detailed in §4.1, we extracted labels for  $\text{pass}@1$  and  $\text{pass}@3$ . Specifically, for  $\text{pass}@1$  evaluation, we considered only the first prediction and excluded UNVERIFIABLE responses, whereas for  $\text{pass}@3$  evaluation, we sorted predictions by their Mean Squared Error (MSE) and selected the best non-UNVERIFIABLE prediction if available.

The ordinal nature of classes calls for an evaluation metric that penalizes misclassifications that are “further” away from the gold label. For example, misclassifying FALSE as MOSTLYFALSE should be less penalized than misclassifying it as MOSTLYTRUE. Several studies have investigated the most appropriate way of handling ordinal classification (Cardoso and Sousa, 2011; Sakai, 2021; Amigó et al., 2020). Literature suggests that MSE remains a better metric when the severity of errors weighs more (Gaudette and Japkowicz, 2009). MSE, thus, serves as our primary metric, with lower values indicating better performance.

Another evaluation metric measures the inter-rater agreement between expert fact-checkers (i.e., gold verdict from our dataset) and LLMs. Cohen’s *quadratic weighted  $\kappa$*  is well-suited for ordinal multi-class classification (Cohen, 1968; Yilmaz and Demirhan, 2023). Similar to MSE, disagreements farther apart are weighed more with quadratic weights. The metric ranges from -1 to 1, with values closer to 1 indicating better agreement. We conducted listwise deletion (i.e., dropping statements if prediction is UNVERIFIABLE) as suggested in the findings of De Raadt et al. (2019).

Macro-average metrics remain crucial in evaluating multi-class classification performance, although ordinality is not considered. Macro metrics and balanced accuracy consider the overall performance without taking into account class sizes, which is well-suited for our purposes since correctness regardless of class is crucial for fact-checking (Grandini et al., 2020). §5 discusses results in detail.

## 5 Results

We evaluated model architectures mentioned in §3.3 and §4.1 with metrics detailed in §4.2.

### 5.1 Comparative Analysis

We observe that both model implementations demonstrate the effectiveness of our pipeline, with MISTRAL-7B demonstrating larger relative MSE reductions over its baseline (55.8% at  $\text{pass}@1$  and 71.9% at  $\text{pass}@3$ ) compared to GEMINI-1.5-FLASH (43.7% and 67.1% respectively). The pipeline consistently showed better coverage across both models, particularly with MISTRAL, making verifiable predictions in nearly all cases (147/150 at  $\text{pass}@3$ ). For  $\text{pass}@3$ , our methodology of selecting the best non-UNVERIFIABLE prediction among

the top three responses allowed both systems to improve their performance compared to pass@1, with the pipeline showing particularly strong gains.

The stark difference in improvement percentages between pass@1 and pass@3 reveals an interesting characteristic of our pipeline. While both systems benefit from multiple prediction opportunities, our pipeline shows a more pronounced improvement with additional chances, suggesting that the pipeline maintains a more reliable ranking of alternative verdicts. Even when the first prediction isn't perfect, the correct verdict is more likely to appear in subsequent predictions, suggesting the pipeline's uncertainty estimation is better calibrated, allowing it to generate meaningful alternative verdicts rather than just variations of the same prediction. In turn, the pipeline also enhances the model's ability to reduce UNVERIFIABLE predictions while maintaining or improving accuracy. This suggests that our structured approach helps models make more definitive verdicts without sacrificing reliability.

pass@ <i>k</i>	Ablation	GEMINI-1.5-FLASH	MISTRAL-7B-v0.3
1	baseline	0.434	0.232
	pipeline	<b>0.681</b>	<b>0.516</b>
3	baseline	0.444	0.284
	pipeline	<b>0.810</b>	<b>0.702</b>

Table 2: Cohen's  $\kappa$  by ablation

**GEMINI-1.5-FLASH.** As shown in Figure 2, for our implementation with GEMINI-1.5-FLASH, our pipeline demonstrated substantial improvements over the baseline. In pass@1 evaluation, where we considered only the first prediction and excluded UNVERIFIABLE responses, the pipeline achieved an MSE of 1.3 compared to the baseline's 2.3, representing a 43.7% reduction in MSE. As for Cohen's quadratic weighted  $\kappa$ , the pipeline achieved 0.68 compared to the baseline's 0.43 on a [-1,1] scale, showing a 56.9% improvement in  $\kappa$ . The pipeline also maintained better coverage, handling 101 of 150 cases (49 UNVERIFIABLE predictions excluded) compared to the baseline's 89 cases (61 UNVERIFIABLE excluded).

