PACAR: Automated Fact-Checking with Planning and Customized Action Reasoning using Large Language Models

Xiaoyan Zhao[1], **Lingzhi Wang**[1,*], **Zhanghao Wang**[3]
Hong Cheng[1], **Rui Zhang**[2], **Kam-Fai Wong**[1]
[1]The Chinese University of Hong Kong, Hong Kong SAR, China
[2]www.ruizhang.info, [3]Peking University, China
{xzhao, lzwang, hcheng, kfwong}@se.cuhk.edu.hk
1900011004@pku.edu.cn, rayteam@yeah.net

Abstract

In an era characterized by the rapid proliferation of information, the pervasive issues of misinformation and disinformation have significantly impacted numerous individuals. Consequently, the evaluation of information's truthfulness and accuracy has attracted substantial attention among researchers. In this work, we present a novel fact-checking framework called PACAR, fact-checking based on Planning And Customized Action Reasoning using LLMs. It comprises four modules: a claim decomposer with self-reflection, an LLM-centric planner module, an executor for carrying out planned actions, and a verifier module that assesses veracity and generates explanations based on the overall reasoning process. Unlike previous work that employs single-path decision-making and single-step veracity prediction, PACAR focuses on the use of LLMs in dynamic planning and execution of actions. Furthermore, in contrast to previous work that relied primarily on general reasoning, we introduce tailored actions such as numerical reasoning and entity disambiguation to effectively address potential challenges in fact-checking. Our PACAR framework, incorporating LLM-centric planning along with customized action reasoning, significantly outperforms baseline methods across three datasets from different domains and with varying complexity levels. Additional experiments, including multidimensional and sliced observations, demonstrate the effectiveness of PACAR and offer valuable insights for the advancement of automated fact-checking.

Keywords: Automated Fact-Checking, Large Language Model, Evidence Retrieval, Justification Generation

1. Introduction

The wide spread of misinformation has prompted a pressing need to develop automated fact-checking tools. Verifying the veracity of claims is an intricate task that requires a thorough understanding of both the claim itself and the accompanying evidence that either substantiates or contradicts it. Previous works (Rao and Daumé III, 2019; Majumder et al., 2021) have primarily focused on verifying atomic claims, which could not encompass the intricacies of real-world claims encountered in practical scenarios. More recent studies (Ousidhoum et al., 2022; Pan et al., 2023) have acknowledged the significance of addressing complex claims. Nevertheless, existing studies often rely on idealized "gold" evidence for predictions, which is unrealistic due to its limited availability in real-world scenarios. Moreover, they largely ignore how to effectively handle the integration of multiple sources of information and the intricate reasoning processes required for veracity prediction.

In this work, we propose a novel automated fact-checking framework called PACAR, comprising four key components: a claim decomposer with self-reflection to break down complex claims into sub-claims, a planner module that utilizes a customized toolset to manage actions at each reasoning step, an executor that executes the planned actions, and a verifier module that assesses the veracity of the original claim and generates explanations based on the overall reasoning process. All four modules in our framework are built upon large language models (LLMs) and operate in a zero-shot manner. LLMs are chosen as basis due to they are trained on vast amounts of data, making them a valuable knowledge source for veracity prediction. What's more, LLMs can comprehensively employ diverse data sources, facilitating the comprehension and comparison of facts across various subjects and domains.

While LLMs have shown remarkable instruction-following capabilities in various domains and applications (Qin et al., 2023), simply querying them with claims may not yield satisfactory performance and lacks explainability due to the black-box nature of prompting-based LLM utilization (Pan et al., 2023). PACAR incorporates multiple strategies, including self-reflection and global planner shown in Fig. 1, to effectively conquer the potential problems may arise when applying LLMs in fact-checking. Claim decomposer is adopted not only because complex claims often consist of multiple subclaims, but also because simplifying the

* Corresponding author.

LREC-COLING 2024, pages 12564–12573

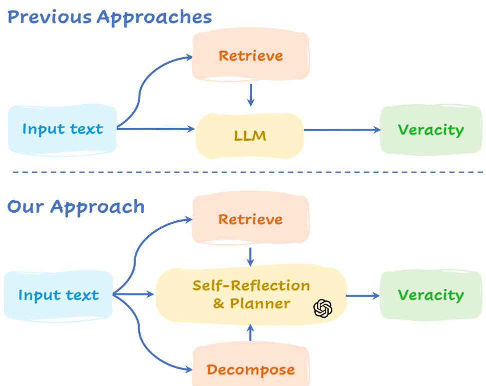

Figure 1: Approach Comparison.

query for LLMs holds promise as LLMs are already proven to be more adept at answering simple queries (Choudhary and Reddy, 2023); The self-reflection mechanism acts like a "semantic" gradient signal by asking LLMs to reevaluate the response according to the prior interaction history and improve the response with a concrete direction (Shinn et al., 2023); Our global planner for actions offers a more explicit reasoning process, synchronized with action results, in contrast to the implicit Chain-of-Thought prompt (Wei et al., 2022).

