Argument-structured Justification Generation for Explainable Fact-checking

Xiaoou Wang, Elena Cabrio, Serena Villata

Argument-structured Justification Generation for Explainable Fact-checking

Xiaoou Wang
Université Côte d'Azur
CNRS, Inria, I3S
Sophia Antipolis, France
xiaoou.wang@univ-cotedazur.fr

Elena Cabrio
Université Côte d'Azur
CNRS, Inria, I3S
Sophia Antipolis, France
elena.cabrio@univ-cotedazur.fr

Serena Villata
Université Côte d'Azur
CNRS, Inria, I3S
Sophia Antipolis, France
serena.villata@cnrs.fr

***Abstract*—Justification production is a central task in automated fact-checking, and most studies cast this task as summarization. However, the majority of previous studies presume the availability of human-written fact-checking articles, which is unrealistic in practice. In this work, we address this issue by proposing a novel approach to generate argument-based justifications to improve fact-checking. Our contribution is threefold. First, our extensive experimental setting shows that, despite lower ROUGE scores, our argument-structured summarizer produces summaries leading to better claim verification performance than the state-of-the-art summarizer in fact-checking on three different benchmarks for this task. Second, our jointly-trained summarization and evidence retrieval system outperforms the state-of-the-art method on ExClaim, the only dataset where no human-written fact-checking articles are provided during verification of news claims. Third, we show that integrating attackability evaluation into the training process of the summarizer significantly reduces hallucinated argument relations, leading to more reliable and trustworthy justification generation.**

I. Introduction

Justification production is an important task in journalistic and automated fact-checking [1] for multiple reasons: readers need to be convinced on the interpretation of the evidence [2], justification allows a feedback loop which corrects judgment errors [3] and finally, using black-box models without explanations can induce a "backfire effect" which leads to an increased conviction in the incorrect claim [4]. Providing justifications for fact-checking verdicts helps increase the credibility of the process [5] and well-crafted justifications can educate readers about how to critically evaluate claims and identify misinformation themselves [6]. However, manually crafting justifications is a time-intensive process and even professional fact-checkers need to spend several hours or even days to verify the accuracy of a claim [7]. While there is an increasing focus on justification production [6], most of these approaches presume the availability of a pre-existing human-written fact-checking article as the basis for justification generation, thereby overlooking the critical step of evidence retrieval. This is unrealistic in practice, as fact-checking articles are rarely available for all new claims. Besides, the generated justifications are mostly evaluated using overlap-based metrics such as ROUGE scores [8] without considering the essential task of automated fact checking, i.e., claim verification.

In this work, we introduce a novel argument-structured justification generation method based on a novel dataset that we built from LIAR-PLUS [9], named LIARArg. Our main findings are as follows:

- Trained with the newly created dataset LIARArg, our argument-based summarizer produces summaries better suited for claim verification, leading to significant improvements in F1 scores across three standard benchmarks compared to the state-of-the-art summarizer in fact-checking [10] as well as human-written summaries. Besides, further analysis shows that ROUGE scores are not correlated with F1 scores, highlighting the fact that the quality of the generated summaries in the context of automated fact-checking is not necessarily reflected by ROUGE scores.
- Our jointly-trained summarization and evidence retrieval system, **ArgLM**, outperforms the state-of-the-art method **JustiLM** on ExClaim [11], the only dataset for this task, where no human-written fact-checking articles are provided during verification of news claims.
- Integrating attackability evaluation into the summarizer's training process significantly reduces hallucinated argument relations, leading to more reliable justification generation.

II. Related work

Automated fact-checking involves four stages [6]: *i)* claim detection, to identify or rank the claims to verify; *ii)* evidence retrieval, to find sources supporting or refuting a claim; *iii)* claim verification, to assign veracity labels to claims, and *iv)* justification production, which explains the verdict. This section discusses works related to the justification phase, which is the focus of this paper.

Concerning justification production, one line of research utilizes attention weights to highlight key parts of the retrieved evidence as explanations [12], [13]. Another approach adopts logic-based rules, such as knowledge graphs [14], [15], where explanations are derived by tracing the rule paths of algorithms like decision trees. Both approaches have some drawbacks:

This work has been partially supported by the ANR project ATTENTION (ANR21-CE23-0037) and the French government through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

Claim (input)	John McCain: Iran "might not be a superpower, but the threat the government of Iran poses is anything but 'tiny,'" as Obama says.
Evidence retrieval module	Doc1: But Obama never said the threat from Iran was "tiny" or "insignificant"... Doc2: In fact, Obama has repeatedly called Iran a grave threat. Doc3: This isn't the first time Obama has talked about the grave threat posed by Iran...
Summarization module	John McCain said that... "But Obama never said..." **attacks** this claim, "In fact, Obama has repeatedly called Iran a grave..." **attacks** this claim ...
Claim verification module	False