The improvement was even more pronounced in pass@3 evaluation, where we sorted predictions by their MSE and selected the best non-UNVERIFIABLE prediction. Here, the pipeline achieved an MSE of 0.8 compared to the baseline's

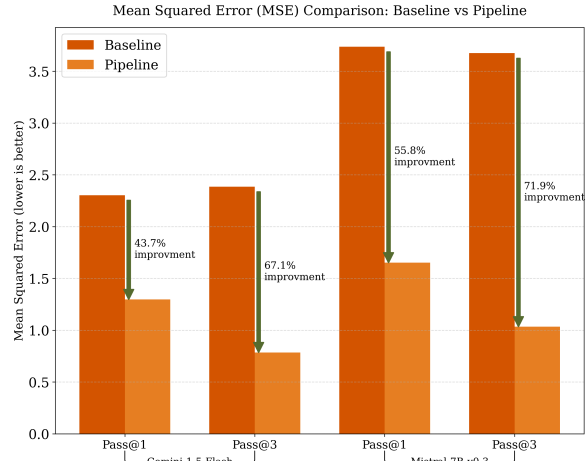


Figure 2: Comparison of Mean Squared Error (MSE), showcasing improvement of pipeline on our multi-class ordinal truthfulness classification task, with sample size (out of 150) as follows (UNVERIFIABLES excluded):

GEMINI: pass@1: base: n=89, pipeline: n=101;

pass@3: base: n=96, pipeline: n=121

MISTRAL: pass@1: base: n=72, pipeline: n=135;

pass@3: base: n=89, pipeline: n=147

2.4, marking a 67.1% reduction in error. Similarly, the pipeline outperformed in terms of Cohen's quadratic weighted  $\kappa$ , showcasing an 82.5% improvement with 0.81 compared to the baseline's 0.44. The pipeline successfully processed 121 cases (excluding 29 cases where all predictions were UNVERIFIABLE) while the baseline handled 96 cases (54 UNVERIFIABLE excluded).

**MISTRAL-7B-v0.3.** Figure 2 also demonstrates our implementation using Mistral-7B showing even stronger improvements. In pass@1 evaluation, the pipeline achieved an MSE of 1.6 compared to the baseline's 3.7, representing a 55.8% improvement. The pipeline achieved Cohen's quadratic weighted  $\kappa$  of 0.12 compared to the baseline's 0.06, demonstrating a 122.5% improvement. The pipeline also demonstrated significantly better coverage, processing 135 out of 150 cases (15 UNVERIFIABLE excluded) compared to the baseline's 72 cases (78 UNVERIFIABLE excluded).

For pass@3 evaluation, the pipeline achieved an MSE of 1.0 compared to the baseline's 3.7, showing a 71.9% reduction in error. Likewise, the pipeline achieved Cohen's quadratic weighted  $\kappa$  of 0.31 compared to the baseline's 0.09, demonstrating a 147.1% improvement. Moreover, pipeline maintained exceptional coverage with 147 cases (only 3 fully UNVERIFIABLE cases excluded) compared to the baseline's 89 cases (61 UNVERIFI-

Model	Ablation	Acc	Prec	Recall	F1	
GEMINI	1	B	0.21	0.24	0.15	0.17
		P	<b>0.25</b>	<b>0.34</b>	<b>0.19</b>	<b>0.21</b>
	3	B	0.23	0.29	0.17	0.19
		P	<b>0.41</b>	<b>0.40</b>	<b>0.33</b>	<b>0.33</b>
MISTRAL	1	B	0.14	0.18	0.10	0.12
		P	<b>0.27</b>	<b>0.34</b>	<b>0.22</b>	<b>0.23</b>
	3	B	0.18	0.22	0.13	0.15
		P	<b>0.43</b>	<b>0.48</b>	<b>0.35</b>	<b>0.35</b>

Table 3: Balanced Accuracy and Macro Metrics (Precision, Recall, and F1-score) by LLMs and ablation: pass@ $k$  and B = baseline, P = pipeline (our method). Best performance in each group are in bold.