Besides, our PACAR surpasses the previous work by taking into account the characteristics of veracity reasoning. In contrast to previous approaches that rely on single-path decision-making (i.e., following a linear sequence of actions) (Pan et al., 2023) and single-step veracity prediction (i.e., making veracity judgments based on collected evidence in one step) (Chen et al., 2023) using LLMs, our PACAR framework leverages LLMs as a central component for planning actions and execution as planned in a dynamic manner, where actions can be conducted synthetically, and veracity prediction is based on the multi-step reasoning process. Furthermore, as opposed to previous methodologies that invariably pursued external retrieval, our planner-based retrieval stands out as more efficient since it engages time-consuming external retrieval only when necessary. Additionally, in contrast to previous work that exclusively relied on general reasoning, we introduce tailored actions such as numerical reasoning and entity disambiguation to effectively address the challenges that may arise in the context of fact-checking.

In addition, we conduct experiments on three datasets (i.e., SciFact (Wadden et al., 2020), FEVEROUS (Aly et al., 2021), HOVER (Jiang et al., 2020)), spanning diverse domains and claim complexities. The results show that our zero-shot framework outperforms ChatGPT, few-shot methods and conventional finetuning methods. Further experiments focusing on instances associated with numerical reasoning and entity disambiguation challenges reveal that our customized tool set plays a significant role in addressing the corresponding challenges which are common in veracity prediction. In brief, our main contributions are:

- We propose a novel automated fact-checking framework comprising four components, each designed to enhance the utility of LLMs or tailored to accommodate the specific characteristics of fact-checking tasks.
- We design a pioneering self-reflection module to proactively address potential error accumulation within the pipeline. Additionally, our customized agents are strategically crafted to adeptly improve the inference process, ensuring accurate reasoning from multiple evidence sources.
- Our proposed zero-shot framework outperforms all the baselines, spanning various categories, such as LLM-based, few-shot, and conventional fine-tuning methods. Further experiments involving multidimensional and sliced observations demonstrate the efficacy of PACAR.

2. Related work

2.1. Fact-checking

The landscape of automated fact-checking has witnessed significant advancements over the years. Previous models (Jiang et al., 2021; Liu et al., 2020) predominantly tackled claims verifiable via singular evidence (Jiang et al., 2020; Hanselowski et al., 2019). However, complex claims in the real world often necessitate multi-evidence reasoning. To bridge this, recent fact-checking models (Krishna et al., 2022; Barnabò et al., 2023) have incorporated retrieval techniques, enabling reasoning across diverse evidence. Notably, Chen et al. proposed an automated retrieval-based pipeline tailored for complex political claims. Pan et al. proposed a fact-checking system that decomposes the claim into a series of subtasks using program-guided reasoning and delegates each subtask to the corresponding handler sequentially. However, existing approaches (Soleimani et al., 2020; Nie et al., 2020; Chen et al., 2023; Pan et al., 2023) often serve as "black boxes" with limited explainability and the heavy interdependence of these components in the proposed unidirectional pipeline hinders their effectiveness when employed.

2.2. Explanation Generation

Explanation generation is important for persuasive automated fact-checking (Guo et al., 2022; Thakur et al., 2021; Shi et al., 2023). Numerous techniques have been proposed to address the limitations of solely providing a veracity label, aiming to enhance its effectiveness in explanation. Strategies range from utilizing attention metrics to emphasize evidence (Yang et al., 2019; Lu and Li, 2020), leveraging knowledge graphs for justification (Gad-Elrab et al., 2019; Ahmadi et al., 2019), and enriching context from sourced documents to aid task-specific response generation (Lewis et al., 2020; Borgeaud et al., 2022; Khattab et al., 2022; Peng et al., 2023). Unlike the previous works, our PACAR framework augments explainability, rectifies current pipeline shortfalls, and adapts to a broader spectrum of real-world situations.

3. Model

3.1. Problem Formulation

Our system's primary objective is to evaluate the veracity of a given claim C. This process potentially together with provided golden evidence, denoted as $E^{gold} = \{e_1^{gold}, e_2^{gold}, ..., e_{|E^{gold}|}^{gold}\}$), where $|E^{gold}|$ represents the total number of golden evidence pieces. The output is a label y indicating the claim's veracity as true or false. Additionally, we aim to provide an explanatory justification X supporting the predicted label. Without specifying, our veracity prediction and justification generation are not reliant on golden evidence.

