Triple-R: Automatic Reasoning for Fact Verification Using Language Models

Mohammadamin Kanaani, Sajjad Dadkhah, Ali A. Ghorbani
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
{mkanaani, sdadkhah, ghorbani}@unb.ca

Abstract

The rise of online social media platforms has made them a popular source of news. However, they are also prone to misinformation and fake news. To combat this, fact-checking is essential to verify the accuracy of claims made on these platforms. However, the existing methods in this field often lack the use of external sources and human-understandable explanations for system decisions. In this paper, we introduce a framework called Triple-R (Retriever, Ranker, Reasoner) that addresses these challenges. The framework uses the Web as an external knowledge source to retrieve relevant evidence for claims and includes a method to generate reasons based on the retrieved evidence for datasets lacking explanations. We then use this modified dataset to fine-tune a causal language model that generates natural language explanations and labels for pairs of retrieved evidence and claims. Our approach aims to improve the transparency and interpretability of fact-checking systems by providing understandable explanations for decision-making processes. We evaluated our method on a popular dataset and demonstrated its performance through an ablation study. The modified dataset is available on the Canadian Institute for Cybersecurity datasets webpage at https://www.unb.ca/cic/datasets/index.html.

Keywords: Fact Verification, Large Language Models, Reasoning, Information Retrieval, Explainable Artificial Intelligence

1. Introduction

Online social media platforms, such as Twitter, have become a primary source of news for a large segment of the population. Despite the numerous benefits they offer, these platforms are vulnerable to the spread of misinformation in the form of fake news, rumors and more. This issue has the potential to cause significant political, social, and financial harm. Though independent fact-checking teams work hard, verifying every piece of news manually is a daunting task due to resource constraints. Therefore, there is an urgent need for an automated system that can accurately and efficiently assess the truthfulness of claims made on social media.

The main objective of fact verification is to classify claims as either true or false. Various studies have attempted to address this problem by exploiting text classification methods such as machine learning and deep learning algorithms (Ahmad et al., 2020; Kaliyar et al., 2020). However, fake news poses a unique challenge in its intent to deceive, which makes it difficult to differentiate from authentic news based solely on its content. In practice, even expert fact-checkers often consult multiple sources before drawing a conclusion about a given claim. Thus, the incorporation of auxiliary information into the fact verification process has the potential to significantly enhance the accuracy of models.

Fact verification systems face the daunting challenge of integrating contextual information from multiple sources. Social media platforms take into account user profiling, source assessment, the relationship between users, propagation of claims, and other factors (Shu et al., 2017a). Knowledge graphs are a popular method to incorporate textual data. These graphs are a graphical representation of interconnected data, which illustrate concepts and relationships between various entities (Mayank et al., 2022a). However, creating the corresponding graph and extracting the relationship between entities can be challenging. Another type of contextual data is evidence retrieved from a factual information database in natural language format, which is related to the claims in question. The claim and its evidence are then processed to determine their veracity.

Fact verification systems often lack explainability, even when they have a detection component. This may lead to users hesitating to trust the system's decision without sufficient justification(Brandtzaeg et al., 2018; Vallayil et al., 2023). Incorporating explainable models would help non-expert users understand domain-specific claims and provide researchers with insight into how the system makes decisions. This would ultimately lead to the development of more robust fact verification systems. Therefore, the inclusion of explainable models is a crucial component of fact verification systems.

[0]https://www.unb.ca/cic/datasets/index.html

LREC-COLING 2024, pages 16831–16840

Fact verification systems use various techniques to improve their explainability. One such technique, suggested by Szczepański et al. (2021), is feature importance analysis. This approach aims to identify the key features that the model depends on for decision-making. However, even after identifying the important words, it is still the user's responsibility to interpret how these words contribute to the model's decision. This can make the explanation less clear to understand.

This paper introduces a framework called Triple-R, which aims to tackle the main challenges in fact verification. The framework consists of three components: **R**etriever, **R**anker, and **R**easoner, each of which plays a specific role in the fact verification process. To enhance the accuracy and efficiency of the system, external knowledge available on the web is used to retrieve the most pertinent evidence for the input claim. Additionally, language models are employed to analyze the evidence and reason about the claim before assigning it an appropriate label. Table 1 illustrates an example analyzed by the Triple-R framework. In summary, the contributions of this paper are:

- Our proposed causal language model is capable of determining the truthfulness of a claim, which enables us to understand how the model makes decisions. This leads to greater transparency and interpretability in the process of fact verification.
- We use a larger language model to supervise a smaller one, which improves the accuracy and effectiveness of our framework.
- We present a hybrid zero-shot ranker that retrieves supporting information to justify the claim. The gathered evidence serves as a form of explanation that reinforces the generated reasoning.