ABLE excluded).

## 5.2 Performance Analysis

Additional classification metrics—balanced accuracy, macro recall, precision, and F1-scores—are presented in Table 3. While we excluded UNVERIFIABLE predictions from the MSE calculations to avoid distorting distance-based penalties—given that this class does not adhere to the ordinal structure—we included them in the macro calculations as it evaluates classification performance across all classes independently, allowing us to evaluate the models’ performance across all possible outcomes.

The results demonstrate that our pipeline consistently outperforms the baseline across all metrics. Notably, the pass@1 pipeline configuration even surpasses the pass@3 baseline for both models.

Notably, MISTRAL-7B-V0.3 achieves the best overall performance among the tested models. This superior performance aligns with our earlier observation in the MSE analysis (Figure 2), where MISTRAL-7B-V0.3 showed a tendency to make fewer UNVERIFIABLE predictions. In the context of these classification metrics, this characteristic suggests that MISTRAL-7B-V0.3 may be more decisive in assigning specific truthfulness categories, potentially contributing to its improved performance across all classes.

To better understand the performance of our pipeline on different classes, we analyzed the distribution of predictions for each right verdict class in our pass@3 evaluation, revealing interesting patterns as shown in Table 4. For FALSE claims, the pipeline shows strong discrimination between FALSE (45%) and MOSTLYFALSE (48%) verdicts, with minimal confusion with more positive ver-

Gold Verdict	Top 3 Predictions		
FALSE (F)	<i>MF</i>	<i>F</i>	<i>HT</i>
	0.48	0.45	0.04
MOSTLYFALSE (MF)	<i>MF</i>	<i>HT</i>	<i>MT</i>
	<b>0.52</b>	0.22	0.15
HALFTRUE (HT)	<i>MT</i>	<i>MF</i>	<i>HT</i>
	0.40	0.33	0.22
MOSTLYTRUE (MT)	<i>MT</i>	<i>HT</i>	<i>MF</i>
	<b>0.63</b>	0.26	0.11
TRUE (T)	<i>MT</i>	<i>T</i>	<i>HT</i>
	0.44	0.22	0.22

Table 4: Top 3 predictions and respective proportions for pass@3 results from MISTRAL-7B-V0.3. Highlighted cells indicate exact match between gold and predicted.

dicts. MOSTLYTRUE claims see the highest confidence predictions, with 63% of cases correctly identified. The pipeline shows some conservative tendency for TRUE claims, more frequently predicting MOSTLYTRUE (44%) than TRUE (22%). However, for HALFTRUE claims, the pipeline is more likely to predict MOSTLYTRUE (40%) than HALFTRUE (22%), and often gets confused with MOSTLYFALSE as well—possibly due to the ambiguity of the verdict itself.

Model	Top misclass.	Top fallacy and count
MISTRAL	F $\rightarrow$ MF (20)	causal oversimplification (6) hasty generalization(3)
GEMINI	HT $\rightarrow$ MT (14)	hasty generalization (3) causal oversimplification (3)

Table 5: Top misclassifications and top 2 fallacy counts for pipeline pass @3, excluding unverifiable

While the results of our approach are promising, they are still relatively low even for this harder task of 5-way classification. To further understand the possible sources of misclassifications in our fact-checking pipeline, we conducted an analysis on a subset of misclassified claims to assess whether they contain some types of fallacies. Our approach involved a two-step process: first, we prompted MISTRAL-7B-V0.3 to generate an open-ended fallacy label for each misclassified claim, based on the general definition of fallacy. Subsequently, these open-ended descriptions were categorized

into a predefined set of fallacy types using a separate classification model, also using MISTRAL-7B-V0.3, drawing categories sourced from [Alhindi et al. \(2023\)](#). The prompts are detailed in Appendix C. This approach allowed us to identify and quantify fallacies present in claims, potentially explaining the pipeline’s difficulties in correct factuality labeling.