3.2. Our Fact-Checking Framework

3.2.1. Claim Decomposor with Self-Reflection

Claims that accurately reflect real-world scenarios are often intricate, demanding a multitude of supporting evidence for predicting their veracity. Hence, given an input C, we propose to decompose it into various sub-claims, denoted as $\{c_1, c_2, ..., c_k\}$, where c_i is the i-th sub-claim. Each sub-claim c_i is a sub-claim in natural language that represents a specific aspect of the claim. Typically, such a decomposition process relies heavily on instructing LLMs with specific prompts. The decomposition process cannot guarantee that LLMs can consistently generate reasonable sub-claims.

To address the above issue, we propose a novel technique called *backward self-reflection*, aimed at enhancing the reliability of the decomposition process. We achieve this self-reflection by prompting LLMs to attempt to generate a claim C' that is semantically equivalent to C based on the decomposed sub-claims $c_1, c_2, ..., c_k$. If LLMs cannot generate such an equivalent claim, we then prompt them to generate new sub-claims after the above reflection. We summarize the entire forward decomposition and backward reflection process as:

$$C \leftrightarrow \{c_1, c_2, ..., c_k\} \tag{1}$$

where $\leftrightarrow$ indicates that the decomposition has been verified bidirectionally and k is the number of decomposed sub-claims.

3.2.2. Toolsets for Retrieval and Action

The toolset module offers two options: evidence retrieval and the LLM's reasoning capabilities. We utilize external retrieval as a supplementary method to obtain more comprehensive and accurate information. To ensure the overall efficiency of the verification process, our framework incorporates a retrieval planner that initiates external retrievals only when deemed necessary.

In contrast to previous work (Chen et al., 2023) that relies on black-box reasoning based on collected evidence, we first propose a set of tailored reasoning actions for fact-checking tasks and employ multi-step reasoning to do the fact-checking. Each agent specializes in addressing a specific challenge encountered during the fact-checking task. We summarize the challenge into three aspects: multi-hop reasoning in numerical, multi-hop reasoning in entity disambiguation, and multi-hop reasoning in other general scenarios. For this situation, we define the corresponding toolset in action, including *numerical reasoning* ($\mathcal{A}_{nr}$), *entity disambiguation* ($\mathcal{A}_{ed}$), and *general reasoning* ($\mathcal{A}_{gr}$). Considering the varying characteristics and requirements of different sub-claims and reasoning tasks, the toolsets for retrieval and action module can dynamically select suitable tools to support the reasoning process.

3.2.3. Planner and Executor

Retrieval Planner. After the claim decomposition, a list of sub-claims is generated. To optimize the overall claim assessment process and minimize reliance on external sources, we introduce a retrieval planner denoted as $\mathcal{R}$. The planner R is responsible for suggesting whether the LLMs can independently verify the decomposed claims. Concretely, we define $r = \mathcal{R}(c_i)$, where the variable r captures the result obtained from $\mathcal{R}$ after analyzing the sub-claim c_i. It's important to note that the return value r is strictly boolean, i.e., $r \in \{\text{Yes}, \text{No}\}$. This binary output signifies whether the model requires supplementary evidence for validation. The incorporation of such a retrieval planner helps streamline the claim assessment process by initiating external retrieval only when deemed necessary, ensuring efficiency in the overall verification procedure.

Evidence Executor. If the result obtained from the advisor retrieval is "Yes" then we conduct an evidence collection process, denoted as $\mathcal{S}$. We categorize the retrieval of evidence into two distinct settings: *Open-Book* and *Gold-Evidence*. The *Open-Book* setting implies that the system has the capability to actively access and reference external knowledge sources (e.g., Wikipedia) during its re-