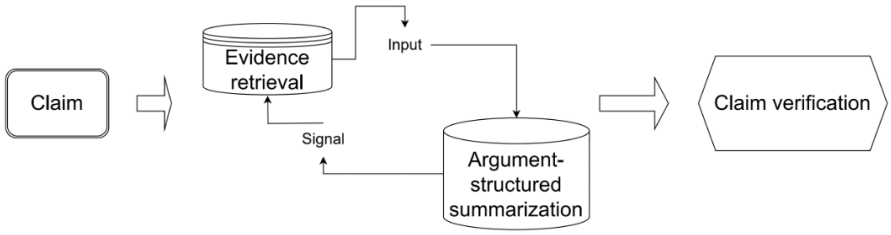

Fig. 1. The fact-checking architecture we propose, where evidence retrieval and summarization are trained jointly.

rule-based explanations are not readily accessible to general users, and the salience-based approach lacks structured information and does not reveal the underlying reasoning process.

Recent studies cast justification production as summarization. Kotonya and Toni [16] propose two steps: SBERT to extract sentences from fact-checking articles, and then BERTSUM model [17] to generate abstractive justifications based on the extracted sentences. Russo et al. [10] examine various existing methods for extractive [18] and abstractive [19] summarization of fact-checking articles. The authors conclude that combining abstractive summarization with a claim-driven extractive step using SBERT yields the best results.

The current summarization methods have several limitations: *i)* they are dependent on human-written fact-checking articles as input, rarely available at deployment; *ii)* the evidence search process is omitted; and, *iii)* generated justifications are mostly evaluated using overlap-based metrics as ROUGE, without considering the essential task of claim verification in automated fact checking. However, summaries for fact-checking should be similar to the input fact-checking article but also relevant for claim verification. We address this twofold issue by proposing a novel argument-based summarization pipeline for fact-checking. Our pipeline only requires human-written fact-checking articles during the training process whilst it retrieves evidence within a large corpus during inference, which is a more realistic scenario. Additionally, we evaluate the generated justifications using both ROUGE scores and the F1 scores of a claim verification system that takes the generated explanations as input. This dual evaluation redefines the generation task from a simple summarization task to fact-checking-oriented justification production. On this line, Khan et al. [20] propose a two-step pipeline to first retrieve evidence using adapted dense passage retrieval [21], and then a RoBERTa-based module to verify the claims based on the concatenated retrieved evidence, thus evaluating the usefulness of retrieved evidence (without summarization). Yao et al. [22] propose a three-step setup on multimedia data with evidence retrieval, claim verification and explanation generation independently trained. Here, we propose to jointly train evidence retrieval and justification production so that both tasks can benefit signals from each other. Similarly, JustiLM [11] uses a perplexity distillation loss [23] to leverage signals from the generation process so that documents contributing to lower-perplexity outputs are ranked higher.

III. Our fact-checking pipeline

The main objective of our work is to produce summaries tailored for automated fact-checking. To this end, we propose an argument-structured summarizer trained using human-written summaries annotated with argument components and relations. Subsequently, we aim to jointly train the evidence retrieval and summarization modules so that human-written fact-checking summary articles are no longer required. Our full pipeline is visualized in Fig. 1. A news claim is provided as input to the evidence retrieval module which identifies relevant articles in a large corpus. These articles are then fed into the summarizer module which generates an argument-structured summary with identified attack and support relations. The summarizer provides in turn supervisory signals to improve the evidence retrieval module. Finally, the summary is fed into the claim verification module to produce a veracity label.

IV. Datasets

The central module of our pipeline is the argument-based summarizer, which requires a fact-checking dataset annotated with argument components and relations. Since such resource does not exist, we built the LIARArg dataset by manually annotating the LIAR-PLUS dataset [9] with argumentative labels (i.e., components and relations). LIAR-PLUS [9] is an extension of the LIAR dataset [24] consisting of 12,836 news claims taken from POLITIFACT[1] and labeled with

[1]https://www.politifact.com/

veracity. This dataset is annotated with six veracity labels: Pants-On-Fire, False, Mostly-False, Half-true, Mostly-True and True. Each claim is accompanied by an automatically scraped summary from the "our ruling" or "summing up" section of a human-written fact-checking article. Fig. 2 shows an instance of the LIAR-PLUS dataset. Note that verdict phrases, such as "it is true" or "this is misleading", have been filtered to minimize label leakage. Originally, LIAR-PLUS doesn't contain the fact-checking article, we scraped all the corresponding articles to make claim-article-summary triples.

Statement:"Says Rick Scott cut education to pay for even more tax breaks for big, powerful, well-connected corporations."
Speaker: Florida Democratic Party
Context: TV Ad
Label: half-true
Extracted Justification: A TV ad by the Florida Democratic Party says Scott "cut education to pay for even more tax breaks for big, powerful, well-connected corporations." However, the ad exaggerates when it focuses attention on tax breaks for "big, powerful, well-connected corporations." Some such companies benefited, but so did many other types of businesses. And the question of whether the tax cuts and the education cuts had any causal relationship is murkier than the ad lets on.

