Fact-Checking with Large Language Models via Cost-Effective First-Order Logic Reformulation

Sara Asghari[1], Laks V.S.Lakshmanan[2], Venkatesh Srinivasan[3], and Alex Thomo[1]

[1] University of Victoria, Victoria, BC V8P 5C2, Canada
`saraasghari@uvic.ca,thomo@uvic.ca`
[2] University of British Columbia, Vancouver, BC V6T 1Z4. Canada
`laks@ubc.ca`
[3] Santa Clara University, Santa Clara, CA 95050, USA
`vsrinivasan4@scu.edu`

Abstract. The rise of misinformation on digital platforms creates a pressing need for robust fact-checking (FC) methods. While large language models (LLMs) have achieved remarkable success in natural language processing, fact-checking complex claims remains a critical challenge. Current state-of-the-art approaches, such as ProgramFC and FOLK, suffer from significant limitations, including inefficiency, high costs, and restricted expressiveness in reasoning. In this work, we propose **FOLDCoVe**, a novel method that addresses these challenges. FOLDCoVe employs a dual-LLM framework, leveraging a powerful LLM to translate claims into First-Order Logic (FOL) and a cost-effective LLM to verify predicates against lengthy sources. Our approach significantly outperforms state-of-the-art methods on the HOVER multi-hop reasoning benchmark in terms of fact-checking accuracy, while offering concise prompts and interpretable outputs, making it accessible to non-technical users.

Keywords: Fact Checking, First Order Logic, Large-Language Models, Social and Media Analysis

1 Introduction

The proliferation of misinformation on digital platforms underscores the critical need for robust fact-checking mechanisms [13, 7]. Unlike traditional reading comprehension tasks involving straightforward question answering [1], real-world fact-checking requires aggregating information from multiple sources and applying complex, multi-step reasoning [12, 16]. For instance, verifying the (false) claim, “Elon Musk and the CEO of Facebook both left Harvard University without completing their degrees,” necessitates breaking down the claim, retrieving information about both individuals and combining this evidence to reach a conclusion.

In recent years, large language models (LLMs) have revolutionized the field of natural language processing (NLP), achieving remarkable milestones in tasks such as language translation, text generation, and sentiment analysis [5, 18]. However, the task of fact-checking—the ability to verify the accuracy of complex claims against diverse and often challenging evidence—remains a critical yet underexplored domain [11, 8].
Two notable approaches to address the complexity of fact-checking include ProgramFC and FOLK. ProgramFC [8] decomposes complex claims into smaller sub-tasks, each solved using specific functions. It uses LLMs to generate a reasoning program, which guides the process by assigning sub-tasks like answering questions or verifying simpler claims to appropriate handlers. FOLK [14], on the other hand, translates a claim into a first-order logic (FOL) clause made up of predicates, each representing a sub-claim to be verified. It then grounds the predicates in real-world knowledge from reliable sources like Google or Wikipedia. FOLK then uses LLMs to evaluate the FOL predicates and make a veracity prediction.
While ProgramFC and FOLK represent significant advancements in LLM-based fact-checking, both have notable limitations. ProgramFC relies on lengthy prompts (200+ lines) structured as reasoning programs in Python-like syntax, which can alienate non-technical users and increase inefficiency and monetary cost (for API access). Managing these prompts and handling intermediate results requires in practice a driver program, making it impractical for casual LLM users. FOLK, on the other hand, simplifies claims into first-order logic predicates but assumes a conjunction of predicates, limiting its ability to handle complex constructs like disjunctions or quantifiers. Its reliance on a single LLM for both decomposition and veracity check can also incur high costs, as grounding each predicate against lengthy sources is resource-intensive despite the brevity of most claims.
To overcome these limitations, we introduce **FOLDCoVe**,[4] a method that leverages full power of FOL — including quantifiers and disjunctions —and employs a two-LLM framework. A powerful LLM translates claims into concise FOL representations, and a cost-effective LLM verifies these against lengthy sources. Unlike existing methods, FOLDCoVe uses concise prompts that are accessible to casual users and eliminate the need to manage intermediate results. As illustrated in Figure 1, our method translates the claim into FOL, restructures it back into natural language, and then verifies it using a weaker LLM. Translating the claim into FOL is necessary because claims are often written in a terse, informal style that is not easily verifiable against sources. Users typically write short, concise claims that reflect their immediate thoughts and may not want to expand them into longer, detailed statements. After translating the claim into FOL, we convert it back into a natural, human-readable form (anglicization) to make it more suitable for the subsequent verification step by a weaker LLM. In contrast to ProgramFC and FOLK, the weaker LLM in our method assesses the restructured claim in a single step without verifying each sub-claim separately. Our approach allows for efficient