Claim: Over 10 years, (extending tax cuts for the wealthy) would add less to the deficit than Obama added with the stimulus in one year.
Evidence: 1. "Over 10 years, (extending tax cuts for (...) 2. "The president says we can't afford the (...) 3. Ultimately, we feel that Will had a point (...) 4. Just as the upper-income tax cuts are (...)
Reason: The statement is partially true but lacks important context. While it is true that the tax cuts for the wealthy would add less to the deficit over a 10-year period than the stimulus did in one year, it is misleading to suggest that the stimulus is vastly more expensive than the tax cuts. The stimulus package included various funding streams and tax breaks, and the Congressional Budget Office (CBO) reported that it would increase the deficit in each year until 2015 before reducing it in subsequent years. Therefore, it is important to consider the overall impact of the stimulus and the tax cuts over a longer period rather than comparing their individual yearly costs.
Label: barely-true

Table 1: Table showcasing an example claim analyzed through the Triple-R framework, utilizing web-based information retrieval to find pertinent evidence. The framework subsequently formulates a reasoned judgment and assigns a label indicating the veracity of the claim.

2. Related Works

In this section, we categorize fact verification research into three camps: content-based detection, context-enriched methods, and explainable fact verification.

2.1. Content-based Detection

When it comes to verifying the accuracy of information, there are methods that rely on analyzing the content itself to determine whether a claim is true or false. This approach involves using various techniques to extract and examine the features of the text. Researchers have explored different ways of doing this to improve the accuracy of fact-checking. Ahmad et al. (2020) utilized the Linguistic Inquiry and Word Count (LIWC) tool to extract textual features from articles, which were then used to train machine learning models such as Logistic Regression, Linear SVM, and Random Forest. Similarly, in the study by Pérez-Rosas et al. (2017), classification models based on linear SVM were developed using a combination of lexical, syntactic, and semantic information and features representing text readability properties.

The FNDNet method (Kaliyar et al., 2020) utilizes a convolutional neural network (CNN) architecture for fake news detection. The method leverages pre-trained GloVe word embedding vectors as input, which are then fed into three parallel convolutional layers with different kernel sizes, followed by max-pooling layers. The outputs of these layers are concatenated and passed to the subsequent convolutional layers. Finally, two dense layers are employed, with the second layer responsible for predicting the final output.

Nasir et al. (2021) proposed a hybrid framework that combines Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) architectures. The framework employs pre-trained GloVe word embeddings to convert words into vectors. A Conv1D CNN layer is used to process the input vectors and extract local features, followed by a layer of Long Short-Term Memory (LSTM) units that learns

the long-term dependencies of the local features.
The FakeBERT model (Kaliyar et al., 2021) leverages the recent transformers-based models. This model utilizes the BERT model to generate word embeddings, which are then passed through three parallel convolutional layers. The resulting outputs are concatenated and fed to two additional convolutional layers. Each convolutional layer is followed by a max-pooling layer. The final output of the convolutional layers serves as input to two dense layers, with the last layer utilizing softmax activation.

2.2. Context-Enriched Methods

Verifying the accuracy of factual information in text can be difficult and different from other text classification tasks. Some studies have included contextual information to enhance the effectiveness of fact verification systems. There are mainly two types of external context that have been considered: social media information and textual evidence.

2.2.1. Social Media Information

Social media is a complex world where so many different factors can influence the accuracy of the information presented. These factors include user information, relationships between users, and patterns of news propagation. To assess the truthfulness of information shared on social media, researchers have explored different types of auxiliary information.
The TriFN method proposed by Shu et al. (2017b) aims to investigate the interdependencies among publisher bias, news stance, and relevant user engagements. The proposed approach incorporates not only document and user features but also user-user relationships, the social engagements of users that reveal their preferred news sources, and the relationship between publishers and their news. Furthermore, the partisan label of each publisher is also taken into account to improve the accuracy of the proposed method.
In their work, Shu et al. (2019) proposed an approach to incorporate two sets of user profiles, namely implicit and explicit. The implicit features are inferred from user meta information or online behaviors, while the explicit features are directly obtained from user meta-data. The authors proposed to compute the user profile feature (UPF) of a news as the average feature scores of all the users that share the news. Moreover, they compared the performance of the UPF with several state-of-the-art feature representations for detecting fake news.
Dou et al. (2021) proposed a novel framework for fake news detection named Use Preference-aware Fake News Detection (UPFD), which comprises three key components. Firstly, user preferences are extracted by encoding historical posts through text representation learning methods, along with the news textual data using the same approach. Secondly, the news propagation graph is constructed and encoded using Graph Neural Networks (GNN) as the graph encoder, where the node features are composed of the news textual embedding and user preference embedding. Finally, the news textual embedding and user engagement embedding are concatenated and fed into a two-layer Multi-Layer Perception (MLP).

