Robust Claim Verification Through Fact Detection

Anonymous ACL submission

Abstract

Claim verification can be a difficult task, even for humans. In this paper, we propose a method to improve automated claim verification through short fact extraction from evidence to enhance reasoning abilities. We propose a framework (FactGen) that uses Large Language Models (LLMs) to generate short factual statements from evidence and then label these facts based on their semantic relevance to the claim and evidence. We then add a relevant fact-detection task (FactDetect) to the claim verification task as a multi-tasking approach to improve performance and explainability.

Our method improves the supervised claim verification model by 15% on the F1 score when evaluated on SciFact (Wadden et al., 2020) and demonstrates competitive results on other challenging scientific claim verification datasets. We also demonstrate that FactDetect can be adjusted to the LLMs as a prompting strategy for verdict prediction. We show that incorporating FactDetect in relatively smaller LLMs such as Llama2-13B and Vicuna-13B can improve the verification performance significantly on the SciFact dataset and higher quality FactGen generated sentences outperform state-of-the-art models in all test sets.

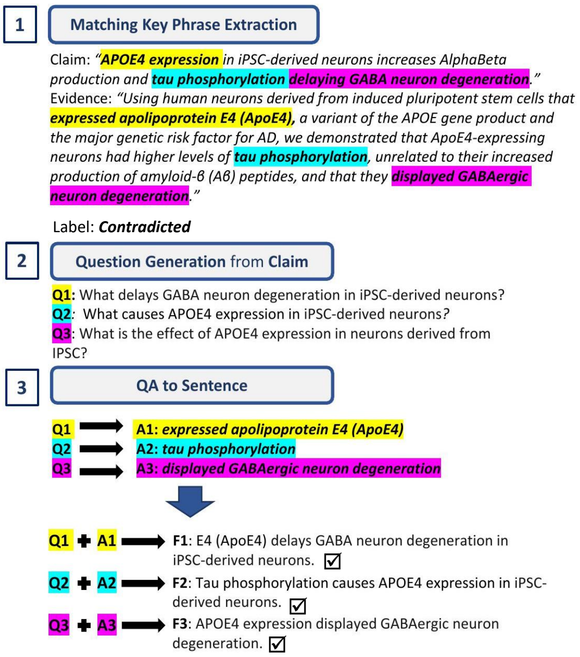
1
Matching Key Phrase Extraction
Claim: "APOE4 expression in iPSC-derived neurons increases AlphaBeta production and tau phosphorylation delaying GABA neuron degeneration."
Evidence: "Using human neurons derived from induced pluripotent stem cells that expressed apolipoprotein E4 (ApoE4), a variant of the APOE gene product and the major genetic risk factor for AD, we demonstrated that ApoE4-expressing neurons had higher levels of tau phosphorylation, unrelated to their increased production of amyloid-β (Aβ) peptides, and that they displayed GABAergic neuron degeneration."
Label: Contradicted
2
Question Generation from Claim
Q1: What delays GABA neuron degeneration in iPSC-derived neurons?
Q2: What causes APOE4 expression in iPSC-derived neurons?
Q3: What is the effect of APOE4 expression in neurons derived from IPSC?
3
QA to Sentence
Q1
Q2
Q3
A1: expressed apolipoprotein E4 (ApoE4)
A2: tau phosphorylation
A3: displayed GABAergic neuron degeneration
Q1 ✚ A1
F1: E4 (ApoE4) delays GABA neuron degeneration in iPSC-derived neurons. ☑
Q2 ✚ A2
F2: Tau phosphorylation causes APOE4 expression in iPSC-derived neurons. ☑
Q3 ✚ A3
F3: APOE4 expression displayed GABAergic neuron degeneration. ☑

Figure 1: Three-step process of short fact generation from evidence. 1) First we use LLM to generate matching phrases between claim and evidence.) Using the extracted phrases from **claim** we design a question generation to generate questions from the claim and the given phrase. 3) The generated matching phrase from **evidence** is concatenated with the question generated from **claim** for short fact generation. Check marks suggest the importance of generated sentences.

1 Introduction

Due to the proliferation of disinformation in many online platforms such as social media, automated claim verification has become an important task in natural language processing (NLP). "Claim verification" refers to predicting the verdict for a claim (supported, contradicted) given the evidence that has been extracted from a corpus of documents (Thorne et al., 2018; Wadden et al., 2022a; Guo et al., 2022).