[4] **F**irst-**O**rder-**L**ogic **D**ecomposition and **Co**st-Effective Claim **Ve**rification.

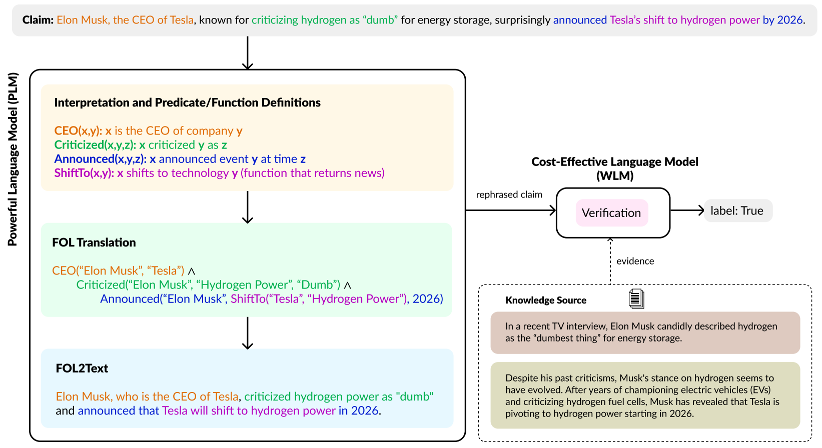
Claim: Elon Musk, the CEO of Tesla, known for criticizing hydrogen as "dumb" for energy storage, surprisingly announced Tesla's shift to hydrogen power by 2026.
Powerful Language Model (PLM)
Interpretation and Predicate/Function Definitions
CEO(x,y): x is the CEO of company y
Criticized(x,y,z): x criticized y as z
Announced(x,y,z): x announced event y at time z
ShiftTo(x,y): x shifts to technology y (function that returns news)
FOL Translation
CEO("Elon Musk", "Tesla") ∧
Criticized("Elon Musk", "Hydrogen Power", "Dumb") ∧
Announced("Elon Musk", ShiftTo("Tesla", "Hydrogen Power"), 2026)
FOL2Text
Elon Musk, who is the CEO of Tesla, criticized hydrogen power as "dumb" and announced that Tesla will shift to hydrogen power in 2026.
rephrased claim
Cost-Effective Language Model (WLM)
Verification
label: True
evidence
Knowledge Source
In a recent TV interview, Elon Musk candidly described hydrogen as the "dumbest thing" for energy storage.
Despite his past criticisms, Musk's stance on hydrogen seems to have evolved. After years of championing electric vehicles (EVs) and criticizing hydrogen fuel cells, Musk has revealed that Tesla is pivoting to hydrogen power starting in 2026.

Fig. 1: FOLDCoVe: A powerful model (PLM) decomposes the complex claim into first-order logic (FOL), restructures it back into natural language, and a weaker model (WLM) verifies the restructured claim against knowledge sources to judge its veracity.

verification of complex claims without the overhead of lengthy prompts or driver programs.
In summary, our contributions are:

- We propose **FOLDCoVe**, a novel method for fact-checking complex claims by leveraging the full power of First-Order Logic to enhance the expressiveness and reasoning capabilities of LLM-based systems.
- Our dual-LLM framework employs a powerful model for translating claims into First-Order Logic formulations and a cost-effective model for verification against lengthy sources, reducing computational and monetary costs.
- We demonstrate that FOLDCoVe outperforms state-of-the-art methods (ProgramFC and FOLK) on the HOVER multi-hop reasoning dataset [3], while providing concise prompts and interpretable outputs accessible to non-technical users.