2.2.2. Textual Evidence

DEAP-FAKED is a novel approach proposed by Mayank et al. (2022b) to detect fake news using a knowledge graph. The proposed approach utilizes a stack of bidirectional Long Short-Term Memory (biLSTM) to encode the news title. Moreover, the authors extract the entities from the news text through Named Entity Recognition (NER) and then map them to the corresponding entities in a Knowledge Graph (KG) using Named Entity Disambiguation. In order to embed the KG entities into vectors, ComplEx embedding is used. Finally, a concatenation of the text embedding and the KG embedding is used to detect fake news.
Hu et al. (2021) propose CompareNet, a method for fake news detection that constructs a directed heterogeneous document graph for each news document, containing sentences, topics, and entities as nodes. They use a heterogeneous graph attention network to learn topic-enriched news representations and contextual entity representations, which are then compared to corresponding KB-based entity representations with a carefully designed entity comparison network to capture semantic consistency between the news content and external knowledge base. Finally, the topic-enriched news representations and entity comparison features are combined for fake news classification.

2.3. Explainable Fact Verification

DeClare is a system introduced by Popat et al. (2018) that incorporates textual evidence to detect fake news. The system searches web articles related to an input claim and considers snippets from these articles as evidence. The claim is then transformed into a vector through word embeddings, and the web articles are encoded with a biLSTM. An attention mechanism is applied to highlight parts of the article that are relevant to the claim, and the attention weight is considered as an explanation. The claim and article sources are also utilized to arrive at a final decision.
Chen et al. (2021) proposed a system called HHGN which is comprised of five major components for claim verification: evidence retrieval, graph construction, node features initialization, reasoning-based node updating, and prediction layer. The system first retrieves evidence sentences related to the claim and then constructs a heterogeneous

graph to model the relationship between different semantic units extracted from the pieces of evidence. The system employs BERT to obtain the initial representations of semantic nodes and uses a reasoning-based node updating component to propagate the node features. Finally, the system applies a prediction layer to capture the reasoning features and produce the result.

Vedula and Parthasarathy (2021) developed a framework named FACE-KEG that investigates the veracity of a textual claim or fact and generates a human-understandable explanation about its truthfulness. To achieve this, they construct a knowledge graph associated with the input claim, extract relevant textual context, and use a bidirectional RNN and a graph transformer network to encode the associated textual context and knowledge graph, respectively. They jointly train a classifier and a decoder to predict the true value of the input fact and generate a natural language explanation about its veracity.

3. Proposed Method

When it comes to verifying the accuracy of a claim, we strive to replicate the process that a human would typically undertake. This involves conducting online research to gather relevant information and evidence, which is then analyzed to determine the validity of the claim. To achieve this process, we have developed a three-stage fact verification model, which is illustrated in Figure 1. Our model is designed to mimic the human approach of gathering information, synthesizing evidence, and using logical reasoning to draw a conclusion about the veracity of a given claim.

The first stage of our model is called the "Retriever". It involves sourcing information related to the claim from the internet, which is used as an external knowledge base. The second stage is called the "Ranker". In this stage, a score is assigned to each paragraph of the retrieved documents based on their similarity to the claim. The top-scoring paragraphs are then selected as evidence. Finally, the third stage is called the "Reasoner". This stage employs a generative language model to label the claim and provide a rationale for its veracity based on the collected evidence.

3.1. Retriever

Fact verification faces a major challenge in dealing with data shifts. As news is dynamic, new events may arise that are not present in the model's training set. This can cause the model to struggle to predict the correct label. To address this, we make use of the Internet as an external knowledge source to provide additional context. The Internet houses constantly evolving information, which ensures that we have access to the latest data. To retrieve relevant information, we use search engines that perform a thorough search to identify the most relevant websites based on the input query. This helps us retrieve documents related to the claim to some extent. To find related articles, we search for the claim verbatim using the Bing Search API. We then scrape the content of the top URLs and divide them into paragraphs for further analysis.