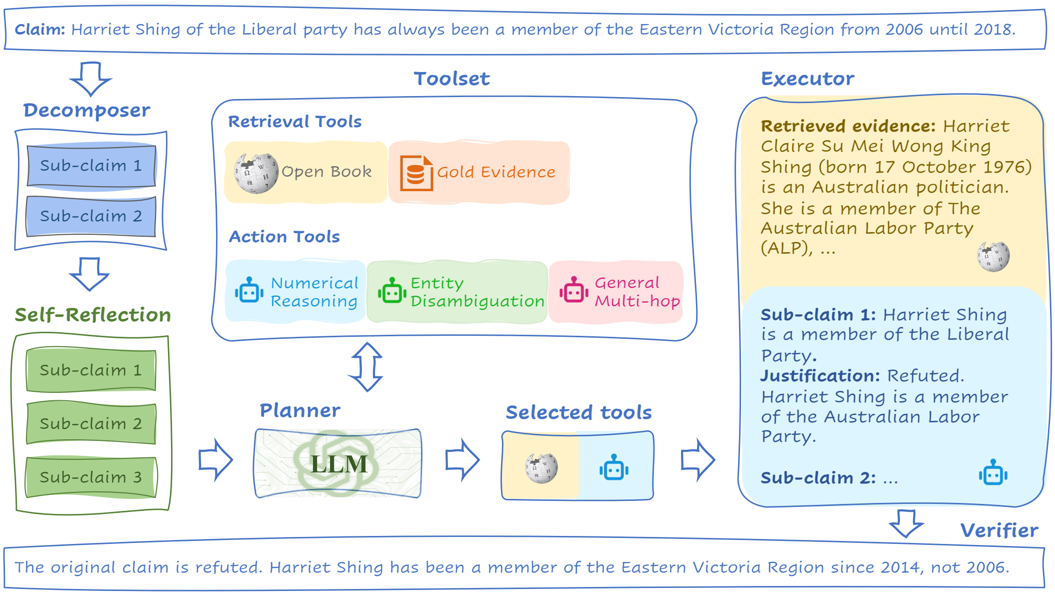

Figure 2: Our LLM-centric PACAR automated fact-checking framework with customized actions.

trieval process. In contrast, the *Gold-Evidence* setting can access gold evidence documents in the dataset, which ensure the high quality of the sources of evidence. We summarize the evidence collection as follows:

$$e_i \leftarrow \mathcal{S}(c_i) \tag{2}$$

where e_i represents the retrieved evidence for the i-th sub-claim. The retrieved evidence bank, denoted as $E = \{e_1, e_2, ..., e_k\}$, will be utilized in the subsequent veracity reasoning processes.

Action Planner. We design an action planner, denoted as $\mathcal{P}$, which is responsible for selecting an agent, denoted as a. Planner $\mathcal{P}$ chooses an agent from the set of available agents $\mathcal{A}_{nr}, \mathcal{A}_{ed}, \mathcal{A}_{gr}$ based on the challenging reasoning features of the content in the claim, formulated as:

$$a \leftarrow \mathcal{P}(\{\{c_1, e_1\}, \{c_2, e_2\}, ..., \{c_k, e_k\}\}) \tag{3}$$

The selected reasoning action a guides the decision-making process, based on the set of sub-claims $\{c_1, c_2, \ldots, c_k\}$ and their corresponding evidence E. The planner $\mathcal{P}$ plays a crucial role in planning this reasoning process, ensuring that the most appropriate action is chosen at each step to facilitate the veracity prediction.

Action Executor. To obtain the veracity analysis of each sub-claim, we employ the selected agent with an explicit role description to generate reasoning analysis among all the sub-claims. Specifically, we define the role of the selected agent by providing specific prompts. By leveraging LLMs' instruction-following capabilities, the agent generates reasoning analysis among all the sub-claims with their corresponding retrieved/generated evidence. This process ultimately yields justification j_i for fact-checking the sub-claim, formulated as:

$$\{j_1, \cdots, j_k\} \leftarrow a(C, \{(c_1, e_1), \cdots, (c_k, e_k)\}) \tag{4}$$

3.2.4. Verifier Module

To enhance the veracity assessment and prediction explainability of the whole fact-checking process, we employ a verifier module to generate reasoning analysis among all the sub-claims. Specifically, we define the role of verifier by providing specific prompts. By leveraging LLMs' instruction-following capabilities, the verifier generates reasoning analysis among all the sub-claims c_i with their corresponding justifications j_i. This process ultimately yields veracity label y and a comprehensive explanation exp for fact-checking the original claim, formulated as:

$$(y, exp) \leftarrow a(C, \{(c_1, j_1), \cdots, (c_k, j_k)\}) \tag{5}$$

The method of generating reasoning explanations through the guidance of specific agents coupled with the instruction-following capabilities represents a novel technique for complex claim fact-checking. By adopting this technique, we aim to improve the accuracy and interpretability of fact-checking results. The use of specialized agents allows us to address specific challenges inherent in the reasoning process among the sub-claims, thereby facilitating a more comprehensive evaluation of the veracity of claim. The resulting reasoning analysis contributes to a more robust and nuanced understanding of the fact-checking task.

Dataset	Domain	Claim Complexity	# of Eval
SciFact	Biomedical	Brief	300
FEVEROUS	Wikipedia	Brief, Complex	2,962
HOVER	Wikipedia	2-hop claims	1,126
		3-hop claims	1,835
		4-hop claims	1,039

Table 1: Statistics of Datasets.