Claim verification can be challenging for several reasons. First, the available human-annotated data is limited, resulting in limited performance by current trained models. The task is even harder for scientific claim verification where the claim and the corresponding evidence belong to specific scientific domains: verification of scientific claims generally requires specialized knowledge of the scientific background, numerical reasoning, and statistics (Wadden et al., 2020). A key challenge in developing automated argumentation systems lies in accurately representing the subtleties of argumentation. This includes the capacity to change a verdict from 'supported' to 'refuted' when a new claim in the test set potentially negates the evidence, in contrast to approving it in the training set.

Human-based reasoning for this task requires making a meaningful connection between the claim and evidence by decomposing the claim and relevant evidence into smaller and potentially simpler pieces and performing reasoning (Pan et al., 2023). A few studies proposed approaches for reasoning (Pan et al., 2023; Liangming Pan, 2021; Dai et al., 2022; Lee et al., 2021). Question-answering (i.e., asking questions from claim or evidence, retrieving the answer from each component, and utilizing the answer for the downstream task) is one of the approaches used for improving reasoning and explanation in claim verification tasks (Liangming Pan, 2021; Dai et al., 2022). Intuitively a question asked from a supported or contradicted claim should be *answerable* by the corresponding evidence. The answer provided by evidence can provide important factual information for veracity prediction.

Motivated by these reasoning approaches, we introduce FactGen. This short sentence generation framework enhances the state-of-the-art trained models – as well as LLMs – by simplifying the connection between claim and evidence pairs through identifying and distilling crucial facts from evidence and then transforming these facts into simpler and concise sentences. We hypothesize that these concise sentences will enhance various reasoning abilities, including scientific understanding, by simplifying the connection between a claim and its complex scientific evidence. FactGen comprises: a) short fact generation through a four-step process of matching key-phrase extraction, question generation, evidence-based question answering, and QA-to-sentence generation; b) weakly labeling short facts based on their importance given the claim; and, c) utilizing these facts in a multi-task learning-based training of a claim verification model or as an extra step to improve the performance of LLMs for zero-shot claim-verification. An overview of the fact-generation process with an example is given in Figure 1.

We evaluate FactDetect in two variations of multi-task-based finetuning of claim verification models and zero-shot claim verification through LLMs on four scientific claim-verification datasets of SciFact (Wadden et al., 2020), CovidFact (Saakyan et al., 2021), HealthVer (Sarrouti et al., 2021) and Scifact-Open (Wadden et al., 2022a). The code and data will be available in github[1].

The contributions of this study are:

1. We introduce FactDetect, a simple yet effective approach for condensing evidence sentences into shorter sentences derived from relevance to the claim.
2. Our extensive experiments show that FactDetect can be easily adapted to claim-verification models to improve their reasoning ability.
3. Augmenting FactDetect generated short facts for a multi-task prompting approach is useful in smaller LLMs whereas it shows less effective in larger LLMs.

2 Background

Automated claim verification means determining the veracity of a claim, typically by retrieving likely relevant documents and searching for evidence within them. The key objective is to ascertain if the evidence either *SUPPORTS* or *CONTRADICTS* the claim in question. Various datasets have been proposed to facilitate research in this area in different domains: e.g., FEVER (Thorne et al., 2018) is a Wikipedia-based claim verification dataset. Claim verification in the scientific domain has also been proposed in recent years to facilitate research in this complex domain (Wadden et al., 2022a, 2020; Saakyan et al., 2021; Sarrouti et al., 2021; Kotonya and Toni, 2020; Diggelmann et al., 2020). These datasets, despite their value, often have limited training data due to the high cost of creation, impacting the reasoning capabilities and robustness of claim verification methods.

In addressing these challenges, the literature shows significant advances in models for verifying scientific claims through reasoning. One notable strategy is the *generation* of explanations. Prior studies have explored using attention mechanisms to identify key evidence segments (Popat et al., 2017; Cui et al., 2019; Yang et al., 2019; Jolly et al., 2022). Recently, the integration of LLMs in explanation generation has been investigated. For example, ProofVer (Krishna et al., 2022) generates proofs for the claim based on evidence using logic-based inference. ProgramFC (Pan et al., 2023) uses LLMs to generate reasoning programs that can be used to guide fact-checking, and FOLK (Wang and Shu, 2023) leverages the in-context learning ability of LLMs to generate First Order Logic-Guided reasoning over a set of knowledge-grounded question-and-answer pairs to make veracity predictions without using annotated evidence.

Our work diverges from these methodologies as we propose an add-on task to enhance the robustness and reasoning ability of existing models.

[1] [Redacted link for anonymity.]