Fig. 2. Excerpt from the LIAR-PLUS dataset.

Two kinds of argument components, claim and premise, are annotated on the automatically scraped summaries. Claims are defined as statements to denote standpoints, and premises as statements that can be verified to some extent, including typically some quotes from original documents or concrete statistics. We annotated fine-grained argument relations, i.e., support, attack, partial attack and partial support. An argument component supports another one when it validates this component, and it attacks another one when it contradicts the proposition of the target component. Partial support is used when an argument component validates certain aspects of another component, but diverges in some other aspects. Partial attack is used when the source argument component is not in full contradiction, but it weakens the target component. Example (1) shows an instance of attack and partial support where $Premise_1$ (in braces) attacks the **Claim** (in brackets), and $Premise_2$ partially supports the **Claim**.

(1) **[Iran might not be a superpower, but the threat the government of Iran poses is anything but "tiny" as Obama says]**.
{*But Obama never said the threat from Iran was "tiny" or "insignificant,"*}$_1$, {*only that the threat was tiny in comparison to the threat once posed by the Soviet Union*}$_2$.

The annotation guidelines have been defined by three experts in fact-checking and argumentation[2]. Two annotators with a background in computational linguistics, with a focus on argument mining, carried out the first annotation phase. Three rounds of annotation have been performed to ensure the robustness of the Inter-Annotator Agreement (IAA). After the first annotation round, the annotators identified and discussed disagreements to reconcile them. After this reconciliation phase, a second annotation round has been addressed followed by a second reconciliation phase. Finally, a third round was performed to confirm that the reconciliation produced consistent results. Table I shows that this annotation process led to stable improvements in IAA scores. It is worth noting that argument annotation is a notoriously challenging task. A score of 0.73 for claim and premise annotation is generally considered very high, while a score of 0.61 is also substantial in argumentation [26], [27].

TABLE I
IAA SCORES IN FLEISS' KAPPA [25] FOR THREE ROUNDS OF ARGUMENT COMPONENT AND RELATION ANNOTATION.

Sample	Arg. comp.	Arg. rel.
150 texts	0.72	0.48
150 texts	0.71	0.59
90 texts	0.73 (substantial)	0.61 (moderate)

LIARArg enables the automatic construction of what we call argument-structured summaries from the original summaries[3]. The process is as follows: each summary begins with "X said..." followed by the concatenation of relations with the claim, formatted as "A attacks this claim," "B supports this claim,", following the appearance order: attack, support, partial attack, and partial support. This form of summary contains the essential information relevant for fact checking, and our hypothesis is that this reformulation should enhance the claim verification module. The final LIARArg dataset comprises 2,832 automatically scraped summaries, the converted argument-structured summaries, their corresponding full-length fact-checking articles (40 sentences on average), and their veracity labels over the 6 categories of LIAR-PLUS.

To assess the effectiveness of our summarizer on out-of-domain data, we evaluate it on FNC-1 [28] and Check-COVID [29] datasets. FNC-1 is a well-known benchmark for fake news classification derived from the Emergent Dataset [30], containing 75385 labeled headline and article pairs across more than 20 topics. Check-COVID is a benchmark of 1504 claims about COVID-19 where each news claim is paired with evidence (full length scientific journal articles). FNC-1 is framed as a stance detection task with 4 labels: agree, disagree, discuss and unrelated, while Check-Covid is framed as a binary task with Refute and Support labels.

To assess our complete pipeline where no human-written fact-checking articles are available, the only corpus available to this date is ExClaim [11], which contains 6,951 real-world claims and their corresponding labels and human-written summaries, together with a large searchable corpus of 957,949

[2]https://anr-attention.github.io/gd.pdf

[3]The LIARArg dataset will be made fully available upon paper acceptance.

TABLE II
SUMMARY OF THE DATASETS USED IN THE PRESENT WORK.

Datasets	Format	Domain	Objective
LIARArg	2832 claims full-length human-written articles automatically scraped human-written summaries converted argument-structured summaries 6 labels	Political News	Train and evaluate the summarizer
LIAR-PLUS	12836 claims completed with full-length human-written articles automatically scraped human-written summaries converted argument-structured summaries 6 labels	Political News	Evaluate the summarizer
FNC-1	75385 claims and article pairs 4 labels	General News	Evaluate the summarizer on out-of-domain data
Check-Covid	1504 claims and article pairs 2 labels	Covid	Evaluate the summarizer on out-of-domain data
ExClaim	6951 claims human-written summaries 957,949 chunk-level documents containing evidence 3 labels	General News	Evaluate the full pipeline illustrated in Fig. 1 where no human-written fact-checking articles are provided

chunk-level documents for fine-grained evidence retrieval. Labels are false, mixture and true. Table II summarizes the main features of the employed datasets.