Table 5 presents the top two types of misclassifications observed for two different models for the results for pipeline pass@3, using MISTRAL-7B-V0.3 and GEMINI-1.5-FLASH, along with the fallacy most frequently associated with each. We excluded "unverifiable" labels from this analysis for clarity. Pipeline pass@3 is chosen for this analysis, since it is our best performing ablation by MSE and Cohen’s  $\kappa$ .

When a claim was fundamentally FALSE, the MISTRAL pipeline frequently identified some fallacy but cautioned against a full refutation, labeling it as MOSTLYFALSE, which suggests a challenge in fully debunking complex or subtly flawed false claims. The most prominent fallacies associated with this error are causal oversimplification (6 instances) and hasty generalization (3 instances). This pattern indicates that MISTRAL struggles to refute claims where a simplified or incorrect cause-and-effect relationship is presented, or where broad, unsupported conclusions are drawn from insufficient evidence, which may give rise to some false sense of plausibility to an otherwise false claim, making it difficult for the pipeline to reach a definitive FALSE label.

Interestingly, for GEMINI, the most common error pipeline pass@3 makes is classifying HALFTRUE claims as MOSTLYTRUE, indicating a tendency for GEMINI to be "overly" permissive in its truth assessment. The top associated fallacies in these cases are hasty generalization (3 instances) and causal oversimplification (3 instances). This suggests that when a claim is partially true but then extends that truth to an unwarranted broad conclusion (hasty generalization) or draws an incorrect causal link (causal oversimplification), GEMINI pipeline may tend to prioritize the verifiable true components and fail to penalize the flawed logical step, causing an inflated truth assessment.

## 6 Conclusion and Future Work

We introduced a challenging benchmark, FACT5, for fine-grained, nuanced fact-checking. By in-

troducing this benchmark, we hope to contribute to the development of more nuanced evaluation frameworks that move beyond binary classification. Our five-class ordinal scale and emphasis on complex, multi-hop reasoning requirements could serve as a foundation for future benchmarks that better capture the complexity of real-world fact-checking scenarios. The dataset’s temporal recency and careful source attribution also address important considerations around data contamination and evaluation validity that should be standard features of fact-checking benchmarks moving forward.

The improved performance of our pipeline over the baseline models, particularly with smaller open-source models such as MISTRAL-7B-V0.3 model, is encouraging. Furthermore, the pass@3 performance of our pipeline indicates that it can be used in a semi-automated setting, where the system provides multiple ranked verdicts for human review. The verdict-wise analysis suggests that our system is particularly adept at identifying clearly false information. The additional analysis of the presence of fallacies in claims helps identify the harder cases for current models. While a broader systematic review—ideally with human annotators—is needed, it is important to understand how fallacies can cause models to either understate (e.g., MISTRAL) or inflate (e.g., GEMINI) the factuality of a statement. Further work could potentially add fallacy classification as part of the factuality assessment to understand if it leads to more accurate classifications.

## Limitations

While our research advances the automated fact-checking of complex statements using LLMs, some limitations need to be carefully considered.

