End-to-End Multimodal Fact-Checking and Explanation Generation: A Challenging Dataset and Models

Barry Menglong Yao  
Virginia Tech  
Blacksburg, USA  
barryyao@vt.edu

Aditya Shah  
Virginia Tech  
Blacksburg, USA  
aditya31@vt.edu

Lichao Sun  
Lehigh University  
Bethlehem, USA  
lis221@lehigh.edu

Jin-Hee Cho  
Virginia Tech  
Blacksburg, USA  
jicho@vt.edu

Lifu Huang  
Virginia Tech  
Blacksburg, USA  
lifuh@vt.edu

ABSTRACT

We propose end-to-end multimodal fact-checking and explanation generation, where the input is a claim and a large collection of web sources, including articles, images, videos, and tweets, and the goal is to assess the truthfulness of the claim by retrieving relevant evidence and predicting a truthfulness label (e.g., support, refute or not enough information), and to generate a statement to summarize and explain the reasoning and ruling process. To support this research, we construct Mocheg, a large-scale dataset consisting of 15,601 claims where each claim is annotated with a truthfulness label and a ruling statement, and 33,880 textual paragraphs and 12,112 images in total as evidence. To establish baseline performances on Mocheg, we experiment with several state-of-the-art neural architectures on the three pipelined subtasks: multimodal evidence retrieval, claim verification, and explanation generation, and demonstrate that the performance of the state-of-the-art end-to-end multimodal fact-checking does not provide satisfactory outcomes. To the best of our knowledge, we are the first to build the benchmark dataset and solutions for end-to-end multimodal fact-checking and explanation generation. The dataset, source code and model checkpoints are available at https://github.com/VT-NLP/Mocheg.

CCS CONCEPTS

• Computing methodologies → Natural language processing; Natural language generation; Computer vision; • Information systems → Multimedia and multimodal retrieval.

KEYWORDS

Multimodal Fact-Checking; Evidence Retrieval; Stance Detection; Explanation Generation; Explainable Fact-Checking

ACM Reference Format:
Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang, 2023. 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 (SIGIR ’23), July 23–27, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591879

1 INTRODUCTION

Misinformation has been a growing public concern in society and caused difficulty in finding reliable information online [15, 20]. For example, as Islam et al. [28] shows, the misinformation about COVID-19 has widely spread and led people to distrust medical treatment and even refuse to get vaccinated. The situation has become even more complicated with the emergence of large language models, like ChatGPT [46] since they could be intentionally misused to generate misinformation [21] or wrongly spread misinformation due to the hallucination issue [77]. To fight against misinformation, many fact-checking websites, such as Snopes3 and PolitiFact2, have been created where journalists manually collect thousands of claims from news and social media and verify them by referring to external reliable and relevant documents. However, it is time-consuming and hard to generalize to more broad claims.

Recently, researchers have started to investigate automatic misinformation detection and fact-checking by developing various

3https://www.snopes.com/
4https://www.politifact.com/
benchmark datasets [43, 47, 58, 64, 70] and start-of-the-art neural network architectures [38, 60, 63, 76]. However, we found the following limitations with the current fact-checking studies: (1) Most of them only consider text while ignoring the multi-media nature (e.g., images) of online articles, which are essential and useful to predict the truthfulness of claims. There are a few multimodal fact-checking datasets existing [1, 41, 45], however, their truthfulness labels [41] or evidence [41, 45] are automatically generated and thus cannot be guaranteed to be consistent with human judgements. (2) While current studies simply predict a truthfulness label, it is also necessary to provide a textual statement to explain the prediction. These explanations are vital to justify how the conclusion is reached step by step based on external evidence, and provide the public with rationale to analyze the reasoning process and share it with others. (3) Some prior studies [55, 70, 78] assume that a short piece of evidence text is already identified, based on which the models can directly predict the truthfulness of the target claim. However, this is not realistic in practice as the claim does not come with evidence, which should be retrieved from a knowledge base or the Internet.

To tackle these challenges, we propose end-to-end multimodal fact-checking and explanation generation, where the input consists of a claim and a large collection of web sources, including articles, images, and tweets, and the goal is to automatically retrieve information sources relevant to the claim (Evidence Retrieval), predict the truthfulness of the claim based on the relevant evidence (Claim Verification), and generate a textual explanation to explain the reasoning and ruling process (Explanation Generation). An example is shown in Figure 1. To support this research, we introduce Mocheg, a new benchmark dataset with 15,601 claims annotated with truthfulness labels, multimodal evidence, and ruling statements, along with a large collection of web articles and images as the evidence sources. To set up the baseline performance, we explore the state-of-the-art pre-trained vision-language models for multimodal evidence retrieval, claim verification, and explanation generation. Experimental results show that there is still a huge room for further improvements in this end-to-end multimodal fact-checking and explanation generation task. Overall, the contributions of our work are as follows:

- To the best of our knowledge, this is the first study that investigates end-to-end multimodal fact-checking and explanation generation task.
- We create the first benchmark dataset for end-to-end multimodal fact-checking and explanation generation. The baseline performance of the state-of-the-art language models demonstrates that the task is still challenging, and there is a huge space to improve.