3.2. Ranker

To ensure that only the most relevant information is used and to minimize noise, it's important to extract and score paragraphs that are related to the claim from the retrieved documents. To accomplish this, we developed a hybrid model that combines a term-based model and a neural network model. By scoring each paragraph based on its relevance to the claim, we can accurately extract the most related paragraphs from the retrieved documents while minimizing the impact of irrelevant content. Our approach is inspired by the work of Ma et al. (2020).

3.2.1. Neural Network Model

Our proposed method for verifying facts uses a neural network model to encode the input text and facilitate comparison. To do this, we use a language model called DistilBERT (Sanh et al., 2019), which is based on the transformer architecture (Vaswani et al., 2017) and is trained to generate dense vector representations of text. Using DistilBERT, we encode both the claim and paragraphs and consider the corresponding vector to the `[CLS]` token as the text representation. Then, we calculate the similarity score between the vector of the claim and each paragraph by taking the inner product.

In our work, we use the DistilBERT model as a zero-shot model, which means that it is not fine-tuned on any specific task or dataset. This allows the model to leverage its pre-trained knowledge to generate embeddings.

3.2.2. Statistical Model

While DistilBERT excels in contextual understanding, we enhance its performance by identifying and retrieving paragraphs that contain the words present in the claim as closely as possible. To achieve this, we incorporate the Okapi BM25 algorithm, a powerful zero-shot model that has demonstrated superior performance over supervised neural models in various applications (Robertson et al., 1995).

The Okapi BM25 algorithm is a widely used information retrieval method that assesses the relevance of a document with respect to a given query. This algorithm takes into account the term frequency and inverse document frequency of the query terms, as well as the length of the document and the average length of documents in the collection. The formula

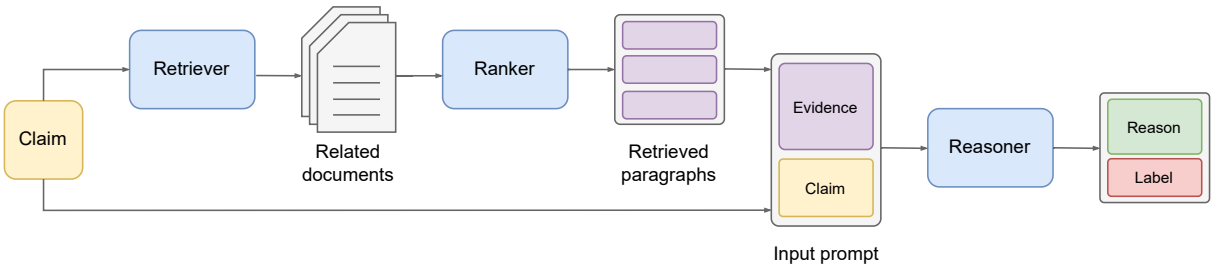


Figure 1: The three-stage for fact verification

for calculating the relevance score of a document is as follows:

$$score(D, Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}$$

where D is the document, Q is the query, n is the number of query terms, $f(q_i, D)$ is the frequency of the i^{th} query term in the document, $|D|$ is the length of the document, $avgdl$ is the average length of documents in the collection, k_1 and b are hyper-parameters, and $IDF(q_i)$ is the inverse document frequency of the i^{th} query term, defined as:

$$IDF(q_i) = \log \frac{N}{n(q_i)}$$

where N is the total number of documents in the collection and $n(q_i)$ is the number of documents that contain the i^{th} query term. The higher the Okapi BM25 score of a document, the more relevant it is to the query.

To reach the final score, we combine these two scores:

$$sim(c^{hyb}, p^{hyb}) = \lambda < c^{bm25}, p^{bm25} > + < c^{nn}, p^{nn} >$$

where λ is an interpolation hyper-parameter that trades off the relative weight of BM25 versus the neural model.