2 Related Work

Fact-checking has emerged as a critical area of research in natural language processing (NLP) due to the widespread dissemination of misinformation on digital platforms [13, 7]. Early efforts in automated fact-checking focused on verifying simple claims against single evidence sources, as exemplified by the FEVER dataset [12]. However, real-world fact-checking often involves complex claims requiring reasoning over multiple pieces of evidence. To address this, the HOVER (Hop-Over Dataset) [3] was introduced as an alternative to FEVER, necessitating multi-hop reasoning over multiple documents and reflecting the challenges encountered in practical fact-checking scenarios.

Traditional reading comprehension tasks [1, 9] involve answering questions based on a given text but generally do not require synthesizing information from multiple sources or performing complex reasoning. In contrast, fact-checking demands aggregating evidence from diverse documents and applying logical reasoning to assess a claim's veracity [17].

Large Language Models in Fact-Checking. Large language models have achieved significant results in tasks like text generation and question answering, but applying them to fact-checking remains challenging due to issues like hallucinations and the lack of interpretable reasoning processes [2]. Efforts to mitigate these issues include integrating external knowledge sources and enhancing reasoning capabilities within LLMs [4, 10].

Decomposition Methods and Logic-Based Approaches. To handle the complexity of verifying multifaceted claims, decomposition strategies and logic-based methods have been employed. Chain-of-thought prompting [15] allows LLMs to generate intermediate reasoning steps, enhancing their ability to solve multi-step problems. **ProgramFC** [8] decomposes complex claims into sub-tasks using Python-like reasoning programs. While effective, it relies on lengthy prompts and a driver program to handle intermediate results, increasing computational costs and usability challenges. Similarly, **FOLK** [14] translates claims into first-order logic (FOL) clauses and grounds predicates using reliable sources like Wikipedia. However, it does not support constructs like disjunctions and quantifiers, limiting its applicability to more complex claims. Additionally, relying on a single LLM for both decomposition and verification can be resource-intensive when processing lengthy evidence. Additionally, predicate checks are treated independently, with no carry-over of values between checks, causing ambiguities and interpretative challenges, besides making the checks expensive.

Despite these advancements, challenges persist in handling real-world fact-checking complexity efficiently. Our method, **FOLDCoVe**, tackles these issues by leveraging first-order logic with quantifiers and disjunctions and employing a dual-LLM framework for optimized performance and accessibility.

3 Methodology

We begin by reviewing evidence availability scenarios, followed by an introduction to FOLDCoVe, our novel approach to automated fact-checking.

3.1 Evidence Availability Scenarios

Fact-checking tasks differ by evidence availability [8], with three primary settings: **Gold-Evidence**, **Closed-Book**, and **Open-Book**. In **Gold-Evidence**, all relevant documents or their fragments are provided with the claim. **Closed-Book** setting requires the model to rely solely on its internal knowledge. **Open-Book** setting provides the model with a document collection from which it must retrieve relevant evidence. HOVER claims are based on Wikipedia, a key source for LLM training. With LLMs storing knowledge in a 'compressed' form, access to

Wikipedia in an Open-Book setup makes the task easier, while Closed-Book, relying solely on internal knowledge, is more challenging. To analyze domain-specific reasoning, we introduce the **Contextual-Closed-Book (CCB)** variant, where the model is guided to focus on a specific domain (e.g., Wikipedia) in its internal knowledge. This paper focuses on the **Gold-Evidence**, **Closed-Book**, and **Contextual-Closed-Book** settings, with the latter explored via an ablation study.

3.2 FOLDCoVe

We introduce **FOLDCoVe** (**F**irst-**O**rder-**L**ogic **D**ecomposition and **Co**st-**Ve**rification), a framework designed to enhance fact-checking of complex claims by leveraging the expressive power of first-order logic (FOL) and the efficiency of a dual-LLM approach. FOLDCoVe addresses the limitations of existing methods by fully utilizing FOL—including quantifiers and disjunctions—and employing a cost-effective verification process. An overview of our method is illustrated in Figure 1. The framework consists of two main components: *Claim Decomposition* and *Verification*.