V. EXPERIMENTAL SETTING

Our experiments are divided into two parts:

1) We train the summarizer alone using LIARArg and compare our summarizer with state-of-the-art summarizers in fact-checking on LIAR-PLUS, FNC-1 and Check-Covid, illustrating that argument-structured summaries improve claim verification. During this process, only 1 human-written fact-checking article is given as input.
2) We train the summarizer and the evidence retrieval module together on ExClaim to evaluate our approach in a real world setting. It is important to note that although human-written summaries are used during the training phase (as detailed in Section V-C), no pre-written fact-checking article is needed during the inference phase and several pieces of evidence are fed as input.

This setup enables us to evaluate our summarization approach independently of any artifacts from the evidence retrieval process before assessing the feasibility of a fully automated pipeline.

A. The argument-structured summarizer

We fine-tune Mixtral-8x22B [31] using QLoRA [32]. Mixtral-8x22B is the updated version of Mistral 7B leveraging grouped-query attention [33] for faster inference and sliding window attention [34] to handle longer sequences more efficiently. We choose this model because it is open source and it produces comparable results with GPT and other open sources LLMs [31]. The official checkpoint on HuggingFace, Mixtral-8x22B-Instruct-v0.1[4], is used with all the default parameters unchanged. For QLoRA parameters, the learning rate is set to 2e-4 with a dropout of 0.1, rank 64, Alpha 16, over 4 epochs. The loss function is Cross-Entropy ($Loss_{ce}$). We denote this summarizer as **SumArg**.

[4]https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1

B. Improving the argument-structured summarizer

Although the standard approach is to fine-tune summarization with Cross-Entropy loss alone, we argue that this process, in the context of argument-structured generation, can be improved by drawing insights from the counter-argument generation literature. More precisely, Jo et al. [35] detect attackable sentences in a text since attacking weak premises is a common counter strategy. We adjust this task to compute not only attackability but also supportability of a claim-premise pair. Based on the results of [35] and [36], we learn to gain an attackability score (-1 for attackable, 0 for neutral, and 1 for supportable). In particular, given a set of sentences and a claim, we first represent each sentence by concatenating its tokens with the claim's tokens, separated by the two special tokens *[cls]* and *[sep]*: *[cls] claim_tokens [sep] premise_tokens [sep]*. Next, the resulting sequences are passed through a BERT model to obtain a vector representation for each claim-sentence pair. Each vector is then projected through a dense layer to get a score $\hat{y}$ that reflects the attackability score of a sentence pair. Finally, a list-wise objective function (using a Softmax loss) is optimized jointly on all sentence pairs: $l(y, \hat{y}) = -\sum_{i=1}^{n} y_i \cdot \log\left(\frac{\exp(\hat{y}_i)}{\sum_{j=1}^{n}\exp(\hat{y}_j)}\right)$ where y is the ground-truth score (1, -1 or 0). For the training corpus, we use the ChangeMyView (CMV) dataset [35] which contains 199,711 claim-sentence pairs labeled as attacked (-1) or not attacked (0), as well as LIARArg where partial attacks and partial supports are generalized to attacks and supports, leading to 18496 sentence pairs labeled as attacked (-1), supported (1), and neutral (0).

Since our argument-structured summary always starts with the claim, this attackability scoring method allows us to compute attackability scores of the first sentence versus the rest, obtaining a global score for the whole summary by summing all the individual scores. We call this module **AttScorer**.