**Dataset Size Limitations** Our current evaluation relies on our own curated FACT5 dataset of 150 statements, which represents a relatively small sample size compared to other NLP benchmarks. Though our pipeline shows promising results on the FACT5 dataset, its performance on a broader range of statement types and domains remains to be fully validated. The current evaluation, while thorough, may not capture all edge cases or statement complexities that could arise in real-world fact-checking scenarios. To mitigate these limitations, we focused on high-quality, professionally fact-checked statements, ensured balanced representation across truthfulness categories, selected



temporally relevant statements to test model performance on current claims, and incorporated multiple evaluation metrics for a comprehensive performance analysis. Future work could focus on expanding this benchmark dataset while maintaining these quality standards through collaboration with professional fact-checking organizations to access larger pools of verified claims or by potentially exploring semi-automated data collection using web-browsing agents (Costabile et al., 2025) or advanced synthetic data generation methods (Chung et al., 2025; Tang et al., 2024; Li et al., 2023). The FACT5 dataset’s recency (2024-2025) presents both advantages and limitations. While it allows testing of models’ capabilities on current events, it also means that the dataset might become less relevant over time as the context of these statements changes, or if the training cutoff date for language models gets extended to incorporate more recent web data. This temporal dependency could affect the long-term utility of both the dataset and the evaluation metrics derived from it.

**Pipeline Robustness** The sequential nature of our pipeline means that errors in early stages (e.g., atomic claim decomposition or question generation) can propagate through the system and affect final verdicts. While our results show strong overall performance, the interdependence of pipeline components could make the system vulnerable to cascading failures, particularly when dealing with especially complex or nuanced statements. For example, the retrieval of top-k results from search engines serves as a fundamental component of our fact-checking pipeline. However, the inherent challenge lies in the lack of a definitive method to ensure the accuracy of the retrieved information. Despite prioritizing reliable sources and performing rigorous post-processing, the inherent accuracy of the information obtained cannot be guaranteed.

**Need for Expert Human Evaluation** While we conducted a preliminary evaluation with participants lacking specialized knowledge in fact-checking, the results demonstrated limited value. The overwhelming agreement between untrained participants and model outputs suggests that this method may not provide sufficiently discriminative or insightful feedback for our purposes. Consequently, we have chosen to focus our analysis on more informative metrics detailed in §4.2, which are better suited to assess the performance of our five-way classification task for truthfulness.

Nonetheless, as noted in Russo et al. (2023), evaluating the quality of model reasoning is also crucial beyond examining the correctness of the model outputs. Therefore, for future work, expert evaluators in journalistic fields or who work as fact-checkers would be necessary for conducting a robust human evaluation.

**Challenges in Data Leakage** The absence of publicly accessible training data restricts our ability to explore the phenomenon of information memorization by LLMs for fact-checking purposes. Despite efforts to mitigate bias by blacklisting certain sources like PolitiFact, the ubiquity of its work across online content poses a challenge. Even if PolitiFact itself is excluded from the training data, its findings may still indirectly influence other sources used during the retrieval process, potentially impacting the reliability of the fact-checking outcomes.

**Political Biases and Logical Fallacies** Previous work has exhibited that political leanings can be embedded into LLMs (Feng et al., 2023). Due to the nature of our research, it is possible that LLMs exhibited political biases when determining the factuality of a statement, which could diverge from the nonpartisan nature of fact-checking tasks. A closer look at the results is needed to verify the presence of potential political biases in judging the factuality of statements. It is also important to note that although our model is primarily designed to fact-check complex statements, it is not yet equipped to identify common fallacies that are often deployed in political speeches (e.g., red herring and straw man fallacies).

## Ethics Statement

Our work aims to contribute positively to the challenge of combating misinformation by introducing a more nuanced approach to automated fact-checking and by developing tools that can be more accessible. However, we acknowledge several ethical considerations and potential risks associated with the development and deployment of such a system. While our pipeline demonstrates promising capabilities for assisted fact-checking, the deployment of our system could inadvertently contribute to reduced trust in legitimate news when the model makes incorrect classifications. Our pipeline’s reliance on web-retrieved data also raises concerns regarding the handling of copyrighted

material and user privacy. Furthermore, inherent biases in LLMs, including potential political leanings, might influence factuality assessments, and the current system is not designed to identify all types of logical fallacies, which can be prevalent in misleading statements.

To mitigate these risks, we strongly advocate for human oversight in any practical deployment, especially for sensitive claims. We have designed the pipeline with transparency in mind, including source citation, to aid such oversight. Future development should prioritize robust data minimization practices, clear protocols for copyrighted content, and the integration of privacy-preserving techniques. Continued research into debiasing LLMs and enhancing fallacy detection will also be crucial.