3.3. Reasoner

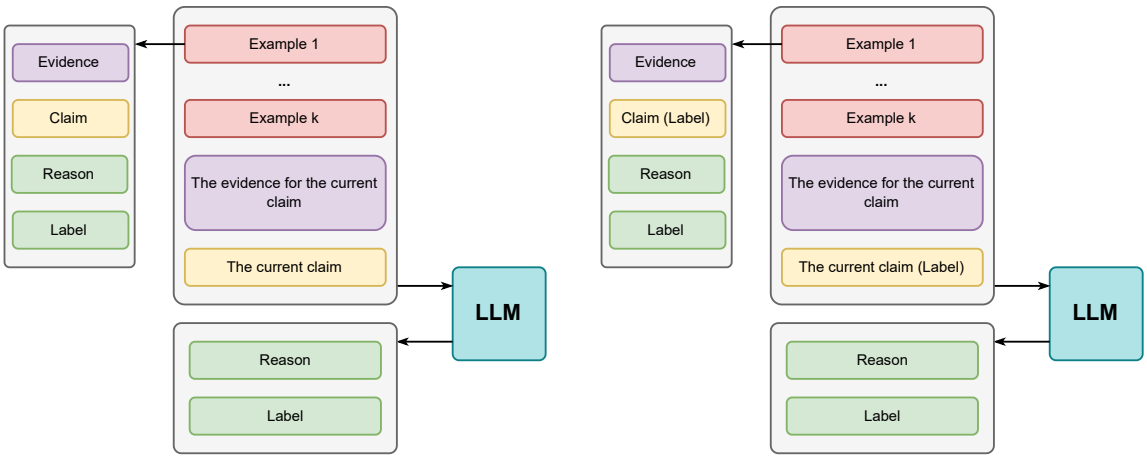
The last component in the Triple-R system is the reasoner, accomplished by developing a causal language model. We utilize the text generation capability of a language model to create reasons and labels for given claims and evidence pairs. Consider a dataset $D = (c_1, y_1), \ldots, (c_N, y_N)$ with N samples, where each c_i is a claim, and y_i is its corresponding label. For every claim c_i, the retriever identifies a set of Internet articles denoted as $A = \{a_1^i, \ldots, a_m^i\}$, where m represents the number of top articles. The ranker then breaks down these articles into paragraphs and selects the top-k most relevant paragraphs for the claim. This forms the evidence set $e_i = \{e_i^1, \ldots, e_i^k\}$. Ultimately, when presented with a set of evidence and a claim, the role of the reasoner is to generate both the reason and the label.

$$\text{reasoner}(\{e_i^1, \ldots, e_i^k\}, c_i) = (r_i, \hat{y}_i)$$

To fine-tune a causal language model for generating reasons and labels based on input evidence-claim pairs, a suitable dataset with annotated reasons is essential for the fine-tuning process. However, many existing datasets lack these annotations, which are vital for establishing the connection between claims and their supporting evidence. Without this linkage, the model's learning process could be hindered. In the following subsection on reason generation, we explore how to create a synthetic dataset where each claim is provided with a reason derived from the evidence. The subsequent part covers the fine-tuning process, detailing how the causal language model is refined using this synthetic dataset to proficiently generate both reasons and labels for the input claims.

3.3.1. Reason Generation

Building upon the concepts of in-context learning (ICL) (Brown et al., 2020) and chain-of-thought (CoT) (Wei et al., 2022), we utilized a large language model to construct a synthetic dataset. Let M represent the large language model, and $D = \{(e_1, c_1, y_1), \ldots, (e_N, c_N, y_N)\}$ denote the dataset. Following the CoT approach, we randomly select k examples from this dataset, where k is significantly smaller than N, and manually compose natural language reasons for the claims based on their retrieved evidence. This process yields a natural language prompt $P = \{I, f(e_1, c_1, r_1, y_1), \ldots, f(e_k, c_k, r_k, y_k)\}$, where $f(e_i, c_i, r_i, y_i)$ is a function mapping the i-th set of evidence, claim, reason, and label to a natural language prompt, the Figure 2a shows the input prompt. Here, r_i is the written reason for claim c_i based on the retrieved evidence e_i, and I provides an instruction describing the fact verification task to the large language model. Similar to the few-shot prompting in the ICL method, we append each example in D to the prompt P. Given this input prompt, the large



(a) The input prompt (b) The input prompt with hint

Figure 2: (a) The structure of the prompt which forces the model to generate a reason and a label according to the examples in the prompt. Each example contains evidence, claim, reason, and label. (b) For the prompt in the rationalization method, we add the label as a hint at the end of claims for current input and examples.

language model's output for the pair (e_j, c_j) from the test set can be formulated as follows:

$$M(I, \underbrace{d_1, \ldots, d_k}_{\text{demonstrations}}, f(\underbrace{e_j, c_j}_{\text{test}}, \underbrace{\_\_\_\_\_}_{\text{reason}}, \underbrace{\_\_\_\_\_}_{\text{label}})) \rightarrow (r_j, \hat{y}_j)$$

Following the provided instruction, the model comprehends the task and then employs the pattern of reasons outlined in the prompt P to generate both a reason r_j and a label $\hat{y}_j$. The demonstrations guide the model to reason based on the information provided in the evidence section. If the model accurately predicts the label, we naturally assume that it effectively utilizes the evidence to produce a logical reason leading to the accurate answer. Therefore, we focus solely on reasons that correspond to correct answers, $(\hat{y}_j = y_j)$.