2 RELATED WORK

Multimodal Fake News Detection and Fact-checking: Most previous benchmark datasets [2, 3, 5, 25, 35, 57, 64, 70] for fake news detection and fact-checking are mainly based on text. As information is naturally in multi-modality, recent studies have started to take images [8, 18, 30, 43, 51, 55, 58, 78] and videos [40, 48, 52] into consideration. Many methods for multimodal fake news detection are based on cross-modality consistency checking [1, 56, 60, 63, 71, 76] or computing a fused representation of multimodal (textual + visual) information for final classification [31, 33, 61, 68], [43, 55, 78] directly predict the truthfulness of multimodal claims without considering explicit evidence. [1, 41, 45] are the most related work to ours in that it considers explicit multimodal evidence. However, their labels or evidence are automatically generated without validating by humans while our label and evidence are annotated by fact-checking journalists. And we further provide journalists explanations regarding the truthfulness prediction. Compared with all these studies, our Mocheg is designed for the end-to-end multimodal fact-checking and explanation generation that requires systems to automatically retrieve multimodal evidence to predict the truthfulness of each claim and generate a ruling statement to explain the reasoning and ruling process. Table 1 compares Mocheg with mentioned datasets.

Explainable Fact-Checking: Providing explanations to the model predictions is beneficial for humans to understand the truthfulness of the claims [22–24, 65, 66]. Current explainable fact-checking studies can be divided into four categories. The first is to directly take the evidence used for claim verification as the explanation [2, 16, 25, 64]. However, the evidence usually consists of several individual sentences extracted from a large collection of documents, which are not logically connected and thus might be hard for humans to interpret. The second is to incorporate external knowledge graphs to compute a set of semantic traces starting from the claim [19]. The semantic traces can serve as explanations to justify the truthfulness of the claims. The third is to generate questions based on claims and link the claims and evidence by using these questions as a proxy [9, 11, 73]. Although these generated questions can improve the explainability, they may be similar or less relevant because, normally, the claim is short. The fourth is to apply natural language generation to generate a paragraph describing the reasoning process [4, 32, 35, 62, 75], which is the most interpretable to humans. Previous studies usually summarize fact-checking articles written by journalists in shorter paragraphs as explanations. In stark contrast, our work generates explanations based on the evidence that is automatically retrieved from the web, which is more realistic in practice. In addition, in our end-to-end multimodal setting, the system needs to sequentially or jointly perform all three sub-tasks, including multimodal evidence retrieval, multimodal claim verification, and multimodal explanation generation.

3 DATASET CONSTRUCTION

3.1 Data Source

PolitiFact and Snopes are two widely used websites to fight against the spreading of misinformation, where journalists are asked to manually check and verify each claim and write a ruling article to share their judgment. Considering this, we use these two websites as the data sources. Specifically, we develop scripts based on [25] to collect all the necessary information from these two websites, including the claims that are purely based on text, truthfulness labels, text and/or image evidence that is extracted from external articles by journalists and help determine the truthfulness of claims.

---

3The example is from https://www.politifact.com/factchecks/2021/may/13/andrew-clyde/ridiculous-claim-those-capitol-jan-6-resembled-not/

4We have obtained permission from both Snopes and PolitiFact to publish the data for the research purpose.
Table 1: Comparison between Mocheg and other related datasets. The columns indicate whether the dataset requires automatic evidence retrieval, multimodal reasoning, or explanation generation and whether its label and evidence are annotated by a human.

evidence references that are linked to external articles/images containing the text and image evidence, and ruling articles that can explain and justify the truthfulness of the claims and can be viewed as a short summary of the various evidence. Note that, the claims were originally manually collected by the journalists of the two websites from many sources, e.g., online speeches, public statements, news articles, and social media platforms, such as Facebook, Twitter, Instagram, TikTok, and so on. The truthfulness labels, evidence, evidence references, and ruling articles are also manually provided by fact-checkers of the two websites.