Ultimately, by highlighting these challenges and proposing a more granular approach to fact-checking, we hope to contribute to the development of more responsible and effective AI systems for combating misinformation and supporting critical information consumption.

## References

- Tariq Alhindi, Tuhin Chakrabarty, Elena Musi, and Smaranda Muresan. 2023. Multitask instruction-based prompting for fallacy recognition. *arXiv preprint arXiv:2301.09992*.
- Enrique Amigó, Julio Gonzalo, Stefano Mizzaro, and Jorge Carrillo-de Albornoz. 2020. An effectiveness metric for ordinal classification: Formal properties and experimental results. *arXiv preprint arXiv:2006.01245*.
- Simone Balloccu, Patrícia Schmidová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Lawrence W Barsalou. 1982. Context-independent and context-dependent information in concepts. *Memory & cognition*, 10(1):82–93.
- Jaime S Cardoso and Ricardo Sousa. 2011. Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(08):1173–1195.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.
- Yi-Ling Chung, Aurora Cobo, and Pablo Serna. 2025. [Beyond translation: Llm-based data generation for multilingual fact-checking](#).
- Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.
- Luigia Costabile, Gian Marco Orlando, Valerio La Gatta, and Vincenzo Moscato. 2025. [Assessing the potential of generative agents in crowdsourced fact-checking](#).
- Alexandra De Raadt, Matthijs J Warrens, Roel J Bosker, and Henk AL Kiers. 2019. Kappa coefficients for missing data. *Educational and psychological measurement*, 79(3):558–576.
- Jacob Devasier, Rishabh Mediratta, Akshith Putta, and Chengkai Li. 2025. Automatic fact-checking with frame-semantics. *arXiv preprint arXiv:2501.13288*.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvassy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. [The faiss library](#).
- Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. *arXiv preprint arXiv:2011.05448*.
- Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair nlp models. *arXiv preprint arXiv:2305.08283*.
- Lisa Gaudette and Nathalie Japkowicz. 2009. Evaluation methods for ordinal classification. In *Advances in Artificial Intelligence: 22nd Canadian Conference on Artificial Intelligence, Canadian AI 2009 Kelowna, Canada, May 25-27, 2009 Proceedings 22*, pages 207–210. Springer.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.
- Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.
- Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in llm fact verification. *arXiv preprint arXiv:2406.20079*.
- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

- Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. 2017. Claimbuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.
- Albert Jiang, Alexandre Sablayrolles, Alexis Tacnet, Antoine Roux, Arthur Mensch, Audrey Herblin-Stoop, Baptiste Bout, Baudouin de Monicault, Blanche Savary, Bam4d, Caroline Feldman, Devendra Singh Chaplot, Diego de las Casas, Eleonore Arcelin, Emma Bou Hanna, Etienne Metzger, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Harizo Rajaona, Jean-Malo Delignon, Jia Li, Justus Murke, Louis Martin, Louis TERNON, Lucile Saulnier, L elio Renard Lavaud, Margaret Jennings, Marie Pellat, Marie Torelli, Marie-Anne Lachaux, Nicolas Schuhl, Patrick von Platen, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Thibaut Lavril, Timoth e Lacroix, Th eophile Gervet, Thomas Wang, Valera Nemychnikova, William El Sayed, and William Marshall. 2024. *Mistral-7b-v0.3*.
- AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. 2023. *Mistral 7b (2023)*. *arXiv preprint arXiv:2310.06825*.
- O. Khattab, K. Santhanam, X. L. Li, D. Hall, P. Liang, C. Potts, and M. Zaharia. 2022. *Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp*. *arXiv preprint arXiv:2212.14024*.
- O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, and C. Potts. 2023. *Dspy: Compiling declarative language model calls into self-improving pipelines*. *arXiv preprint arXiv:2310.03714*.
- Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. 2019. Spoc: Search-based pseudocode to code. *Advances in Neural Information Processing Systems*, 32.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K uttler, Mike Lewis, Wen-tau Yih, Tim Rock-t aschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. *Synthetic data generation with large language models for text classification: Potential and limitations*.
- Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, and Shu Wu. 2023. Ex-fever: A dataset for multi-hop explainable fact verification. *arXiv preprint arXiv:2310.09754*.
- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.
- Ollama. 2024. Ollama. <https://github.com/ollama/ollama>.
- Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346*.
- Mark Rothmel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. Infact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112.
- Daniel Russo, Serra Sinem Tekiro lu, and Marco Guerini. 2023. Benchmarking the generation of fact checking explanations. *Transactions of the Association for Computational Linguistics*, 11:1250–1264.
- Tetsuya Sakai. 2021. Evaluating evaluation measures for ordinal classification and ordinal quantification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2759–2769.
- Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2024. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36.
- Liyan Tang, Philippe Laban, and Greg Durrett. 2024. *Minicheck: Efficient fact-checking of llms on grounding documents*.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22.
- Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR ’21*, page 317–324, New York, NY, USA. Association for Computing Machinery.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. *Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers*.