Generating reasons for claims using this approach has a limitation: progress stalls when the model struggles to create a reason for a claim. This is due to the lack of training signals from unsuccessful examples. Taking inspiration from the STaR method (Zelikman et al., 2022), we adopt a technique known as rationalization. In this process, we provide the model with the answer as a hint and ask it to generate a reason in a manner similar to the earlier reason generation step. With the label at hand, the model can work backward, making it easier to formulate reasons that lead to the correct answer. For example, as shown in Figure 2b, we embed the hint in the prompt within parentheses to guide the reason generation. Rationalization is applied to claims where the model fails to accurately predict the label. When we include a reason generated through rationalization in the synthetic dataset, we omit the hint from its corresponding prompt, giving the impression that the model devised the reason autonomously.

3.3.2. Fine-tuning

After creating the new dataset containing the generated reason, every input prompt is the concatenation of evidence, a claim, a generated reason, and at the end, a label, i.e., $X_i = \{e_i, c_i, r_i, y_i\}$. We fine-tune the model on this dataset with loss only being calculated on the reason and the label tokens. In other words, given the evidence, the claim, the reason, and the label, the model is trained to predict only the reason and the label. We calculate the loss for a sample using the cross-entropy loss function, which is the negative log probability the model assigns to the next word in the training sequence:

$$\mathcal{L}_{CE}(\hat{y}, y) = -\sum_{i=r}^{N} \log(\hat{y}_i)$$

where r is the index of the reason, and N is the number of the tokens in the input prompt. The model estimates this probability by using a softmax over the set of possible outputs:

$$p(x_t|x_{t-1}, \ldots, x_1) = softmax(Wh_t + b)$$

where $h_t \in \mathbb{R}^d$ is the output vector of the neural network at time t, $W \in \mathbb{R}^{|V| \times d}$ is a parameter matrix that is learned during training.

After training, our fine-tuned model is employed to generate reasons and labels for test samples. This process involves executing the retriever and ranker steps on the test samples to gather supporting evidence. By combining the retrieved evidence

with the claim, we create a natural language input prompt $X_j = f(e_j, c_j)$, where the function f maps them into the appropriate format. Using this prompt, our fine-tuned model generates a reason and corresponding label:

$$(r_j, \hat{y}_j) = \text{reasoner}(f(e_j, c_j)) \quad (1)$$

To extract the label, we incorporate a specific sentence in the reason generation step that encloses the label. After the model's output is generated, we search for this sentence to extract the label, which is then used for evaluation purposes.

4. Experimental Results

4.1. Dataset

In this study, we evaluate our proposed method using the LIAR dataset (Wang, 2017), which consists of claims classified into six categories: *true*, *mostly true*, *half true*, *barely true*, *false*, and *pants-on-fire*. As the claims in the dataset are sourced from the PolitiFact[1] website, we restrict the search engine to retrieve documents only from the PolitiFact domain. We then select the top three results and score the paragraphs using the ranker component. To balance the neural network and statistical score, we set the $\lambda = 0.9$, and then select the top four paragraphs.

4.2. Experimental Setup

For reason generation, we construct an input prompt comprising six examples by randomly selecting one from each label and manually providing a reason based on the available evidence. We harness the reasoning capabilities of the GPT-3 large language model (Brown et al., 2020) to automate reason generation for the entire training dataset. Specifically, we utilize the GPT-3.5-Turbo model accessible via the OpenAI API[2], which is optimized for dialog and equipped with three designated roles: `system`, `user`, and `assistant`. The `system` role is employed to specify the task instruction. For each demonstration, we assign the evidence and claim to the `user` role and the reason and label to the `assistant` role. Finally, for each pair of evidence and claim in the dataset, we place them in the `user` role, and the model, acting as the `assistant`, generates a reason and label based on the provided demonstrations. It's noteworthy that we set the temperature to 0.3 to control the level of randomness in the generated text.