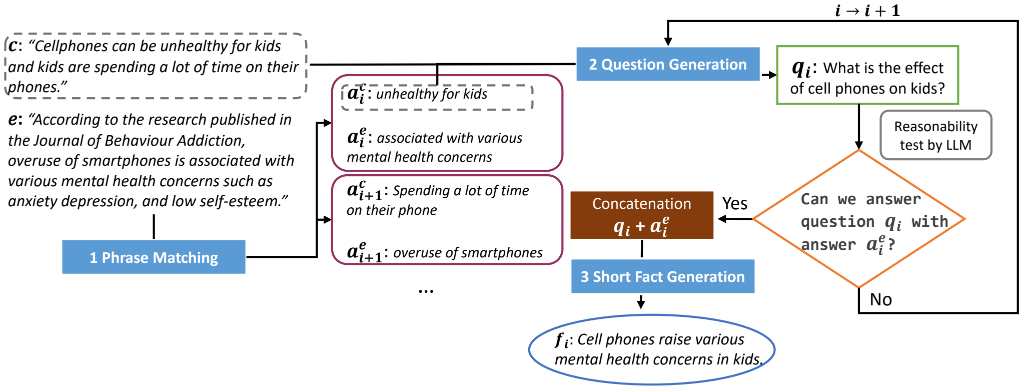
$i \to i+1$
c: "Cellphones can be unhealthy for kids and kids are spending a lot of time on their phones."
e: "According to the research published in the Journal of Behaviour Addiction, overuse of smartphones is associated with various mental health concerns such as anxiety depression, and low self-esteem."
1 Phrase Matching
a_i^c: unhealthy for kids
a_i^e: associated with various mental health concerns
a_{i+1}^c: Spending a lot of time on their phone
a_{i+1}^e: overuse of smartphones
...
2 Question Generation
q_i: What is the effect of cell phones on kids?
Reasonability test by LLM
Can we answer question q_i with answer a_i^e?
Yes
No
Concatenation
$q_i + a_i^e$
3 Short Fact Generation
f_i: Cell phones raise various mental health concerns in kids.

Figure 2: Overview of the framework.

This is achieved through a novel data augmentation strategy, which improves the connection between claims and evidence by focusing on learning critical, relevant short facts essential for effective scientific claim verification.

3 Methodology

We introduce FactGen, a novel approach designed to enhance the performance of claim verification solutions by leveraging automatically generated short facts extracted from the evidence. FactGen is a versatile tool that can be integrated into various claim verification methods, improving the robustness and reasoning capabilities of existing models. The core of FactGen relies on weakly labeled short facts, which are categorized as either *important* for verifying a given claim or *not important* for that purpose, which are used to train a multi-task learning-based model (FactDetect) for importance detection and claim verification.

3.1 Definition

Here, we formally define the primary task of fact generation and labeling: Given a claim statement (c) and corresponding evidence statement (e), our objective is to generate concise "facts" from the e. We denote this set of facts by $\mathcal{F}_e = \{f1, \ldots, f_m\}$. Each fact is subsequently labeled as either "important" or "not important," denoted as $y_{f_i} \in \{important, not\ important\}$.

It is crucial to emphasize that these facts are intentionally designed to be shorter in length compared to the original evidence (e). They serve as distilled pieces of information extracted from the broader context of the evidence. These succinct facts are intended to capture essential details or insights within the evidence, making them more manageable for claim verification tasks. An overview of the FactGen is given in Figure 1.

We next elaborate on the processes of short fact generation and weak labeling.

3.1.1 Short Fact Generation

To generate short facts from the evidence (e), we adopt a three-step approach.

1) Phrase matching: Initially, we extract matching phrases from both the claim (c) and the evidence, treating these phrases as potential answers to questions ($\mathcal{A} = (a_1^c, a_1^e), \ldots, (a_n^c, a_n^e)$). Phrases that "match" refers to a pair of phrases that convey similar meanings and/or are semantically similar. We call these answer pairs. To accomplish this, we employ LLM Vicuna-13B (Chiang et al., 2023)[2] for short fact generation. Importantly, we do not restrict the LLM to follow specific phrase rules such as length restrictions, using only entities or nouns, ensuring the capture of diverse answer pairs more likely to be relevant. The prompt used to extract answer pairs (phrase matches) is as follows:

Extract relevant keyphrase pairs from claim and evidence that determine the verdict. There can be more than one relevant keyphrase pairs.