Claim Decomposition The first step is to decompose complex claims into simpler, verifiable components by translating them into first-order logic (FOL) using a powerful language model (PLM) like GPT-4. PLM performs:

1. **Interpretation and Predicate Definition**: Identifies key entities, relationships, and concepts, defining FOL predicates.
2. **FOL Translation**: Translates claim into FOL formula that captures its logical structure.
3. **FOL2Text**: Converts FOL formula back into structured text to ease verification.

We employ a specific prompt to guide the PLM in this process, ensuring precision and avoiding common pitfalls.

Prompt Engineering for FOL Decomposition Prompt Engineering is the practice of developing and optimizing prompts — text-based inputs provided to large language models (LLMs) to guide their outputs [6]. The prompt used in FOLDCoVe for breaking down the claim into sub-claims is as follows:

```
Convert the input to first-order logic. Your output should consist
of a set of predicates and a logical formula that combines them.
- Never miss a constant (entity) in the input that contributes
to the formula.
- Replace variables with constants whenever possible.
- Avoid unary predicates solely for modeling constants.
- Avoid unused variables in the formula.
After producing the formula, anglicize it.
```

```
Example: {
Input: The logo of the American athletic shoe and clothing manufacturer
who trademarked a slogan coined by Dan Wieden has made $26 billion
in profit.

Predicates:
M(x): x is an American athletic shoe and clothing manufacturer.
S(x, y): x trademarked the slogan y.
C(y, z): Slogan y was coined by z.
P(x, w): The logo of the manufacturer x has made w profit.

Formula: Exists x, y (M(x) and S(x, y) and C(y, 'Dan Wieden') and
P(x, '$26 billion'))

Anglicized: There exists an American athletic shoe and clothing
manufacturer, let's call it 'X', which has trademarked a slogan,
let's call it 'Y'. This slogan 'Y' was coined by Dan Wieden. Additionally,
the logo of this manufacturer 'X' has generated a profit of $26
billion. }

Input: <target_claim>
```

Illustrative Example. Consider the claim: *"Elon Musk and the CEO of Facebook both left Harvard University without completing their degrees."*

Applying the FOLDCoVe framework:

1. **Interpretation and Predicate Definition**:
   - $A(x, y)$: Person x attended university y.
   - $L(x, y)$: Person x left university y without completing their degree.
   - $C(x, y)$: Person x is the CEO of company y.
   - Constants: 'Elon Musk,' 'Harvard University,' 'Facebook.'
2. **FOL Translation**:
   $A(\text{'Elon Musk'}, \text{'Harvard University'}) \wedge$
   $L(\text{'Elon Musk'}, \text{ 'Harvard University'}) \wedge$
   $\exists z \;\; A(z, \text{ 'Harvard University'}) \wedge$
   $L(z, \text{ 'Harvard University'}) \wedge$
   $C(z, \text{ 'Facebook'})$
3. **FOL2Text**: "Elon Musk attended Harvard University. Elon Musk then left Harvard University without completing his degree. There exists a person z who attended Harvard University, left without completing their degree, and is the CEO of Facebook."

Verification After decomposing and restructuring the claim, the next step is verification. A weaker, cost-effective language model (WLM), such as Mixtral or Llama, is used to assess the claim's veracity. The WLM evaluates the restructured claim using its internal knowledge and, if available, external sources of evidence like Wikipedia. It determines whether the claim is *Supported*, *Refuted*, or if there is *Not Enough Information* (NEI), and provides a brief explanation of its judgment. Processing the restructured claim in a single step (as opposed to processing each predicate separately as in FOLK) reduces computational overhead and simplifies verification.

Verification Example. The WLM evaluates the restructured claim:
Assessment: Elon Musk did not attend Harvard University; he attended the University of Pennsylvania and Queen's University. Mark Zuckerberg, the CEO of Facebook, left Harvard University without completing his degree.
Veracity Judgment: *Refuted*
Justification: The claim is false because Elon Musk did not attend Harvard University.