In our experimental setup, we employ a transformer decoder derived from the Llama 2 family (Touvron et al., 2023). Specifically, we utilize Llama-2 7b-chat, a variant of Llama 2 (7b) fine-tuned for dialogue-based applications. We structure the input using the Llama 2 template. The instruction employed for reason generation is integrated within the «`SYS`» tag. Furthermore, to ensure a clear distinction between input and output, we employ the `[INST]` delimiter to separate the evidence-claim pairs from the generated reason-label pairs.

We perform fine-tuning on the Llama-2 (7b) model using a V100 GPU equipped with 16GB of VRAM. However, the GPU's VRAM is barely sufficient to accommodate the model's extensive weights (7b $\times$ 2 bytes = 14 GB in FP16). Factoring in additional requirements for optimizer states, gradients, and forward activations, we face limitations. To address this, we implement a parameter-efficient fine-tuning technique known as QLoRA (Dettmers et al., 2024). QLoRA utilizes 4-bit quantization to backpropagate gradients through a frozen pretrained language model into Low Rank Adapters, significantly reducing VRAM usage. We set the QLoRA rank to 8 and the scaling parameter to 16. The fine-tuning process involves 5 epochs, utilizing the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of $1e-4$. A warm-up linear schedule (Goyal et al., 2017) with 100 warm-up steps is incorporated into the learning rate schedule, with a batch size of 4 for optimization.

During the fine-tuning process, a tag denoted as "Label: " is added to the end of each sample, representing the label for the claim. After the models are fine-tuned, a regular expression (regex) search is conducted to identify this sentence at the end of the generated text. If the correct format is found, the extracted label is considered the predicted label for the claim.

4.3. Result and Discussion

In our experiment, our framework performs two tasks: identifying relevant evidence for each claim and generating a reason for the claim based on the identified evidence. We have compared the performance of our model with other models that classify claims with and without evidence.

In our ablation study, we used the BERT base model in two settings. In the first setting, we trained a model to classify claims by using the vector corresponding to the `[CLS]` token as the representation for the claim. In the second setting, we trained a model on pairs of claims and evidence, separated by the `[SEP]` token, and considered the vector corresponding to the `[CLS]` token as the representation for the claim-evidence pair. We also performed few-shot classification using the GPT-3.5.Turbo with the same prompt used for the reason generation part. The results of the ablation study are presented in Table 2a.

The results of fact verification for the Llama 2 (7b) model's performance on a subset of the LIAR

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

[2] https://openai.com/product

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
BERT/Without Evidence	29.3	29.83	27.7	27.92
GPT-3.5-Turbo	37.1	36.16	34.35	30.3
BERT/With Evidence	40.29	43.19	39.74	40.83
Llama 2 (7b)	42.72 $\pm$ 0.19	50.1 $\pm$ 2.7	41.1 $\pm$ 0.52	39.16$\pm$ 0.34

(a) Ablation study result

	Accuracy (%)
Wang (2017)	27.4
Koloski et al. (2022)	26.75
Long et al. (2017)	41.5
Triple-R	42.72

(b) Previous works vs. Triple-R framework

Table 2: Table (a) presents a comparative analysis of model performance in an ablation study. Table (b) provides a summary of accuracy scores achieved in previous studies, contrasting them with the proposed method.

dataset are summarized in Table 2. The evaluations show that GPT-3.5-Turbo was only able to generate a reason for around 36% of the dataset during reason generation, leaving the rest without correctly predicting the label. However, when combined with the rationalization method, the model managed to generate a reason for nearly 97.8% of the dataset.

Our analysis includes a comparison between the Llama 2 (7b) model's performance, as detailed in Table 2a, and other models used in the ablation study. We also contrasted our results with those from previous studies, which can be found in Table 2b. It is particularly interesting that, despite having significantly fewer parameters, the fine-tuned Llama 2 (7b) model outperforms the GPT-3.5-Turbo model. This underscores the potential of fine-tuned smaller models over larger and generalized models.

The results also showed that augmenting context, such as providing evidence, substantially enhanced the fact verification system's effectiveness, as demonstrated by the results from the BERT models.

In the test set, there were 1283 samples, out of which 19 samples did not return any search engine results. Despite this, Llama 2 demonstrated an average accuracy of nearly 34.7% in all experiments when evaluating its performance on these samples. This indicates that the model can achieve accuracy levels comparable to the overall dataset, even when trained on a dataset where all samples include evidence. In the five evaluations, the model mainly predicted labels as next-sentence predictions, but only once it generated a label for a claim not covered by the predefined labels.