4. Experimental Setup

4.1. Datasets

We evaluate our automated fact-checking model on three datasets, i.e., HOVER (Jiang et al., 2020), FEVEROUS (Aly et al., 2021), and SciFact (Wadden et al., 2020). These datasets span diverse domains and levels of complexity, which are widely adopted by researchers to benchmark the performance of automated fact-checking systems. And they cover broad topics (Wikipedia vs. biomedical), and different text types (news articles vs. research publications). Table 1 summarizes the datasets used in experiments. As we can see that, HOVER is divided into subsets based on the number of reasoning "hops" needed for claim verification. FEVEROUS, on the other hand, is designed for fact-checking over unstructured and structured data, annotating claims with evidence from sentences or cells from tables in Wikipedia. We use the same setup as the previous method (Pan et al., 2023), only selecting claims that require only sentence evidence. SciFact focuses on verifying scientific claimsby utilizing evidence extracted from abstracts of scientific papers in the research literature. These datasets provide a comprehensive platform to assess and improve the performance of fact-checking.

4.2. Baselines

To demonstrate the effectiveness of PACAR, we conducted comprehensive experiments comparing it against various baseline approaches categorized into three groups: *Fine-tuning*, *Few-shot* Prompting, and *Zero-shot* Prompting.

(1) The ***Fine-tuning*** methods aim to fine-tune pretrained language models specifically for performing fact-checking as a downstream task. The following baselines were employed: BERT-FC (Soleimani et al., 2020): This method involves fine-tuning the pretrained BERT language model using two loss functions, namely pointwise and pairwise. LisT5 (Jiang et al., 2021): It explored listwise evidence reasoning by utilizing the pretrained T5 language model for fact-checking. RoBERTa-NLI (Nie et al., 2020): This baseline involves fine-tuning the RoBERTa model using NLI datasets. MultiVerS (Wadden et al., 2022): This method predicts fact-checking labels and identifies explanations using a multi-task learning approach.

(2) The ***Few-shot*** prompting approaches leverage the powerful in-context learning capabilities of large language models. These approaches provide the model with a limited set of examples before prompting it with specific test cases. The following baselines were considered: CodeX (Chen et al., 2021): This approach first provides the CodeX model with 20 in-context examples and then prompts it with a template containing the test case. Flan-T5 (Chung et al., 2022): It prompts the Flan-T5 model for fact-checking by supplying 20 few-shot examples. ProgramFC (Pan et al., 2023): This baseline utilizes the LLMs (CodeX and Flan-T5) to generate reasoning programs that guide the verification process, assuming the availability of a few in-domain examples.

(3) The ***Zero-shot*** prompting methods involve feeding the language models with test cases without providing any examples. We employed the following baseline: ChatGPT: This method involves directly prompting the ChatGPT model to collect evidence first and then generate a judgment to verify the veracity of claims.

4.3. Implementation Details and Evaluation

We run all experiments using the gpt-3.5-turbo-0301 model. We leverage the Google service provided by Serper API as the retriever for our PACAR model. By incorporating this service, we are able to obtain a comprehensive collection of web page rankings, snippets, and other relevant metadata associated with a given query. For each sub-claim, we utilize the top paragraph (Recall@1) retrieved from the provided online website as supporting evidence. We adopt macro-F1 score to evaluate the fact-checking results by following (Pan et al., 2023; Feng et al., 2023).

5. Experimental Results

5.1. Main Comparison Results

Table 2 presents a comprehensive comparison between our proposed PACAR model and state-of-the-art models across all settings. We have the following observations based on Table 2.

The Effectiveness of PACAR in General Domains. The HOVER and FEVEROUS datasets are challenging as they contain lengthy and intricate claims that often necessitate the integration of

Models	OPEN-BOOK					GOLD-EVIDENCE				
	HOVER			FEVEROUS	SciFact	HOVER			FEVEROUS	SciFact
	2-hop	3-hop	4-hop			2-hop	3-hop	4-hop		
Fine-tuning										
BERT-FC	50.68	49.86	48.57	51.67	-	53.40	50.90	50.86	74.71	-
LisT5	52.56	51.89	50.46	54.15	-	56.15	53.76	51.67	77.88	-
RoBERTa-NLI	63.62	53.99	52.40	57.80	-	74.62	62.23	57.98	88.28	-
MULTIVERS	60.17	52.55	51.86	56.61	44.90	68.86	59.87	55.67	86.03	<u>72.54</u>
Few-shot										
CodeX	65.07	56.63	57.27	62.58	-	70.63	<u>66.46</u>	63.49	89.77	-
FLAN-T5	69.02	60.23	55.42	63.73	-	73.69	65.66	58.08	90.81	-
ProgramFC	<u>69.36</u>	<u>60.63</u>	<u>59.16</u>	<u>67.80</u>	<u>56.34</u>	<u>74.10</u>	66.13	<u>65.69</u>	<u>91.77</u>	71.82
Zero-shot										
ChatGPT	66.94	60.56	58.73	55.72	45.32	71.42	64.87	63.65	83.49	65.60
PACAR (Ours)	**73.13**	**64.07**	**63.82**	**72.61**	**61.24**	**76.86**	**70.10**	**69.95**	**94.43**	**75.06**