Rationale for the Dual-LLM Framework The dual-LLM framework provides distinct advantages in cost efficiency, reasoning, and accessibility. Limiting the use of PLM to claim decomposition reduces overall costs, as verification tasks against lengthy sources are delegated to a cost-effective WLM. The PLM handles complex language and logical structures, while the WLM verifies claims in structured natural language. Additionally, concise and clear prompts and outputs eliminate the need for complex driver programs, making the method accessible to non-technical users.

4 Experiments

4.1 Dataset

We evaluated our method using the HOVER (Hop-Over Dataset) [3], a large-scale dataset simulating real-world fact verification tasks that require aggregating information from multiple sources. In HOVER, a **hop** refers to retrieving evidence from an additional Wikipedia article to verify a claim; thus, an n-hop claim requires at least n different articles, increasing reasoning complexity as n grows. We focused on the HOVER validation set, which contains 1,126 two-hop claims, 1,835 three-hop claims, and 1,039 four-hop claims.
Derived from HOTPOTQA [17], HOVER began with two-hop claims generated by crowd-workers from question-answer pairs. To create more complex three-hop and four-hop claims, entities in the two-hop claims were replaced with information from additional Wikipedia articles, increasing the complexity of the reasoning task. Each claim in HOVER includes the claim text, curated evidence collected by crowd-workers, and the ground-truth label indicating the claim's veracity.

4.2 Evaluation Metrics

In our experiments, various metrics were used to evaluate the model's performance. They include **Precision**, **Recall**, **Accuracy**, **F1 score**, and **F1 Macro Average**. As the first four metrics are well-known, we only define the last metric.

F1 Macro Average: F1 Macro Average calculates the F1 score by first computing the F1 score for each class individually and then averaging these scores across all classes. It gives equal weight to each class, making it especially useful when dealing with imbalanced datasets where the performance across all classes needs to be assessed fairly.

Furthermore, in fact-checking, where the goal is to predict whether claims are true or false — the performance metrics **precision-0**, **precision-1**, **recall-0**, **recall-1**, **F1-0**, and **F1-1** are used to evaluate the model's effectiveness for each class individually.

Approach	**Language Model**	**HOVER (2-Hop)**	**HOVER (3-Hop)**	**HOVER (4-Hop)**
ProgramFC		0.65	0.53	0.61
FOLK	GPT-4-Turbo + Mixtral-8x7B-Instruct	0.64	0.54	0.51
FOLDCoVe		0.67	0.61	0.64
ProgramFC		0.66	0.61	0.62
FOLK	GPT-4-Turbo + Llama-3-70B-Chat	0.42	0.43	0.46
FOLDCoVe		0.74	0.73	0.71

Table 1: F1-MacroAvg scores for FOLDCoVe and SOTA in Gold-Evidence setting, using Mixtral-8x7B-Instruct and Llama-3-70B-Chat as WLMs.

4.3 Main Results

We compared FOLDCoVe to SOTA methods–ProgramFC and FOLK– under Gold-Evidence and Closed-Book settings, with Contextual-Closed-Book (CCB) analyzed in Section 4.5. While FOLK typically relies on a single LLM, we used PLM for its FOL formulation and WLM for claim verification for fairness. Similarly, for ProgramFC, which employs multiple LLMs, we standardized the setup by using PLM for all formulation tasks and WLM for claim verification. This setup separates the core strengths of each method from system-dependent factors, such as LLM choice or fine-tuning. Each method has two distinct phases: formulating multiple prompts and verifying the claim, and our approach enables fair evaluation on equal footing.

Specifically, we employed GPT-4-Turbo as the PLM for claim processing. For verification, we varied the WLMs, utilizing Mixtral-8x7B-Instruct and Llama-3-70B-Chat as cost-effective options. Gold-Evidence results are in Table 1, and Closed-Book results in Table 2.

Approach	**Language Model**	**HOVER (2-Hop)**	**HOVER (3-Hop)**	**HOVER (4-Hop)**
ProgramFC	GPT-4-Turbo + Mixtral-8x7B-Instruct	0.58	0.50	0.49
FOLK		0.57	0.50	0.48
FOLDCoVe		0.63	0.58	0.55
ProgramFC	GPT-4-Turbo + Llama-3-70B-Chat	0.60	0.57	0.52
FOLK		0.46	0.45	0.47
FOLDCoVe		<u>0.70</u>	<u>0.61</u>	<u>0.58</u>

Table 2: F1-MacroAvg scores for FOLDCoVe and SOTA in the Closed-book setting, using Mixtral-8x7B-Instruct and Llama-3-70B-Chat as WLMs.