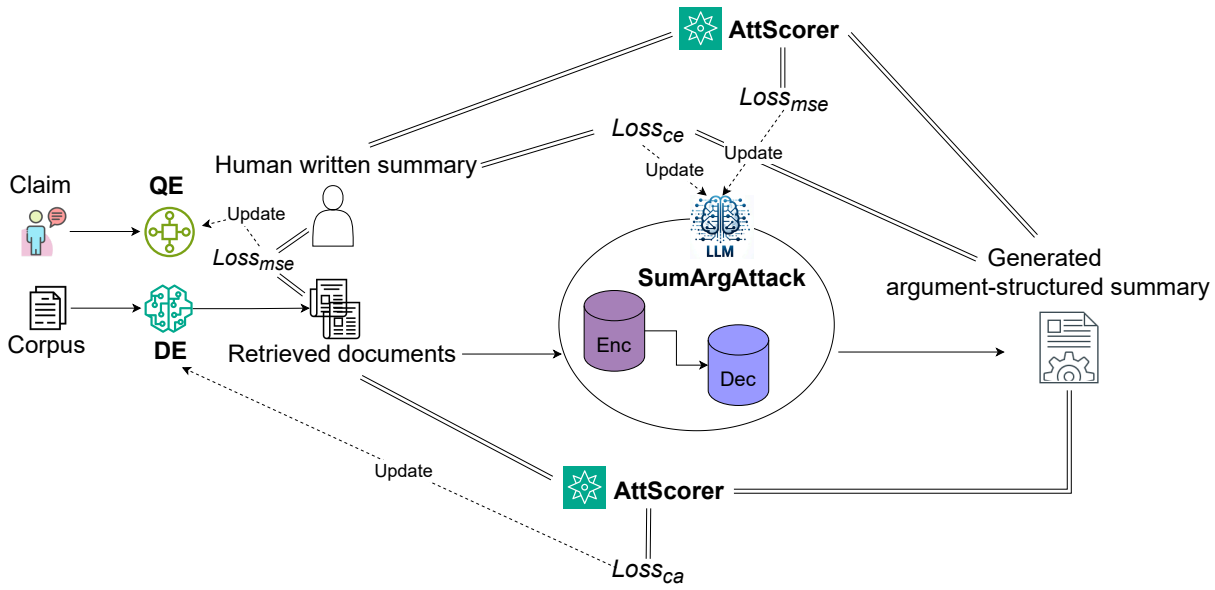

Fig. 3. Our evidence retrieval system jointly trained with justification generation. $Loss_{mse}$ compares the retrieved documents and human-written summary to update the query encoder. $Loss_{ca}$, based on the attackability scores of the generated summaries and retrieved documents, updates the document encoder and the language model. $Loss_{ce}$, based on the lexical distribution of the generated and human-written summaries, updates the language model. Additionally, $Loss_{mse}$, comparing the attackability scores of groundtruth summary and generated summary, updates the language model to achieve similar attackability scores. QE: Query Encoder; DE: Document Encoder.

Based on the hypothesis that the attackability scores of the generated summary and the ground truth should be similar, we add the Mean Squared Error (MSE) loss function in the fine-tuning process (Section V-A) to minimize the attackability score difference: $L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. We denote the loss $Loss_{mse}$ and the attackability-enhanced summarizer **SumArgAttack**.

C. Jointly training the summarizer and evidence retrieval modules

The whole architecture is illustrated in Fig. 3. To jointly train the summarizer and evidence retrieval module, we add two additional losses to the training process. We base our approach on the retrieval-augmented generation (RAG) framework [37], which includes a retriever for detailed evidence retrieval and an LLM for generating textual justifications. The retriever uses the claim text as input to retrieve the top-N chunk-level documents from a textual knowledge corpus. The LLM then uses these documents along with the claim to generate justifications. The retriever and LLM can be jointly trained within a single RAG framework, allowing us to use fact-checking articles as auxiliary resources to provide supervisory signals during training, thus enhancing the quality of generated justifications. As in [11], we use Atlas [23] as our backbone model because of its strong few-shot learning capability and its flexibility in incorporating fact-checking articles into the training process.

Given a claim $\mathbf{x}$, the retriever should return the documents that help the LM to generate better justifications. To enable the training of the retriever, Atlas utilizes a dense retriever named Contriever [38], which is pre-trained using the MoCo contrastive loss [39]. Contriever is a dual-encoder architecture that the pre-trained query encoder E_c and document encoder E_d encode the claim $\mathbf{x}$ and each document $d_j \in \mathcal{D}$, respectively. Documents are ranked by the similarity score $s(\mathbf{x}, d_j) = E_c(\mathbf{x})^\top E_d(d_j)$ that is calculated by taking the dot product of the embeddings of the claim $\mathbf{x}$ and document d_j.

In the best current method **JustiLM** [11], only the parameters corresponding to the query encoder are updated while the document encoder remains frozen. To improve this method, we propose to unfreeze the document encoder. This approach allows us to introduce two retriever losses instead of one, as in [11].

The first loss is based on the similarity between the entire fact-checking article $\mathbf{z}$ and the retrieved documents D_N. We use the trainable query encoder E_c to represent $\mathbf{z}$ by aggregating the embeddings of all its chunks and obtain $\bar{E}_c(\mathbf{z}) = \frac{1}{M}\sum_{i=1}^{M} E_c(\mathbf{z}_i)$. The training objective is to minimize the MSE loss between the embeddings of $\mathbf{z}$ and d_i, we name it $Loss_{mse}$: $L = \frac{1}{N|\bar{E}_c(\mathbf{z})|}\sum_{j=1}^{N}||\bar{E}_c(\mathbf{z}) - E_d(d_j)||_2^2$.

We use a second loss, $Loss_{ca}$, to train the document encoder, based on the Cauchy loss [40] between two distributions: the attackability score distribution between the generated summaries and the mean attackability of retrieved documents D_N for each summary: $L = \sum_{i=1}^{n}\log\left(1 + \left(\frac{y_i - \hat{y}_i}{\sigma}\right)^2\right)$. This allows us to leverage the supervisory signals from the generation process. We use Cauchy loss as we observe a heavy-tailed distribution for the attackability scores of retrieved documents and Cauchy loss provides strong resistance to gross outliers

while being as efficient as least squares for Gaussian errors [41]. Ten documents are retrieved for each claim.

For generation, $Loss_{ce}$ has been used to compare the predicted probability distribution of words in the generated summaries with the actual target distribution, while $Loss_{mse}$ is used to compare the attackability scores of the groundtruth summary and the generated summary. We use **SumArgAttack** as the language model backend.