Based on the evidence references, we further develop scripts to collect the articles and images that contain the evidence. Since the evidence references are linked to thousands of websites with distinct HTML templates, we utilize Boilerpipe [34] to extract the text and newspaper\(^8\) to obtain all image links contained in the webpages and download the images based on urllib\(^7\). Some evidence references are linked to Twitter. To collect them, we first extract the Tweet IDs from the URLs of evidence references and then apply Twitter API\(^8\) to collect the text and images from the corresponding Tweets.

3.2 Data Preprocessing

Since fact-checking websites adjust their labels over time, the initial data contains more than 75 truthfulness labels, and some labels overlap with each other, such as “True”, “TRUE”, and “Status: True”. Also, some labels have only a few instances. For example, the label “Labeled Satire” has only 23 instances in total. Considering these, we follow [25] and map 68 of these labels into three general categories, including Supported, Refuted, and NEI (Not Enough Information). We remove the claims that are annotated with other labels. In this way, each claim is just assigned one of the three target labels. The initial dataset also contains a lot of advertisement images. To clean the dataset, we design several rules, including (1) removing an image if its name contains any of the keywords, including “-ad-”, “logo”, “gif”, “.ico”, “lazyload”, “.gif”, “Logo”, “.php”, “icon”, “Bubble”, “svg”, “rating-false”, “rating-true”, “banner”, “-line”, or its size is smaller than 400×400; (2) removing a claim if we can not crawl any evidence or the ruling article; (3) for each ruling article, there is usually a paragraph starting with “Our ruling” or “In sum” which summarizes the whole ruling and reasoning process to achieve the fact-checking conclusion, thus we use this paragraph as the target explanation. As a result, we collect 15,601 claims with 33,880 text evidence, where each piece of text evidence is an individual paragraph extracted from a particular evidence reference article and 12,112 image evidence\(^9\). Based on the evidence references, we finally collect 91,822 articles and 122,246 images which are further combined to form a constant collection of web resources for the evidence retrieval task. Within the web source collection, only 30% (27,566 out of 91,822) of articles and 10% (12,112 out of 122,246) of images contain the evidence of claims, making the evidence retrieval task realistic and challenging enough.

3.3 Task Definition

We name the dataset Mocheg and propose End-to-End\(^10\) Multimodal Fact-Checking and Explanation Generation, with three subtasks\(^11\):

**Task 1. Multimodal Evidence Retrieval**: Given a claim and a collection of web sources containing both documents and images, the Evidence Retrieval task is to determine which paragraphs contained in the documents and images are related to the claim and can be further used to determine the truthfulness of the claim.

**Task 2. Multimodal Claim Verification**: Based on the text and image evidence retrieved in Task 1, the Multimodal Claim Verification task is to predict the truthfulness (Supported, Refuted, or NEI) of the claim. As both the input claim and retrieved evidence may contain both text and images, this task requires cross-modal reasoning.

\(^8\)We illustrate the detailed fact-checking processing in Snopes and Politifact in Section 8.

\(^9\)https://newspaper.readthedocs.io/en/latest/

\(^10\)https://docs.python.org/3/library/urllib.html

\(^11\)https://developer.twitter.com/en/docs/api-reference-index

\(^8\)Among the 15,601 claims, 19% of them have tweets as evidence while the remaining 81% only use other sources such as news articles or government reports as evidence. Note that the image and text evidence may be from separate sources with no clear association.

\(^10\)The end-to-end setting in our fact-checking task means it starts with only the claim and goes through the evidence retrieval, claim verification, and explanation generation, which is almost the complete pipeline for a journalist to do fact-checking in real life.

\(^11\)Note that we don’t consider claim extraction as a subtask as all the input claims are considered worthy of being checked.
Figure 2: Overview of framework, consisting of a text evidence retrieval module (top left), an image evidence retrieval module (bottom left), a claim verification module (bottom right), and an explanation generation module (top right).

Task 3. Explanation Generation: Given an input claim, the evidence retrieved from Task 1, as well as the truthfulness predicted from Task 2, the Explanation Generation task aims to generate a paragraph that summarizes the evidence based on the predicted truthfulness label and explains the ruling process.