5. Conclusion and Future Works

Fact verification has become increasingly important in recent years. However, to ensure trust in these systems, it's crucial for users to understand the decision-making processes. Access to external knowledge sources is also necessary for these systems to accurately predict the truthfulness of claims, particularly in fast-paced events. To address these challenges, we've proposed a framework called Triple-R, which consists of three essential components: Retriever, Ranker, and Reasoner. The Retriever component uses the Web to fetch relevant documents, while the Ranker identifies the most pertinent paragraphs as evidence for the claim. The Reasoner then utilizes this evidence to produce explanations that are easy for humans to understand and predict the label for the claim. Additionally, we've explored using a pre-trained large language model's reasoning ability to generate explanations for datasets without any form of explanation. Our extensive experiments on a real-world dataset demonstrate the effectiveness of our proposed method compared to current state-of-the-art baselines.

Our future research will delve into innovative methods that surpass simple verbatim searches of claims on the web and, instead, concentrate on generating queries that are related to the information mentioned in the claim. This approach could potentially retrieve more targeted and relevant documents for fact verification. Additionally, since language models can generate multiple answers, it is crucial to develop techniques to evaluate and select the most appropriate answer that incorporates more information from the evidence section and provides a better explanation of the decision-making process. Furthermore, we will focus on developing methods to verify the reliability and credibility of the

sources retrieved from the web for fact-verification purposes. Such efforts would significantly improve the accuracy and interpretability of fact-checking systems.

6. Bibliographical References

Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. 2020. Fake news detection using machine learning ensemble methods. *Complexity*, 2020:1–11.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. *arXiv preprint arXiv:2004.05773*.

P. B. Brandtzaeg, Asbjørn Følstad, and María Ángeles Chaparro Domínguez. 2018. How journalists and social media users perceive online fact-checking and verification services. *Journalism Practice*, 12:1109 – 1129.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chonghao Chen, Fei Cai, Xuejun Hu, Wanyu Chen, and Honghui Chen. 2021. Hhgn: A hierarchical reasoning-based heterogeneous graph neural network for fact verification. *Information Processing & Management*, 58(5):102659.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yingtong Dou, Kai Shu, Congying Xia, Philip S Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2051–2055.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 754–763.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788.

Rohit Kumar Kaliyar, Anurag Goswami, Pratik Narang, and Soumendu Sinha. 2020. Fndnet–a deep convolutional neural network for fake news detection. *Cognitive Systems Research*, 61:32–44.

Boshko Koloski, Timen Stepišnik Perdih, Marko Robnik-Šikonja, Senja Pollak, and Blaž Škrlj. 2022. Knowledge graph informed fake news classification via heterogeneous representation ensembles. *Neurocomputing*, 496:208–226.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake news detection through multi-perspective speaker profiles. In *Proceedings of the eighth international joint conference on natural language processing (volume 2: Short papers)*, pages 252–256.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. *arXiv preprint arXiv:2004.14503*.

Mohit Mayank, Shakshi Sharma, and Rajesh Sharma. 2022a. Deap-faked: Knowledge graph based approach for fake news detection. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 47–51. IEEE.

Mohit Mayank, Shakshi Sharma, and Rajesh Sharma. 2022b. Deap-faked: Knowledge graph based approach for fake news detection. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 47–51. IEEE.

Jamal Abdul Nasir, Osama Subhani Khan, and Iraklis Varlamis. 2021. Fake news detection: A hybrid

cnn-rnn based deep learning approach. *International Journal of Information Management Data Insights*, 1(1):100007.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. *arXiv preprint arXiv:1809.06416*.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Kai Shu, Suhang Wang, and Huan Liu. 2017a. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 8.

Kai Shu, Suhang Wang, and Huan Liu. 2017b. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 8.

Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. 2019. The role of user profiles for fake news detection. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 436–439.

Mateusz Szczepański, Marek Pawlicki, Rafał Kozik, and Michał Choraś. 2021. New explainability method for bert-based model in fake news detection. *Scientific reports*, 11(1):23705.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Manju Vallayil, P. Nand, Wei Qi Yan, and Héctor Allende-Cid. 2023. Explainability of automated fact verification systems: A comprehensive review. *Applied Sciences*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Nikhita Vedula and Srinivasan Parthasarathy. 2021. Face-keg: Fact checking explained using knowledge graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 526–534.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

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.

Eric Zelikman, Jesse Mu, Noah D Goodman, and Yuhuai Tony Wu. 2022. Star: Self-taught reasoner bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems (NeurIPS)*.