D. The claim verification module

We use the state-of-the-art method in fake news identification for the claim verification module as in [42]. First, we concatenate each claim and justification by inserting [SEP] between the two. A special token [CLS] is then added to the beginning of each sentence pair, from which the final embedding of the input is extracted. We construct the entity graph of each concatenated text based on Wikidata and then extract the graph embedding using graph attention networks [43]. A softmax layer is applied to the concatenation of graph and textual embeddings to get the logits for each label. We name our full pipeline (summarizer, evidence retrieval module and claim verification module) **ArgLM**.

E. Baselines

At the time of writing, two approaches produce the best results using ROUGE scores as metrics: the claim driven abstractive-extractive method (**CDAE**) of [10] with a human-written fact-checking article given as input and no evidence retrieval module, and **JustiLM** [11] where evidence is directly retrieved from a corpus. Both methods are strong baselines since they have been extensively evaluated against well-performing models such as GPT-4 [44].

VI. Evaluation

To evaluate our summarizers (**SumArg** with only $Loss_{ce}$ and **SumArgAttack** with supplementary $Loss_{mse}$ based on attackability scores) against **CDAE**, we performed a 10-fold cross-validation on the whole LIAR-PLUS dataset excluding LIARArg, FNC-1, and Check-Covid. We used mean F1 scores for the evaluation of the claim verification module. Mean ROUGE scores (R1, R2 and RL) are adopted for LIAR-PLUS, comparing generated summaries against human-written ones.

To evaluate our full pipeline against **JustiLM** [11], we performed a 10-fold cross-validation on ExClaim. Mean F1 scores (macro F1 when there are multiple labels) of the claim verification module and mean ROUGE scores are adopted.

A. Results

Table III reports the F1 scores of the claim verification module on the three datasets depending on the summarizer. The two-sided Wilcoxon signed rank test has been performed across the 10 crossfolds, and statistically significant improvements have been highlighted in bold. Note that ground-truth summaries for LIAR-PLUS and original articles for FNC-1 and Check-Covid (**Ground**) have also been tested as input for comparison.

TABLE III
F1 scores of the claim verification system of various summarizers on LIAR-PLUS, FNC-1 and Check-Covid.

Method	LIAR-PLUS	FNC-1	Check-Covid
Ground	0.51	0.90	**0.76**
CDAE	0.41	0.76	0.62
SumArg	0.48	0.85	0.69
SumArgAttack	**0.54**	**0.92**	0.74

Our results first demonstrate that argument-structured summaries significantly improve the performance of the claim verification module compared to the state-of-the-art summarization method for fact-checking (**SumArg**, **SumArgAttack** vs. **CDAE**), both on in-domain and out-domain data. This confirms our hypothesis that the current SOTA summarization approach is not perfectly tailored for fact-checking.

The comparison between **Ground** and **SumArgAttack** shows that by integrating attackability-based loss function, our argument-structured summaries yield even better results on LIAR-PLUS and FNC-1 than original human-written texts. Since we use the latest claim verification algorithm, this indicates that the performance of the best current claim verification method can be further boosted by using argument-enhanced summaries instead of human-written summaries. The first explanation could be that the generated summaries contain more comprehensive and structured evidence, while human-written summaries have limited evidence coverage. Example (2) shows the evidence (4 premises) provided by **SumArgAttack** versus human-written summary (2 premises) for the claim "Hillary Clinton supported NAFTA and permanent China trade".

(2) **Evidence provided by SumArgAttack:** 1) she said NAFTA had some positive effects "but unfortunately it had a lot of downside."; 2) Both promised to crack down on China's practice of manipulating its currency to give its products an unfair advantage. 3) Both said they opposed the Chinese government subsidizing industry to the detriment of U.S. competitors. 4) At a debate in December 2007, she announced her intention to review and reform NAFTA if she were elected.
Evidence provided in the human-written summary of LIAR-PLUS: 1) Clinton has in the past verbally supported NAFTA and permanent trade with China; 2) she has spoken forcefully about the need to reform NAFTA and to much more stringently enforce trade agreements with China.

The second reason could be that **SumArgAttack** generates summaries containing explicit fine-grained argument relations (partial support and partial attack), especially useful in the case of half-truths [45] such as half-true and mostly-false. Indeed, the two datasets on which **SumArgAttack** outperforms human-written texts have both multiple labels (6 for LIAR-PLUS and 4 for FNC-1). After further analysis of F1 scores, we observe that the most significant difference is on half-true and mostly-true labels on LIAR-PLUS according to

the input of the claim verification module (0.41 and 0.32 for **SumArgAttack**, 0.30 and 0.23 for **CDAE**, and finally, 0.36 and 0.28 for **Ground**). As for FNC-1, the F1 score for the intermediate label *discuss* is 0.33 for **SumArgAttack**, 0.24 for **CDAE** and 0.29 for **Ground**.