3.4 Train / Dev / Test Split

We split the whole dataset into training (Train), development (Dev), and test (Test) sets with the percentage of 75%, 10%, and 15%, respectively. Table 2 shows the detailed statistics for each split.

| Data                        | Train | Dev | Test |
|-----------------------------|-------|-----|------|
| # Claims                    | 11,669| 1,490| 2,442|
| Ave. # Tokens in Claim      | 20    | 20  | 21   |
| Max. # Tokens in Claim      | 81    | 77  | 89   |
| # Text evidence (Paragraphs)| 23,545| 4,067| 6,268|
| # Image evidence            | 8,927 | 1,178| 2,007|
| # Refuted Labels            | 4,542 | 488 | 825  |
| # Supported Labels          | 3,826 | 501 | 817  |
| # NEI Labels                | 3,301 | 501 | 800  |
| Ave. # Tokens in Explanation| 132   | 90  | 105  |
| Max. # Tokens in Explanation| 600   | 521 | 600  |

Table 2: Dataset statistics of MocheG.

4 APPROACH

To establish the baseline performance on MocheG, we design a framework for End-to-End Multimodal Fact-checking and Explanation Generation. As illustrated in Figure 2, it consists of three components for the corresponding sub-tasks.

4.1 Evidence Retrieval

To solve this task, we apply two baseline models to retrieve text and image evidence separately.

Text Evidence Retrieval: The top left in Figure 2 illustrates the approach for text evidence retrieval. Given an input claim and a document corpus, we first split each document into sentences and then apply SBERT (Sentence-BERT) [53, 54] to take in the input claim and a sentence from the document corpus and output their contextual representations, based on which we can further compute a cosine similarity score for each pair. Based on these similarity scores, we rank all the sentences and select the top-1000 as the candidate evidence. We fine-tune the SBERT based on the following InfoNCE loss [67]:

\[
    L(C_i, T^P, T) = -\log \left( \frac{\exp(\text{cosine}(C_i, T^P))}{\sum_{T_j \in T} \exp(\text{cosine}(C_i, T_j))} \right)
\]

where \(T^P\) is a piece of positive evidence to a claim \(C_i\), \(T\) contains \(T^P\) and a set of other negative evidence to \(C_i\). For each claim, we use the evidence of other claims in the same batch as the negative ones. \(C_i, T^P\) and \(T_j\) are the sentence level representations encoded from SBERT. In this work, we use bold symbols to denote vector representations.

We further apply a re-ranking model based on BERT [12], which encodes each pair of the input claim and a candidate evidence sentence and outputs a score based on a linear classification layer. Based on these scores, we further rank all the candidate evidence and select the top-\(K\) as the text evidence. The BERT-based re-ranking model is pre-trained on the MS MARCO Passage Ranking dataset [6] which is designed for text retrieval.

Image Evidence Retrieval: As shown in the bottom left of Figure 2, given an input claim and the image corpus, we use CLIP [50] as
the encoder to learn an overall representation for the claim and a representation for each image, then compute the cosine similarity between each image and the input claim. We sort all the images in the corpus based on the cosine similarity scores and take the top-K as the candidate image evidence. We fine-tune CLIP based on the same InfoNCE loss as text evidence retrieval. Note that, during inference, we always retrieve top-K text and image evidence respectively though it’s possible that there is no image or text evidence contained in the background corpus.

4.2 Claim Verification

Based on the text and image evidence, we further design a claim verification approach to predict the truthfulness of each input claim, which is shown in the bottom right of Figure 2.

Encoding with CLIP. We formulate an input claim as \( C = \{c_0, c_1, ..., c_n\} \), a piece of text evidence as \( T_i = \{t_{i0}, t_{i1}, ..., t_{in}\} \), a piece of image evidence as \( M_j = \{m_{j0}, m_{j1}, ..., m_{jn}\} \), where \( c_k \) denotes the \( k \)-th token of the claim, \( t_{i0} \) is the \( k \)-th token of the \( i \)-th text evidence \( T_i \), and \( m_{jk} \) is the \( k \)-th patch of the \( j \)-th image evidence \( M_j \). Given a claim \( C \) and its text evidence \( \{T_0, T_1, \ldots\} \) and image evidence \( \{M_0, M_1, \ldots\} \), we concatenate them as an overall sequence \( \{C, T_0, T_1, \ldots, M_0, M_1, \ldots\} \) and feed it into CLIP to obtain their contextual representations:

\[
\begin{align*}
H_C & = [h_{c0}, h_{c1}, \ldots, h_{cn}], \\
H_{T_i} & = [h_{t_{i0}}, h_{t_{i1}}, \ldots, h_{tin}], \\
H_{M_j} & = [h_{m_{j0}}, h_{m_{j1}}, \ldots, h_{mjn}].
\end{align*}
\]