Table 2: Main results (macro-F1 in %) on of HOVER, FEVEROUS, and SciFact datasets. The best and second-best results in each column are in **bold** and <u>underlined</u> respectively.

multiple pieces of evidence. As shown in Table 2, our PACAR model exhibits superior performance compared to the baselines, with improvements of 4.66% and 4.81% in the open-book settings in HOVER's 4-hop claims and FEVEROUS dataset, respectively. These results highlight the model's exceptional analytical and reasoning capabilities when dealing with complex claims. Moreover, the strong baseline ProgramFC operates in a few-shot setting which requires 20 in-domain examples, imposing a significant burden on the LLM. In the zero-shot setting, the baseline ChatGPT demonstrates its impressive fact-checking abilities while its performance is suboptimal. However, our model is both in zero-shot learning and further improves the performance by utilizing claim decomposition with self-reflection, allowing for dynamic evidence collection.

The Effectiveness of PACAR in Professional Domains. In the SciFact dataset, claims are expert-written sentences from scientific literature, requiring fact-checking models to gather external evidence for verification. In Table 2, there are 4.9%, and 3.24% improvements on the SciFact dataset in open-book setting and gold-evidence setting, respectively. The results surpass the performance of strong baselines such as ProgramFC and ChatGPT. It demonstrates the effectiveness of our retrieval planner and evidence executor strategies in addressing the need to retrieve pertinent evidence for fact-checking purposes. Furthermore, the diverse experimental datasets encompass real-world claim scenarios, spanning general and specialized domains with claims of varying lengths and complexities, facilitating a comprehensive evaluation of PACAR's effectiveness.

	2-hop	3-hop	4-hop	FEVEROUS	SciFact
PACAR	**76.86**	**70.10**	**69.95**	**94.43**	**75.06**
-w/o SR&Agents	75.51	68.39	67.82	92.78	74.02
-w/o SR	76.25	69.03	69.36	93.65	74.54
-w/o Agents	75.83	68.95	68.57	93.24	74.33

Table 3: Ablation results of PACAR.

5.2. Ablation Study

We conduct an ablation study to further assess the effectiveness of the proposed mechanisms.

5.2.1. The Effectiveness of Self-Reflection

Through our experimental analysis, the backward self-reflection module serves two purposes: correction and refinement of sub-claims. The backward self-reflection module appropriately adjusts the sub-claims and can occasionally modify the sentence structure of correct sub-claims to make it more sound. The proper decomposition of sub-claims is facilitated by forward claim decomposition with a backward self-reflection module affects the model's retrieval process and contributes to performance gains, as shown in Table 3. We evaluate two ablation scenarios: PACAR without the self-reflection module (marked as w/o SR) and PACAR excluding both self-reflection and agents (marked as w/o SR&Agents). The experimental results at different hop levels, namely 2-hop, 3-hop, and 4-hop, demonstrate the increasing prominence of the benefits brought about by the self-reflection module. Specifically, we observed improvements of 1.35%, 1.71%, and 2.13% for per hop level, respectively. We also observed significant performance gain in complex claims such as HOVER, and FEVEROUS, while the improvements in SciFact are less pronounced.

5.2.2. The Effectiveness of Specific Agent

In Table 3, we evaluate the removing numerical reasoning (nr) and entity disambiguation (ed) agents (marked as w/o Agents). To thoroughly investigate the effectiveness of these two agents, we further analyze the distribution of claims by different agents in the dataset and the resulting improvements. Figure 3 (a) displays the proportions of numerical reasoning, entity disambiguation, and general reasoning claims based on the original annotations in the FEVEROUS dataset. Figure 3 (b) analyzes the proportions of operations performed by the numerical reasoning agent, entity disambiguation agent, and general reasoning agent in our PACAR model. We observed that the category distribution of numerical reasoning, entity disambiguation, and other multi-hop reasoning obtained by the PACAR model is inconsistent with the category distribution in the FEVEROUS dataset. This discrepancy arises because the FEVEROUS dataset primarily categorizes claims based on the claims themselves, while our model simultaneously analyzes both the claims and evidence, resulting in a representation that better aligns with real-world scenarios.