2) Question Generation: After identifying the answer pairs, we move on to formulate concise questions. For each answer a_i^c in the pair (a_i^c, a_i^e), and corresponding to the claim c, we generate a question (q_i) as follows: The claim c serves as the context and a_i^c as the answer. to create a question based on these inputs—namely, the *context* and the *answer*. We only incorporate the answer from the claim (a_i^e) in this stage and not the answer from evidence (a_i^e). This is to 1) ensure the generation of a high-quality question that can be associated directly with the claim, achievable only by pairing

[2] Used following model checkpoint: https://huggingface.co/lmsys/vicuna-13b-v1.3

the claim with an internal answer, and 2) incorporate the essential context from the claim into the question, which will later be aligned with the a_i^e for short sentence generations. From the previous stage (Figure 1, the first claim phrase results in the question *What is the effect of cellphones for the kids?*. The prompt used to generate the question q_i is as follows:

Generate a question based on input context and the answer.

3) Short Fact Generation: Finally, We generate short fact sentences by pairing each question (q_i) with its corresponding evidence-based answer (a_i^e) which was extracted in the first step. These questions along with the answers are then converted into full sentences f_i. For example, the previous question and answer results in *Cellphones cause various mental health concerns for the kids.* Please note that not all the (q_i, a_i^e) pairs are *reasonable*. i.e., a generated q_i may not align semantically well with the a_i^e due to possible errors during generation or the structure of the context (c) therefore to ensure a reasonable and useful fact sentence we further refine these question and answer pairs for only the *reasonable* ones by incorporating a new reasoning-infused strategy. To do this we first focus on the "reasonability" of the generated questions from the previous step. Here, we query the LLM to determine if the (q_i, a_i^e) pair is not reasonable. If the output is "not reasonable", we move forward with other candidates i.e., (q_{i+1}, a_{i+1}^e) otherwise, the sentence f_i generated from pairing q_i and the evidence-based answer (a_i^e) is added to the candidate answers $\mathcal{A}_c$. This step is crucial for two reasons: 1) it serves to eliminate any unsuccessful question generations that can occur with LLMs (e.g., the failures can be due to the inconsistent and hallucinated generations), and 2) it helps FactGen to extract the most important question-answer pairs for claim verification. The prompt used in generating the short facts is as follows:

Generate full sentence from the given "question" and "answer". If the "question" is not answerable by the provided "answer", output "not reasonable".

3.2 Weak labeling

Labeling each generated fact as "important" or "not important" is a crucial step in the FactDetect process. After extracting the candidates we label a short fact sentence f_i as "important" if the cosine similarity between f_i and the claim c and f_i and evidence e exceeds a predefined threshold (t) and "not important" if not. More specifically:

$$sim(f_i, c, e) = \gamma(\cos(f_i, c) + \cos(f_i, e)) \quad (1)$$

$$y_{f_i} = \begin{cases} \text{“important”} & \text{if } sim(f_i, c, e) \geq t \\ \text{“not important”} & \text{otherwise} \end{cases}$$

Here γ is a hyperparameter and $\cos(.)$ is calculated using the Sentence Transformers (Reimers and Gurevych, 2019) embedding of f_i, c and e.

3.3 Joint Claim Verification and Fact Detection Framework

Because of the success of the full context training of claim verification tasks within state-of-the-art models such as MULTIVERS (Wadden et al., 2022b), PARAGRAPHJOINT (Li et al., 2021), and ARSJOINT (Zhang et al., 2021), we propose a similar enhancement approach. Our framework revolves around performing full context predictions by concatenating the claim (c), title (t), gold evidence (e), and all the facts in $\mathcal{F}_e$ with a special separator token to separate each fact in $\mathcal{F}_e$.

Our approach employs a multi-tasking-based strategy where the model is jointly trained to minimize a multitask loss defined as follows:

$$L = L_{cv} + \alpha L_{fact} \quad (2)$$

where L_{cv} represents the cross-entropy loss associated with predicting the overall claim verification task. Specifically, we predict $y(c, e)$ where:

$$y(c, e) \in \{\textit{support}, \textit{contradict}, \textit{nei}\} \quad (3)$$

by adding a classification head on the </s> token. In addition, L_{fact} denotes the binary cross-entropy loss for predicting whether each fact (f_i) is important to the claim (c) or not, and α is a hyperparameter. During inference, we only predict $y(c, e)$, setting aside the fact detection part.