Finally, it is important to note that integrating attackability-based loss function enhances significantly the performance of claim verification (**SumArgAttack** *vs.* **SumArg**). Our explanation is that introducing this loss function reduces the hallucination of attack-support relations in the generated summaries. This is evidenced by the average number of relations dropping from 7 to 3 in LIAR-PLUS. Given that the average number of relations in LIARArg is 2.5, **SumArgAttack** produces summaries with a more trustworthy argumentative structure.

Table IV reports ROUGE scores for the summaries produced by **CDAE** and our summarizers on LIAR-PLUS versus human-written summaries. Along with the results reported in Table III, it can be observed that ROUGE scores of our argument-structured summaries are lower, as these summaries are less similar to those manually written by humans. However, higher ROUGE scores don't lead to better performance of the claim verification module. This highlights the fact that argument-structured summaries, although dissimilar to human-written summaries, are more suitable for the deep-learning algorithms used for the fack-checking task. Summary generation in the context of claim verification should therefore not be evaluated solely by overlap statistics such as ROUGE scores.

TABLE IV
ROUGE SCORES FOR GENERATED SUMMARIES COMPARED WITH HUMAN-WRITTEN SUMMARIES ON LIAR-PLUS.

Summarizer	R1	R2	RL
CDAE	0.348	0.159	0.272
SumArg	0.273	0.130	0.158
SumArgAttack	**0.268**	**0.128**	**0.143**

Table V reports the F1 scores of the claim verification module of **JustiLM** and **ArgLM** where the evidence retrieval module is fully automated. ROUGE scores of summaries produced by **JustiLM** and **ArgLM** are also reported. When compared on ExClaim, our pipeline **ArgLM** achieves an F1 score of **0.73**, significantly higher than 0.65 of **JustiLM** [11]. It is worth noting that with human-written explanations, the F1 score is 0.72, confirming that our method achieves comparable precision to hand-written justifications even in a fully automated fact-checking pipeline. ROUGE scores further confirm the insufficiency of overlap-based metrics for evaluating fact-checking-oriented summarization.

TABLE V
F1 SCORES OF THE CLAIM VERIFICATION SYSTEM AND ROUGE SCORES OF GENERATED SUMMARIES COMPARED WITH HUMAN-WRITTEN SUMMARIES ON EXCLAIM.

Pipeline	F1	R1	R2	RL
JustiLM	0.65	0.376	0.189	0.343
ArgLM	**0.73**	**0.298**	**0.156**	**0.223**

Finally, the average number of argument relations rises from 3 to 8 when all the attackability-based loss functions are removed, confirming again the relevance of our approach.

B. Error analysis

We randomly selected a sample of 50 instances to analyze cases where **ArgLM** fails to predict the label correctly. When no correct evidence is retrieved, an error rate of 80% is observed, compared to a 30% error rate when at least one correct piece of evidence is retrieved. This result is expected, as accurate evidence is crucial for meaningful summarization.

When at least one correct piece of evidence is retrieved (38 instances), we check the relations and find that when all the argument relations are wrong, the error rate is 65% *vs.* 25% when at least one argument relation is correct. This confirms that the claim verification module does exploit the argument structure of generated summaries.

Example (3) shows an example where a wrong relation is generated in the summary for the claim "Donald Trump has changed his mind on abortion", leading to a wrong label prediction. This is a typical example highlighting the necessity to consider temporal factors in argument relations. The claim "changed his mind" can only be attacked or supported by considering a sequence of events over time. It is thus impossible to determine the exact relation in this case.

(3) **Summary**: ""I am very pro-choice," Trump said. "I hate the concept of abortion." **attacks** the claim, "So I'm pro-life, but with the caveats." **partially supports** the claim...
Label: True
Predicted Label: mixture

VII. CONCLUSIONS

The main contributions of this work are: *1)* we introduce a novel argument-structured dataset and summarization technique for justification generation yielding higher precision than SOTA methods and human-written summaries when used in automatic claim verification; *2)* we demonstrate that incorporating attackability scores in summarization significantly enhances performance and reduces hallucinated argument relations; *3)* we highlight that overlap-based metrics like ROUGE scores, are insufficient for evaluating summarization in the context of automated fact-checking; *4)* our pipeline combining evidence retrieval and argument-structured summarization achieves superior results compared to SOTA approach. Our work demonstrates the effectiveness of argument-based justifications for claim verification. For future work, we plan to integrate partial relations into the attackability score and enhance explainability about attacks from evidence to claims, with a focus on relations involving temporal factors. Finally, additional language models will be tested to further show the generalizability of our method.

REFERENCES

[1] J. Thorne and A. Vlachos, "Automated Fact Checking: Task formulations, methods and future directions," Sep. 2018.

[2] M. A. Amazeen, "Revisiting the Epistemology of Fact-Checking," *Critical Review*, vol. 27, no. 1, pp. 1–22, Jan. 2015.

[3] C. O'neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown, 2017.

[4] S. Lewandowsky, U. K. H. Ecker, C. M. Seifert, N. Schwarz, and J. Cook, "Misinformation and Its Correction: Continued Influence and Successful Debiasing," *Psychological Science in the Public Interest*, vol. 13, no. 3, pp. 106–131, Dec. 2012.