- William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.
- Yuxia Wang, Revanth Gangi Reddy, Zain Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, et al. 2024. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230.
- Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024a. A closer look at claim decomposition. *arXiv preprint arXiv:2403.11903*.
- Miriam Wanner, Benjamin Van Durme, and Mark Dredze. 2024b. Dndscore: Decontextualization and decomposition for factuality verification in long-form text generation. *arXiv preprint arXiv:2412.13175*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, et al. 2024. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*.
- Ayfer Ezgi Yilmaz and Haydar Demirhan. 2023. Weighted kappa measures for ordinal multi-class classification performance. *Applied Soft Computing*, 134:110020.
- Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*.

## A FACT5 Pilot Dataset

### A.1 Explanation of Ratings

Label	Description
True	The statement is accurate, and there's nothing significant missing.
Mostly True	The statement is accurate but needs clarification or additional information.
Half True	The statement is partially accurate but leaves out important details or takes things out of context.
Mostly False	The statement contains an element of truth but ignores critical facts that would give a different impression.
False	The statement is not accurate.
Pants on Fire	The statement is not accurate and makes a ridiculous claim.

Table 6: Truth-O-Meter rating used by PolitiFact

### A.2 Mapping of Fact-checking Metrics

Numeric Label	Our Mapping	Credible Sources		
		PolitiFact	The Fact Checker (WaPo)	Snopes
0	Unverifiable	N/A	No Verdict	Unproven / Unfounded
1	False	Pants on Fire / False	Four Pinocchios	False
2	Mostly False	Mostly False	Three Pinocchios	Mostly False
3	Half True	Half True	Two Pinocchio	Mixture
4	Mostly True	Mostly True	One Pinocchio	Mostly True
5	True	True	The Geppetto Checkmark	True

Table 7: Mappings of Fact-checking Metrics

### A.3 Snippet of the Pilot Dataset

verdict	statement_originator	statement	statement_date	factchecker
FALSE	Joe Biden	"Remember in 2020, 55 of the biggest companies in America made 40 billion and paid zero in federal income taxes. [...]"	3/7/2024	CNN
MOSTLY FALSE	Elissa Slotkin	"[...] Mike Rogers 'believes he should make that decision' about whether to end pregnancies."	9/30/2024	PolitiFact
HALF TRUE	Dana Loesch	"Buncombe County 'is still demanding property taxes on homes destroyed by Hurricane Helene [...]"	1/6/2025	PolitiFact
MOSTLY TRUE	Ron DeSantis	"[Nikki Haley] spent 100% of her money attacking me..."	1/26/2024	CNN
TRUE	David Crowley	"Under [Biden] [...] fastest growth of Black-owned small businesses in more than 30 years."	5/16/2024	PolitiFact