Additionally, Figure 4 illustrates the improvement achieved by the numerical reasoning agent and the entity disambiguation agent on their respective claims. Specifically, we present the results of PACAR w/o nr agent and w/o ed agent on the numerical reasoning data and the entity disambiguation data, respectively, as shown in Figure 3 (b). The results demonstrate the significant impact of the agent modules, particularly when they provide explicit and useful clues for reasoning, leading to better explanations. This highlights the importance of tailored reasoning actions performed by specific agents. We observed that claims involving changes in numerical values or years are often assigned to the numerical reasoning agent by the coordination mechanism. The claims with different nouns tend to be arranged to the entity disambiguation reasoning agent. Through the involvement of the planner mechanism, our selected agents gain a clearer understanding of the types of claims, which provides useful explanations and more detailed insights behind the predictions. These findings emphasize the importance of the agent modules in our framework, as they enable customized reasoning operations based on the characteristics of the claims. This customization plays a crucial role in enhancing the performance of the veracity prediction.

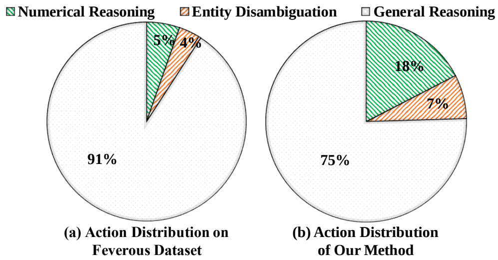

Figure 3: The proportions of numerical reasoning, entity disambiguation, and general reasoning categories in the original annotations of FEVEROUS dataset and the actions performed by PACAR.

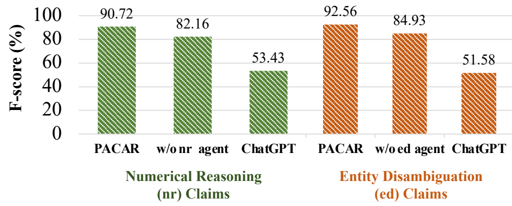

Figure 4: The ablation results of numerical reasoning agent and entity disambiguation agent.

5.3. Quantitative Analysis

5.3.1. The Comparison with ChatGPT

In our section, we compare our proposed model, PACAR, with the base model ChatGPT in various settings to further evaluate their performance. We considered four different models for comparison: (i) Prompt-only ChatGPT (Prompt): ChatGPT only takes the prompt and claim as input, without any additional evidence. (ii) ChatGPT with gold evidence (ChatGPT-G): ChatGPT is provided with the prompt containing the claim along with the corresponding gold evidence. (iii) Chain of Thought with gold evidence (CoT-G): ChatGPT is given gold evidence and a specific prompt “Let’s think step by step” to guide its reasoning process. (iv) PACAR with gold evidence (PACAR-G): Our proposed PACAR in the gold-evidence setting.

The results shown in Table 4, demonstrate that ChatGPT exhibits sub-optimal performance in fact-checking tasks, both in leveraging its notable inference ability and when provided with gold evidence. These findings highlight the inherent limitations of LLMs in effectively addressing fact-checking tasks, such as the problem of hallucination and the limited reasoning ability. In contrast, our proposed PACAR model addresses the shortcomings of LLMs by incorporating forward claim decomposition with backward self-reflection, and customized reasoning actions performed by specific agents. By leveraging these strategies, PACAR can incorporate diverse sources of evidence and effectively integrate them, leading to more reliable and explainable fact-checking performance.

	2-hop	3-hop	4-hop	FEVEROUS	SciFact
Prompt	58.73	52.65	49.39	52.56	36.71
ChatGPT-G	71.42	64.87	63.65	83.49	65.60
CoT-G	<u>72.85</u>	<u>65.61</u>	<u>64.08</u>	<u>84.22</u>	<u>67.85</u>
PACAR-G	**76.86**	**70.10**	**69.95**	**94.43**	**75.06**

Table 4: Comparison results of our proposed PACAR and variants based on ChatGPT.