Stance detection: We then pair each piece of evidence with the input claim and detect the stance of the evidence towards the claim. As Figure 3 describes, taking text evidence as an example, we first compute an attention distribution between the claim and the evidence by using \( H_C = [h_{c0}, h_{c1}, \ldots, h_{cn}] \) as query, \( H_{T_i} = [h_{t_{i0}}, h_{t_{i1}}, \ldots, h_{tin}] \) as key and value to compute cross attention and obtain an updated claim representation \( H_{T_i2C} = [h_{t_{i0}}, h_{t_{i1}}, \ldots, h_{t_{iin}}, h_{c0}, h_{c1}, \ldots, h_{cn}] \) where \( h_{c1} \) is defined by:

\[
h_{c1} = \text{Softmax}(h_{c1} \cdot H_T^T) \cdot H_T \]

We then fuse the updated claim representation \( H_{T_i2C} \) with its original representation \( H_C \) by two arithmetic operations, subtraction (-) and multiplication (*), which work best as comparison functions in [69], and obtain the stance representation \( G_{T_i2C} \) of evidence \( T_i \) towards the claim \( C \) based on max pooling.

\[
\begin{align*}
\tilde{G}_{T_i2C} & = \sigma([H_{T_i2C} \cdot H_C : H_{T_i2C} - H_C] \cdot W_a + b_a), \\
G_{T_i2C} & = \text{MeanPooling}(\tilde{G}_{T_i2C}),
\end{align*}
\]

where \([::]\) denotes concatenation operation, \(W_a \) and \(b_a \) are learnable parameters for aggregating the representations, and \( \sigma \) denotes a LeakyReLU activation function.

Figure 3: Stance Detection

\[
\begin{align*}
\hat{y}_{cls} & = W_{h}^T \cdot [G_{T_i2C} : G_{M_j2C}] + b_h, \\
\mathcal{L}(y_i|C) & = -\log\left(\frac{\exp\left(\hat{y}_{cls,i}\right)}{\sum_{j=0}^{n} \exp\left(\hat{y}_{cls,j}\right)}\right)
\end{align*}
\]

where \( \hat{y}_{cls} \) denotes the probabilities over all possible labels. \( y_i \) is the truthfulness label of claim \( C \). During training, we fix the parameters of CLIP while tuning all the other parameters.

4.3 Explanation Generation

To justify the truthfulness prediction, we further generate a ruling statement by considering the input claim, the predicted truthfulness label as well as the text evidence. The top right of Figure 2 illustrates the overall architecture for explanation generation.

Specifically, given an input claim \( C \), its truthfulness label \( y_C \), and text evidence \( \{T_1, T_2, \ldots\} \), we concatenate them into an overall sequence \( X \) with a separator \( <s> \). Then we feed this sequence as input to BART [37], which is a state-of-the-art pre-trained sequence-to-sequence model, to generate a ruling statement \( S = \{s_1, s_2, \ldots, s_q\} \).

During training, we use the gold truthfulness label of each claim as input. During the evaluation, we use the truthfulness label predicted by the claim verification model. The training objective is to minimize the following negative log-likelihood based on the gold ruling statement \( \hat{S} = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_q\} \):

\[
\mathcal{L}_g = -\sum_{i=1}^{q} \log(p(\hat{s}_i|\hat{s}_{i-1}, X; \phi))
\]

To ensure the generated ruling statement is consistent with the truthfulness label of the claim, we apply a truthfulness reward and optimize the generation model with reinforcement learning (RL) [36]. Specifically, we pre-train a truthfulness classification model based on BERT [15], which takes the ruling statement as input and outputs a confidence score for each truthfulness label. We use the difference between the confidence score of the correct label and the score of the wrong labels as the reward \( R_{cls} \):

\[
R_{cls} = p(y_C|S) - \sum_{y_j \neq y_C, y_j \leq Y} p(y_j|S),
\]

where \( y_C \) is the gold truthfulness label of \( C \), \( Y \) is the target label set, and \( S \) is the generated ruling statement.

We then apply the reward \( R_{cls} \) for policy learning, and the policy gradient is computed as:

\[
\nabla_{\theta} J(\theta) = \mathbb{E}[R_{cls} \cdot \nabla_{\theta} \sum_{i} \log(p(s_i|s_{i-1}, X; \phi))].
\]

\[\]Since the evidence in our corpus is annotated by journalists on Politifact and Snopes, we assume the evidence is reliable and fuse the stance of evidence to the claim to predict the truthfulness. We leave it as a future work to check the trustworthiness of evidence.
where $X$ is the concatenated sequence of the input claim, its truthfulness label, and text evidence, and $\phi$ denotes the model parameters.