Table 8: Selected Rows and Columns from Pilot Dataset

## B Pipeline Details

### B.1 Atomic Claim Generation

DSPy signature for claim extraction, which consists of the criteria for each claim.

```
"""Extract specific claims from the given statement.
1. Split the statement into multiple claims, but if the statement is atomic (has
one main claim), keep it as is.
2. If context is included (e.g., time, location, source/speaker who made the
statement, etc.), include the context in each claim to help verify it. Do not
make up a context if it is not present in the text.
3. Consider the source (e.g. name of the speaker, organization, etc.) and date
of the statement if given in the context, and include them in each claim.
4. Each claim should be independent of each other and not refer to other claims.
5. Always extract claims regardless of the content """
```

Output field

```
"""JSON object containing:
{
  "claims": [
    {
      "text": string, }
  ]
}"""
```

### B.2 Question Generation

DSPy signature for question generation

```
"""Break down the given claim derived from the original statement to generate
independent questions and search queries to answer it. Be as specific and concise
as possible, try to minimize the number of questions and search queries while
still being comprehensive to verify the claim."""
```

Output field

```
"""JSON object containing: {
  "questions": [
    {
      "question": string, # question text (e.g. "What was the GDP growth rate during
the Trump administration?")
      "search_queries": [string], # independent search queries used to answer the
question, try to be as specific as possible and avoid redundancy, 1-2 queries
is ideal
    }
  ]}"""
```

### B.3 Answer Synthesis

DSPy signature for answer synthesis

```
"""Synthesize an answer based on retrieved documents with inline citations."""
```

Output field

```
"""JSON object containing:
{
  "text": string, # answer with inline citations where the number in the brackets
is the index of the citation in the citations list (e.g., "The wage gap was
shrinking [1]")
  "citations": [{ # list of citations
  "snippet": string, # exact quote from source
  "source_url": string,
  "source_title": string,
  "relevance_score": float
}]
}"""
```



## C Fallacy Classification Prompt

### C.1 Open-ended fallacy detection

DSPy signature for open-ended fallacy detection

```
"""Classify logical fallacies given the statement"""  
Input field statement = "Statement to analyze"
```

Output field

1. **fallacy** (str) = "A fallacy or a fallacious argument is one that seems valid but is not. Identify the fallacy in the statement. If no fallacy is present, return 'none'."
2. **confidence** (float) = "0-1 confidence score"
3. **rationale** (str) = "Step-by-step reasoning"

### C.2 Fallacy Categorization

Predefined categories, from [Alhindi et al. \(2023\)](#)

```
'ad hominem', 'appeal to emotion', 'hasty generalization', 'irrelevant authority',  
'red herring', 'black and white fallacy', 'causal oversimplification', 'doubt',  
'exaggeration or minimization', 'appeal to fear/prejudice', 'flag-waving',  
'loaded language', 'name calling or labeling', 'reductio ad hitlerum',  
'slogans', 'strawman', 'thought-terminating cliches', 'whataboutism', 'ad  
populum', 'circular reasoning', 'deductive fallacy', 'equivocation', 'fallacy  
of extension', 'intentional fallacy', 'evading burden of proof', 'cherry-picking',  
'post hoc (causal oversimplification)', 'vagueness', 'none'
```

DSPy signature for fallacy categorization

```
"""Categorize an open-ended fallacy description into a predefined list of fallacy  
types.If the detected fallacy does not clearly fit into any of the predefined  
categories,classify it as 'Other'."""
```

1. **open\_ended\_fallacy** (str) = "The name or description of the fallacy detected by an open-ended system (e.g., 'This is an ad hominem because...', or 'Attacking the person instead of the argument', or 'none')."
2. **target\_categories** List[str] = A list of predefined fallacy categories to map the detected fallacy into.

Output field

1. **categorized\_fallacy**: (str) The category from the target\_categories list that best matches the detected fallacy. If the open\_ended\_fallacy is 'none' or doesn't fit any category, return 'Other' or 'None Detected' as appropriate."
2. **confidence**: (float) "0-1 confidence score for this categorization."
3. **rationale**: (str) = "Step-by-step reasoning for choosing the category, or for choosing 'Other'/'None Detected'."