[5] I. Eldifrawi, S. Wang, and A. Trabelsi, "Automated justification production for claim veracity in fact checking: A survey on architectures and approaches," *arXiv preprint arXiv:2407.12853*, 2024.

[6] Z. Guo, M. Schlichtkrull, and A. Vlachos, "A Survey on Automated Fact-Checking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 178–206, Feb. 2022.

[7] B. Adair, C. Li, J. Yang, and C. Yu, "Progress toward "the holy grail": The continued quest to automate fact-checking," in *Computation+ Journalism Symposium,(September)*, 2017.

[8] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004, pp. 74–81.

[9] T. Alhindi, S. Petridis, and S. Muresan, "Where is Your Evidence: Improving Fact-checking by Justification Modeling," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 85–90.

[10] D. Russo, S. S. Tekiroğlu, and M. Guerini, "Benchmarking the Generation of Fact Checking Explanations," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1250–1264, Oct. 2023.

[11] F. Zeng and W. Gao, "JustiLM: Few-shot Justification Generation for Explainable Fact-Checking of Real-world Claims," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 334–354, Apr. 2024.

[12] J. Ma, W. Gao, S. Joty, and K.-F. Wong, "Sentence-level evidence embedding for claim verification with hierarchical attention networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019.

[13] V. Dua, A. Rajpal, S. Rajpal, M. Agarwal, and N. Kumar, "I-FLASH: Interpretable Fake News Detector Using LIME and SHAP," *Wireless Personal Communications*, pp. 1–34, 2023.

[14] N. Vedula and S. Parthasarathy, "FACE-KEG: Fact Checking Explained using KnowledgE Graphs," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. Virtual Event Israel: ACM, Mar. 2021, pp. 526–534.

[15] A. Krishna, S. Riedel, and A. Vlachos, "Proofver: Natural logic theorem proving for fact verification," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 1013–1030, 2022.

[16] N. Kotonya and F. Toni, "Explainable Automated Fact-Checking for Public Health Claims," Oct. 2020.

[17] Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders," Sep. 2019.

[18] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[19] S. Shleifer and A. M. Rush, "Pre-trained Summarization Distillation," Oct. 2020.

[20] K. Khan, R. Wang, and P. Poupart, "WatClaimCheck: A new dataset for claim entailment and inference," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1293–1304.

[21] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," Sep. 2020.

[22] B. M. Yao, A. Shah, L. Sun, J.-H. Cho, and L. Huang, "End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 2733–2743.

[23] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Atlas: Few-shot learning with retrieval augmented language models," *Journal of Machine Learning Research*, vol. 24, no. 251, pp. 1–43, 2023.

[24] W. Y. Wang, ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 422–426.

[25] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[26] M. Stede and J. Schneider, "Argumentation mining," *Synthesis Lectures on Human Language Technologies*, vol. 11, no. 2, pp. 1–191, 2018.

[27] T. Mayer, E. Cabrio, M. Lippi, P. Torroni, and S. Villata, "Argument Mining on Clinical Trials," p. 12.

[28] A. Hanselowski, A. PVS, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance-detection task," in *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1859–1874. [Online]. Available: https://aclanthology.org/C18-1158

[29] G. Wang, K. Harwood, L. Chillrud, A. Ananthram, M. Subbiah, and K. McKeown, "Check-COVID: Fact-Checking COVID-19 News Claims with Scientific Evidence," May 2023.

[30] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Knight, A. Nenkova, and O. Rambow, Eds. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1163–1168. [Online]. Available: https://aclanthology.org/N16-1138

[31] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," Oct. 2023.

[32] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," May 2023.

[33] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints," Dec. 2023.

[34] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating Long Sequences with Sparse Transformers," Apr. 2019.

[35] Y. Jo, S. Bang, E. Manzoor, E. Hovy, and C. Reed, "Detecting Attackable Sentences in Arguments," Oct. 2020.

[36] M. Alshomary, S. Syed, A. Dhar, M. Potthast, and H. Wachsmuth, "Counter-Argument Generation by Attacking Weak Premises," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1816–1827.

[37] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," Feb. 2021.

[38] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Unsupervised Dense Information Retrieval with Contrastive Learning," Aug. 2022.

[39] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[40] P. J. Huber, "Robust Estimation of a Location Parameter," in *Breakthroughs in Statistics*, S. Kotz and N. L. Johnson, Eds. New York, NY: Springer New York, 1992, pp. 492–518.

[41] J. T. Barron, "A more general robust loss function," *arXiv preprint arXiv:1701.03077*, vol. 7, 2017.

[42] J. Ma, C. Chen, C. Hou, and X. Yuan, "KAPALM: Knowledge grAPh enhAnced Language Models for Fake News Detection," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3999–4009.

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," Feb. 2018.

[44] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[45] A. Estornell, S. Das, and Y. Vorobeychik, "Deception through half-truths," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 10 110–10 117.