4.4 Observations

The results in Tables 1 and 2 reveal several key insights.

Performance. FOLDCoVe consistently outperforms both ProgramFC and FOLK across all hop levels (2-hop, 3-hop, and 4-hop) and settings (Gold-Evidence and Closed-Book). This shows the robustness and effectiveness of our method in handling complex multi-hop reasoning tasks.

Impact of Language Models. The choice of WLM significantly impacts performance. In the Gold-Evidence setting, using powerful models like GPT-4-Turbo as the WLM leads to the highest F1 scores for FOLDCoVe, achieving up to 0.78 on 2-hop claims. However, even when using less powerful and more cost-effective models like Mixtral-8x7B-Instruct, FOLDCoVe still surpasses the SOTA methods, highlighting its efficiency.

Higher Hop Levels. FOLDCoVe exhibits remarkable performance even as the number of hops increases. For instance, with Llama-3-70B-Chat as the WLM in the Gold-Evidence setting, FOLDCoVe achieves F1 scores of 0.74, 0.73, and 0.71 for 2-hop, 3-hop, and 4-hop claims respectively, significantly higher than both ProgramFC and FOLK.

Closed-Book Setting Challenges. In the Closed-Book setting, performance generally decreases due to the lack of external evidence. Despite this, FOLDCoVe maintains a lead over the SOTA methods, indicating its ability to leverage internalized knowledge effectively.

Benefits of Dual-LLM Framework. The consistent use of GPT-4-Turbo as the PLM ensures high-quality claim decomposition across all methods. By varying the WLM, we demonstrate that FOLDCoVe's verification process is both flexible and effective, capable of achieving high performance even with less powerful models.

4.5 Ablation Study on the Contextual-Closed-Book Setting

In addition to the Closed-Book (CB) and Gold-Evidence settings, we evaluated our method under the **Contextual-Closed-Book (CCB)** setting. This setting aims to direct the model's focus toward a specific domain—in our case, Wikipedia—by modifying the verification prompt to include the instruction:

"Use only Wikipedia knowledge."

We conducted an ablation study comparing the performance of our method using Mixtral-8x7B-Instruct and Llama-3-70B-Chat in the CCB setting versus the CB setting. The goal was to assess whether constraining the model's knowledge retrieval to Wikipedia would enhance its fact-checking performance.

Results and Analysis The results of the ablation study are presented in Table 3. The findings indicate that the impact of the CCB setting on model performance is mixed and not consistently beneficial across different models and tasks.

For Mixtral-8x7B-Instruct:

– On the **2-hop** dataset, the CCB setting led to slight improvements in metrics like Precision-0 (+5.97%), F1-1 (+4.84%), and overall F1-MacroAvg (+4.76%), but a minor decrease in Recall-0 (-1.59%).
– On the **3-hop** dataset, performance generally decreased in the CCB setting, with F1-MacroAvg dropping by 1.72%.
– On the **4-hop** dataset, there were improvements in Precision-0 (+3.64%) and Recall-1 (+11.32%), but also decreases in Recall-0 (-7.02%) and F1-0 (-1.79%).

For Llama-3-70B-Chat:

– On the **2-hop** dataset, Recall-0 improved by +9.86%, but Recall-1 decreased by -14.49%. The overall F1-MacroAvg decreased slightly by -1.43%.
– On the **3-hop** dataset, Recall-0 improved by +8.82%, but Recall-1 decreased by -10.91%. The overall F1-MacroAvg remained the same.
– On the **4-hop** dataset, Recall-0 improved by +11.11%, but Recall-1 decreased by -7.41%. The overall F1-MacroAvg increased by +3.45%.