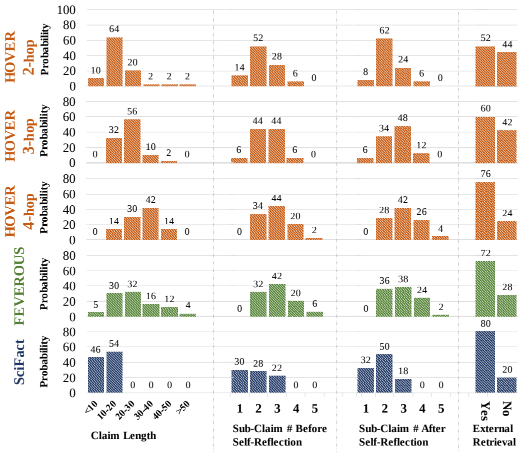

Figure 5: Distribution of the length of claims, the number of sub-claims, and whether need to retrieve evidence, respectively.

5.3.2. Distribution

The claim decomposition with self-reflection strategy leverages LLMs to divide claims into sub-claims and reflect the correctness of the decomposition. To evaluate the strategy, we manually analyzed 50 claims and performed statistical analysis on the claim decomposition results. Figure 5 shows the distribution of the length of claims, the number of sub-claims at forward claim decomposition and backward self-reflection, and whether need to retrieve evidence. Our analysis identified three linguistic cues considered during the decomposition process. First, keywords, i.e., words or phrases, indicate important concepts related to a claim. Second, logical connections reveal the interdependency and structure within a claim. Finally, semantic relationships involve analyzing the underlying meaning and connections between words and phrases. The LLM employs these linguistic cues to identify potential sub-claims that contribute to the overall claim. By dividing claims into sub-claims during the decomposition and self-reflection stages, our framework enhances the decomposition process.

5.4. Qualitative Analysis

We conducted a comprehensive analysis of the interpretability of our proposed model, PACAR. To evaluate its interpretability, we selected a sample of 30 claims from the FEVEROUS datasets. We observe that PACAR effectively enhances the interpretability of fact-checking compared to previous models. This improvement is attributed to the explicit claim decomposition, dynamic action planner, and executor, which aid in human understanding of the fact-checking process. Specifically, Table **??** presents an illustrative analysis example where the PACAR model successfully identifies both supporting and refuting evidence. Through the generation of informative and contextually relevant explanations for the predicted veracity labels, the PACAR model significantly improves the transparency and interpretability of the fact-checking process. Moreover, the model exhibits a remarkable capability to incorporate diverse evidence sources and seamlessly integrate them, resulting in more robust and reliable fact-checking outcomes. These findings underscore the efficacy and potential of the PACAR model in advancing the field of automated fact-checking.

Moreover, we also conducted an error analysis during this manual checking process to examine the error types encountered in our PACAR model. We manually classify the errors into three categories: (i) Syntax errors, which pertain to issues with the grammatical structure or composition of the subclaims, (ii) Semantic errors, which involve inaccuracies or inconsistencies in the meaning or interpretation of the subclaims, and (iii) Reasoning errors, which encompass flaws in the logical or rational connection between the subclaims and the overall claim. We think these are the persisting challenges encountered by fact-checking models. Hope to provide valuable insights for future improvements and advancements in the field.

6. Conclusion

In summary, our work introduces PACAR, an innovative fact-checking framework featuring four distinct modules. These modules are each designed to inspire the utility of LLMs to tackle the specific complex characteristics of fact-checking. Unlike previous approaches, PACAR optimally leverages LLMs, adopting dynamic planning and tailored actions to tackle the challenges in fact-checking. Extensive experiments show the effectiveness of PACAR.

Acknowledgments

This research was supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commissions' InnoHK program and by grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14217622).

7. Bibliographical References

Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable fact checking with probabilistic answer set programming. *arXiv preprint arXiv:1906.09198*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Giorgio Barnabò, Federico Siciliano, Carlos Castillo, Stefano Leonardi, Preslav Nakov, Giovanni Da San Martino, and Fabrizio Silvestri. 2023. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media*, 33:100244.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Nurendra Choudhary and Chandan K Reddy. 2023. Complex logical reasoning over knowledge graphs using large language models. *arXiv preprint arXiv:2305.01157*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. 2023. Knowledge refinement via interaction between search engines and large language models. *arXiv preprint arXiv:2305.07402*.

Mohamed H Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019. Exfakt: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 87–95.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. Proofver: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

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

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of* *the Association for Computational Linguistics*, pages 7342–7351.

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312.

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.

Nedjma Ousidhoum, Zhangdie Yuan, and Andreas Vlachos. 2022. Varifocal question generation for fact-checking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2532–2544.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. *arXiv preprint arXiv:2305.12744*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 143–155.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 359–366. Springer.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.

David Wadden, Kyle Lo, Lucy Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. Multivers: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76.

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.

Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Hu. 2019. Xfake: Explainable fake news detector with visualizations. In *The world wide web conference*, pages 3600–3604.