5 EXPERIMENTS

5.1 Evidence Retrieval

For each claim, we retrieve the top-K text and image evidence from the corresponding text and image corpus and evaluate the retrieval performance based on Precision, Recall, NDCG [29], MAP (Mean Average Precision), and S-Recall (Similarity-based Recall) scores. In S-Recall, it first computes a recall score for each gold text or image evidence based on the highest cosine similarity score between it and all retrieved text or image evidence, while each piece of evidence is represented with a vector learned from SBERT or CLIP. We use the average recall of all gold evidence as the S-Recall.

![Table 3: Performance of text and image evidence retrieval.](image)

We show the performance of text and image evidence retrieval on the text set of MOCHEG in Table 3. We can see that the performance of both image and text evidence retrieval is low, indicating the difficulty of both tasks. Taking text evidence retrieval as an example, the model needs to retrieve 2 pieces of text evidence on average for each claim from a collection of 2,792,639 sentences, which is very challenging. Also, the proposed evidence retrieval is based on semantic matching. However, in many cases, it is more important to find evidence that is relevant to the claim but describes different aspects or is against the claim, especially for refuted claims. For example, given an input claim, “H.R. 6666 provides $100 billion to entities that perform COVID-19 testing but prohibits them from allowing any non-vaccinated persons into their facilities.” the retrieval model missed an important piece of evidence “No provision in this bill would make testing or quarantining mandatory.”. This is against the claim and has lower similarity compared with the retrieved text “It would provide $100 billion to organizations that do COVID-19 testing or contact tracing or that provide services to people who are isolated at home.”. In addition, for many claims, their evidence come from the comprehension of long paragraphs instead of several sentences. Although our approach successfully retrieves several relevant sentences, they are insufficient to cover all the background and indicate the truthfulness of the claims.

5.2 Claim Verification

For claim verification, we first design two common baselines: (1) Majority Label, which predicts the majority label (i.e., Refuted) in the Training set for all the claims in the Test set; and (2) Average Similarity, which computes average cosine similarity between the target claim and all the gold text and image evidence based on their embeddings learned from CLIP. If the average similarity is higher than $\alpha_1 \in \{0.5, 0.6, 0.7, 0.75, 0.8\}$, predict it as Supported; if the average similarity is lower than $\alpha_2 \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.65, 0.7\}$ and $\alpha_2 < \alpha_1$, predict it as Refuted; otherwise, predict it as NEL. We search for the best value of $\alpha_1$ and $\alpha_2$ on the Development set and then apply them to the Test set. We then adapt Pre-CoFactv2 [14], a multimodal fact-checking model which achieves state-of-the-art results at the Factly 2 challenge [42] at AAAI 202314, to be the third baseline model. As there is very little existing work on multimodal fact-checking, we further adapt SpotFakePlus [59], a multimodal fake news detection approach, to our fact-checking task, by using their model to compare the consistency of input claim and image evidence and adding a new component to check the consistency of input claim and text evidence15. As shown in Table 4, Majority Label and Average Similarity yield a performance score that is close to a random baseline, while Pre-CoFactv2 and SpotFakePlus underperform our approach, demonstrating that MOCHEG does not contain any label distribution bias and cannot be easily solved simply by comparing the semantics between claims and evidence.

![Table 4: Performance of claim verification.](image)

To evaluate the impact of each type of evidence to claim verification, we design ablated models of our approach by considering the text evidence only, image evidence only, or no evidence. In addition, we compare its performance based on the system-retrieved evidence and the gold evidence to show the impact of evidence retrieval. As shown in Table 4, without considering any evidence, the model can still outperform the majority based baseline on claim verification due to the fact that some claims, such as “Paying taxes is optional!” contain obvious clues or are against common sense so that the model can directly predict the truthfulness based on the claim itself. By adding text and/or image evidence, the performance of claim verification can be boosted, proving the usefulness of the evidence. The text evidence provides more significant gain than image evidence due to two reasons: (1) for about 32% of the claims (787 out of 2,442) in the Test set, they only have text evidence

14https://ai.sci.aaai.org/2023/abstract/paper/14.pdf
15Most of existing multimodal fake news detection studies aim to detect fake news by comparing the consistency between news text and news image or between the news articles and external knowledge graphs, thus cannot be directly applied to fact-checking task.
without any associated image evidence. However, our approach always returns the top-5 most relevant images as evidence, introducing noises; (2) Texts usually carry more information than images. However, we also observe many examples that the image evidence complements the text evidence. For example, for claim #1 “A Boeing B-17E bomber from World War II was found in the jungle” in Figure 4, its image evidence plays a crucial role in confirming “the aircraft was found in the jungle.”