Interpretation The mixed results suggest that adding the instruction to use only Wikipedia knowledge does not consistently enhance the models' performance. While there are some improvements in specific metrics, these are offset by declines in others. Notably, the changes are relatively small, and in some cases, the overall performance measured by F1-MacroAvg remains unchanged or shows minimal variation. This could indicate that the models are already effectively utilizing their internal knowledge bases, which are substantially trained on Wikipedia content. Therefore, explicitly directing them to focus on Wikipedia may not significantly alter their behavior. Additionally, the models might not need additional guidance to narrow their focus, as they inherently prioritize relevant information when making veracity judgments.

5 Limitations

While FOLDCoVe advances automated fact-checking by integrating first-order logic with language models, several limitations merit attention.
First, the effectiveness of FOLDCoVe hinges on the precise decomposition of complex claims into sub-claims using FOL. Although FOL provides a structured framework, not all real-world statements are easily or fully expressible in logical form. Ambiguous or context-dependent claims might lose critical subtleties when translated into FOL, leading to incomplete or inaccurate sub-claims and potentially erroneous verifications.
Second, in the Contextual-Closed-Book setting, the model is directed to focus on specific domains like Wikipedia. While this focus can sometimes improve performance within the targeted domain (see 4-hop results in Section 4.5), it can reduce the framework's effectiveness when handling information outside these areas.
Additionally, FOLDCoVe inherits any biases present in the underlying language models. These biases can influence fact-checking outcomes, potentially leading to unfair or skewed verifications. Addressing such biases is essential to ensure the system's impartiality and reliability.

6 Conclusion

We introduced **FOLDCoVe**, a framework for fact-checking complex claims using full first-order logic (FOL) and a cost-effective dual-language model approach. Decomposing claims into FOL with a powerful language model (PLM) and verifying them using a weaker, model (WLM), FOLDCoVe addresses the limitations of methods like ProgramFC and FOLK. Our framework introduced the concept of anglicizing FOL formulas, making the logical representation of claims more human-readable. Our experiments on the HOVER dataset showed that FOLDCoVe consistently outperforms SOTA across various settings and hop levels, highlighting the benefits of using full FOL and our dual-LLM framework.
The current implementation of **FOLDCoVe**, while promising, offers several avenues for further exploration. One potential direction is to refine the verification process. Instead of directly anglicizing the FOL formula, each predicate could be treated as a “sub-query" to be verified independently by the WLM. The verification results for each predicate could then be aggregated, potentially using a many-valued logic system that considers the probabilities or confidence scores associated with each sub-query. This approach could lead to a more nuanced and accurate assessment of the overall claim's veracity.
Additionally, this approach allows for the parallelization of the variable instantiation and knowledge grounding process. Since the verification of each sub-query is independent of the others, these steps could be performed in parallel, potentially leading to significant speedups in the overall fact-checking process. This would be particularly beneficial when dealing with large volumes of claims or claims with a large number of atomic propositions.

HOVER 2-hops Dataset

Metric	**Mixtral-8x7B-Instruct**			**Llama-3-70B-Chat**		
	CB	CCB	% Chg	CB	CCB	% Chg
Precision-0	0.67	0.71	+5.97%	0.72	0.69	-4.17%
Recall-0	0.63	0.62	-1.59%	0.71	0.78	+9.86%
F1-0	0.65	0.66	+1.54%	0.72	0.74	+2.78%
Precision-1	0.60	0.61	+1.67%	0.67	0.70	+4.48%
Recall-1	0.64	0.70	+9.38%	0.69	0.59	-14.49%
F1-1	0.62	0.65	+4.84%	0.68	0.64	-5.88%
Accuracy	0.63	0.66	+4.76%	0.70	0.70	0.00%
F1-MacroAvg	0.63	0.66	+4.76%	0.70	0.69	-1.43%