Finally, we also set up a human performance for claim verification by randomly sampling 50 claims and asking two annotators to label truthfulness by providing gold evidence, system evidence, or no evidence, which reach a Fleiss κ score [17] of 0.67, 0.59, and 0.42, respectively. We take a human prediction as true only if both of the two annotators provide the true label. As we can see, there is still a significant gap between machine and human performance.

5.3 Explanation Generation
We fine-tune BART based on a pre-trained bart-large16 checkpoint [72] to generate the ruling statement. We use ROUGE [39], BLEU [49], and Bertscore [74] as the evaluation metrics. The BERT-based17 classifier is pre-trained on the gold explanations and reaches an F-score of 87.59%. We fix the classifier during training of the generation model. To evaluate the impact of the evidence retrieval and claim verification on explanation generation, we compare the performance of our approach based on gold evidence and/or gold truthfulness labels with the system-based evidence and truthfulness labels. Note that we only train the model based on gold evidence and truthfulness but perform inference by taking different types of evidence or truthfulness as input. Similar as [35], we further compare our method to LEAD-3, which selects the first three sentences in evidence, and the ORACLE baseline [44], which greedily select18 multiple evidence sentences that maximize the ROUGE-2 score. Table 5 shows the results with the following observations: (1) Without generation, the explanation is directly from the concatenation of all the text evidence. The explanation may contain all the necessary information but is not interpretable to humans as the sentences are not connected coherently or logically; (2) Evidence retrieval has a more significant impact on explanation generation than claim verification. This is reasonable because the evidence carries most of the content in the explanation and truthfulness is usually implied when comparing the evidence and the input claim. (3) The explanation in our corpus is pretty abstractive, as corroborated by the low performance of ORACLE baseline, which is the upper bound of extractive summarization, and LEAD-3 baseline.

5.4 Implementation Details
We use 2 Quadro RTX 8000 to run our experiments. The retrieval models cost 15 GB and are trained for about 20 runs with a batch size of 256. The claim verification models cost 3 GB and are trained for about 50 runs with a batch size of 128. The explanation generation model costs 45 GB and is trained for about 30 runs with a batch size of 10. We use grid search to tune the hyperparameters: for evidence retrieval, the learning rate ∈ {10−5, 10−4, 10−3} and batch size ∈ {256, 480, 512}; for claim verification, the learning rate ∈ {10−1, 10−2, 10−3, 10−4} and batch size ∈ {64, 128, 256, 512, 1024, 2048}; for explanation generation, the learning rate ∈ {5×10−2, 5×10−3, 5×10−4, 5×10−5} and batch size ∈ {10, 12, 48, 192}.

6 REMAINING CHALLENGES
6.1 Claim Verification
We randomly sample 50 claims with gold evidence that are incorrectly verified from the Test set and identify the following remaining challenges for multimodal fact-checking:

Cross-modality Reasoning: Both text evidence and image evidence provide complementary information to verify the truthfulness of the claims. 30% of verification errors are due to deep cross-modality reasoning and evidence fusion. For example, for claim #2 “‘If you just count all the deaths in the red states, we are number two in the world in deaths.’” in Figure 4, since there are two different definitions for the red state, the model needs to refer to the image map to confirm the mentioned states.

Cross Document/Sentence Reasoning: 30% of verification errors are due to the reasoning across multiple pieces of textual evidence or across multiple sentences. For example, given the claim “The Biden administration’s American Jobs Plan will be ‘the biggest non-defense investment in research and development in the history of our country’,” the model needs to first know the current largest investment is $11 billion by referring to evidence “The largest increase in research and development came in 1964, and totaled $11 billion”, and then refer to another piece of evidence “experts say the plan is likely to far exceed $11 billion in spending on research and development.” to understand that the Plan will exceed $11 billion.

Deep Visual Understanding: For 6% of wrongly predicted claims, their image evidence is charts, tables, or even maps. The current visual understanding techniques, such as CLIP, cannot deeply understand the content and semantics of such images. For example, given claim #3 “San Francisco had twice as many drug overdose deaths as COVID deaths last year” in Figure 4, to determine the truthfulness of this claim, the model needs to obtain the number of drug overdose deaths from the image.

Other Complex Reasoning: Many claims also require various types of complex reasoning, such as mathematical calculation (4% of errors) and commonsense (8% of errors). For instance, the model needs to understand that “29,000 recipients” plus “12,700 recipients” is “41,700 recipients”, “from 2019 to 1998” is “22 years”, “there are fifty states in US”. In addition, the model has difficulty in dealing with claims (12% of errors) that are partially supported or refuted. For example, for the claim “Since 2010, student debt has increased by 102% and real wages have fallen by over 8%.”, it’s true that “student debt has increased by 102%” but the “real wages have fallen by over 8%” is not correct.