HOVER 3-hops Dataset

Metric	**Mixtral-8x7B-Instruct**			**Llama-3-70B-Chat**		
	CB	CCB	% Chg	CB	CCB	% Chg
Precision-0	0.55	0.54	-1.82%	0.58	0.57	-1.72%
Recall-0	0.63	0.58	-7.94%	0.68	0.74	+8.82%
F1-0	0.59	0.56	-5.08%	0.62	0.65	+4.84%
Precision-1	0.62	0.60	-3.23%	0.66	0.68	+3.03%
Recall-1	0.53	0.57	+7.55%	0.55	0.49	-10.91%
F1-1	0.57	0.58	+1.75%	0.60	0.57	-5.00%
Accuracy	0.58	0.57	-1.72%	0.61	0.61	0.00%
F1-MacroAvg	0.58	0.57	-1.72%	0.61	0.61	0.00%

HOVER 4-hops Dataset

Metric	**Mixtral-8x7B-Instruct**			**Llama-3-70B-Chat**		
	CB	CCB	% Chg	CB	CCB	% Chg
Precision-0	0.55	0.57	+3.64%	0.58	0.59	+1.72%
Recall-0	0.57	0.53	-7.02%	0.63	0.70	+11.11%
F1-0	0.56	0.55	-1.79%	0.61	0.64	+4.92%
Precision-1	0.55	0.55	0.00%	0.58	0.62	+6.90%
Recall-1	0.53	0.59	+11.32%	0.54	0.50	-7.41%
F1-1	0.54	0.57	+5.56%	0.56	0.55	-1.79%
Accuracy	0.55	0.56	+1.82%	0.58	0.60	+3.45%
F1-MacroAvg	0.55	0.56	+1.82%	0.58	0.60	+3.45%

Table 3: FOLDCoVe ablation study on HOVER datasets (Closed-Book vs. Contextual Closed-Book). The % Change columns represent the percentage change from CB to CCB settings for each metric.

References

1. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. Advances in neural information processing systems **28** (2015)
2. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Computing Surveys **55**(12), 1–38 (2023)
3. Jiang, Y., Bordia, S., Zhong, Z., Dognin, C., Singh, M., Bansal, M.: Hover: A dataset for many-hop fact extraction and claim verification. arXiv preprint arXiv:2011.03088 (2020)
4. Lazaridou, A., Gribovskaya, E., Stokowiec, W., Grigorev, N.: Internet-augmented language models through few-shot prompting for open-domain question answering. arXiv preprint arXiv:2203.05115 (2022)
5. Li, J., Tang, T., Zhao, W.X., Nie, J.Y., Wen, J.R.: Pre-trained language models for text generation: A survey. ACM Computing Surveys **56**(9), 1–39 (2024)
6. Marvin, G., Hellen, N., Jjingo, D., Nakatumba-Nabende, J.: Prompt engineering in large language models. In: International conference on data intelligence and cognitive informatics. pp. 387–402. Springer (2023)
7. Ognyanova, K., Lazer, D., Robertson, R.E., Wilson, C.: Misinformation in action: Fake news exposure is linked to lower trust in media, higher trust in government when your side is in power. Harvard Kennedy School Misinformation Review (2020)
8. Pan, L., Wu, X., Lu, X., Luu, A.T., Wang, W.Y., Kan, M.Y., Nakov, P.: Fact-checking complex claims with program-guided reasoning. arXiv preprint arXiv:2305.12744 (2023)
9. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: EMNLP. pp. 2383–2392. Association for Computational Linguistics (2016)
10. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 3784–3803 (2021)
11. Thorne, J., Vlachos, A.: Evidence-based factual error correction. In: ACL-IJCNLP (Long Papers). pp. 3298–3309 (2021)
12. Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., Mittal, A.: The fact extraction and verification (fever) shared task. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). pp. 1–9 (2018)
13. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science **359**(6380), 1146–1151 (2018)
14. Wang, H., Shu, K.: Explainable claim verification via knowledge-grounded reasoning with large language models. arXiv preprint arXiv:2310.05253 (2023)
15. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

16. Xu, S., Pang, L., Shen, H., Cheng, X., Chua, T.S.: Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In: Proceedings of the ACM on Web Conference 2024. pp. 1362–1373 (2024)
17. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2369–2380 (2018)
18. Zhang, W., Deng, Y., Liu, B., Pan, S., Bing, L.: Sentiment analysis in the era of large language models: A reality check. In: Findings of the Association for Computational Linguistics: NAACL 2024. pp. 3881–3906 (2024)