6.2 Explanation Generation
We also sample 50 system-generated explanations and analyze their error types as follows.

Limited Encoding and Decoding Length: Our approach is based on pre-trained language models, such as BERT and BART, which
#9: The man next to Mike Pompeo in a November 2020 photo 'is the guy the Trump administration helped get out of jail in 2018 and who is now the 'president' of Afghanistan

Figure 4: Examples of Multimodal Fact Checking. The Truthfulness column shows gold labels.
can only encode or decode a limited length of the sequence. In our dataset, some evidence and ruling statements exceed the maximal length. For those cases, we truncate the sequence and lose part of the information.

**Missing Evidence:** As we construct the evidence source collection based on the evidence links listed on Snopes and Politifact, some evidence used in the ruling statements is not included. For example, given the claim “By revoking the Keystone pipeline permit, Biden is destroying 11,000 jobs” the gold explanation contains the information “A 2014 report found that the company would need only 50 employees to maintain the Keystone XL pipeline” which is not covered in any of the background documents. In addition, our current explanation generation approach only leverages text evidence while image evidence can also provide complementary information.

**Logical Coherence:** One critical challenge for explanation generation is to determine the logical connection among the evidence sentences and organize them coherently, a common issue in long-form text generation [26, 27]. For example, given the claim, “A new, independent study found that at least 55 of our largest corporations used various loopholes to pay zero federal income tax in 2020.”, our explanation generation approach fails to correctly organize the following two evidence: “many of the relevant provisions are deliberate attempts to set incentives” and “Some critics say the financial disclosures used to compile the report are imperfect estimates”

### 7 CONCLUSION

We created MoCHEG, an end-to-end multimodal fact-checking and explanation generation benchmark dataset which consists of 15,601 claims annotated with truthfulness labels, together with 33,880 text evidence, 12,112 image evidence as well as explainable statements. We explore the state-of-the-art neural architectures to set up the baseline performance on three sub-tasks (i.e., multimodal evidence retrieval, claim verification, and explanation generation). Our experimental results show that the performance of all three sub-tasks is still far from enough. For future work, an obvious next step is to explore more advanced techniques to improve the three sub-tasks and deep visual understanding. Furthermore, open-domain fact-checking is another promising direction to detect hallucination errors in large language models like ChatGPT [46]. In the open-domain setting, evaluating the trustworthiness of evidence will play a critical role.

### 8 ETHICAL STATEMENT

For dataset release, we have obtained permission from both Snopes and Politifact to publish the data for the research purpose. Our dataset is licensed under the CC BY 4.0\(^1\), while the associated codes to MoCHEG for data crawler and baseline are licensed under Apache License 2.0\(^2\). Our dataset contains 2,916 tweets. In accordance with the Twitter developer terms\(^3\), we will only share the Twitter IDs and scripts to crawl tweets based on Twitter API. Our work can be used to predict the truthfulness of various claims in the web and stop the spread of misinformation. Our dataset does not use features or label information about sensitive personally identifiable information, like individual names. Since our dataset contains internet claims, some claims may be offensive. However, we crawl the articles from some reputational fact-checking websites, like Politifact and Snopes, to decrease the possibility of offensive content.

Given the importance of fact-checking in secular societies, we introduce the fact-checking process of Snopes and Politifact to show how our data sources reduce bias. According to Politifact\(^4\) and Snopes\(^5\), they always attempt to contact the person, website, or organization that made the statement they are fact-checking. They will have consultations with a variety of expertise. They seek direct access to government reports, academic studies, and other data. They also have one to two rounds of reviews. Finally, they will accept the error correction from the public and mark the corrected articles. According to Politifact, Politifact journalists avoid the public expression of political opinion and public involvement in the political process to set their own opinions aside as they work to uphold principles of independence and fairness. 23 of 36 journalists are women. According to Snopes, members of their editorial staff are precluded from donating to or participating in political campaigns, political party activities, or political advocacy organizations. 6 of 10 journalists are women.

### ACKNOWLEDGMENTS

This research is based upon work supported by U.S. DARPA KMASS Program # HR00121IS0034. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

---

\(^1\)https://creativecommons.org/licenses/by/4.0/
\(^2\)https://www.apache.org/licenses/LICENSE-2.0
\(^3\)https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases
\(^4\)https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology/
\(^5\)https://www.snopes.com/transparency/