3.4 Zero-shot prediction with LLMs

In the zero-shot approach, without the need for human annotated training dataset and finetuning a claim verification model, we leverage Large Language Models (LLMs) to extract the encoded knowledge in them using a prompting strategy aimed at eliciting the most accurate responses from them. This is achieved by adjusting the LLM to draw external knowledge in response to the prompt from the learned parameters. This is done as follows. Here, we introduce a zero-shot approach where we augment FactDetect generated short fact sentences $\mathcal{F}_\rceil$ into the prompt for claim verification through fact-detection: given c, e and $\mathcal{F}_e$ we first ask an LLM to detect the most important facts and then, by providing an explanation we ask it to predict the verdict $y(c, e)$.

This approach is similar to the popular Retrieval Augmented Generation (RAG) (Lewis et al., 2020) approach used in optimizing the output of the Large Language Models using external sources. A difference between our approach to the "retrieval" augmented approach is that we augment the candidate facts from the evidence into the input rather than retrieving any external knowledge.

The approach is formulated as follows: let $\mathcal{M}$ be a language model and $\mathcal{P}$ be the prompt. The $\mathcal{P}$ for the test inputs is generated by concatenating c, e and $\mathcal{F}_e$. We first extract *important facts* and then get the predicted the verdict. i.e., $p(y(c, e)|\mathcal{M}(\mathcal{P}))$.

4 Experiments

We evaluate the effect of including FactDetect within different claim verification models and encoders. To evaluate this, we first explain the datasets used and introduce the baseline models we compared to our approach.

4.1 Datasets

SciFact (Wadden et al., 2020) consists of expert annotated scientific claims from biomedical literature with their corresponding evidence sentences that were retrieved from abstracts. *SUPPORTED* claims are human-generated using citation sentences in abstracts and *CONTRADICTED* claims are negations of original claims.

SciFact-Open (Wadden et al., 2022a) constitutes a test collection specifically crafted for the assessment of scientific claim verification systems. In addition to the task of verifying claims against evidence within the SciFact domain, this dataset contains evidence originating from a vast scientific corpus of 500,000 documents.

HealthVer (Sarrouti et al., 2021) is a compilation of COVID-19-related claims from real-world scenarios that have been subjected to fact-checking using scientific articles. Unlike most available datasets where *contradict*ed claims are usually just the negation of the supported ones, in this dataset *contradicted* claims are themselves extracted from real-world claims. The claims in this dataset are more challenging compared to other datasets.

4.2 Baselines

We evaluate FactDetect in two settings: 1) supervised models and 2) unsupervised models. In supervised models, we either train the state-of-the-art models on *few-shot* or *full* training settings. For *un*supervised models, we use several best-performing LLMs for a *zero-shot* and *few-shot* prompting where we compare FactDetect prompting with different prompting strategies. For few-shot, we train on $k = 45$ training samples.

4.2.1 Supervised Baselines

We incorporate FactDetect as an add-on for a multi-task learning-based approach on two transformer-based encoders. We train the supervised models on NVIDIA RTX8000 GPU and overall model parameters do not exceed 1B. We set the learning rate to $2e-5$ and save the best model in 20 epochs. We choose 0.5 for the γ parameter and 10 for the α hyperparameter. The threshold t for the cosine similarity between fact sentences and claim and evidence is set to 0.6. [3]

We also evaluate the effectiveness of FactDetect on an end-to-end fact-checking MULTIVERS (Wadden et al., 2022b) approach.

Longformer (Beltagy et al., 2020) With the self-attention mechanism incorporated into this model and its ability to process long sequences, we use this encoder to concatenate short reasoning sentences into the claim along with additional context provided in the title (if any).

RoBERTa-Large (Liu et al., 2019) RoBERTa has proven to be an effective Language model to be used for training different classification tasks. We use this model as a base encoder in our experiments for this claim verification task.

MULTIVERS (Wadden et al., 2022b) is a state-of-the-art supervised scientific claim verification approach which uses Longformer as a base encoder for long-context end-to-end claim verification in a multi-task learning based approach where in addition to the claim and title it incorporates the whole document (abstract) for both claim verification and rationale (evidence) selection. We augment the FactGen sentences into the model as an input and train FactDetect on top of MULTIVERS in a multi-tasking based approach.

[3] We performed experiments with 5, 10 and 15 and the best performing value was 15.

Setting	Model	HealthVer			SciFact			SciFact-Open		
		F1/Acc	P	R	F1/Acc	P	R	F1/Acc	P	R
Few shot	RoBERTa-Large	38.7/34.7	39.1	38.3	37.0/43.0	36.3	37.8	36.3/30.1	35.8	36.8
	RoBERTa-Large+FactDetect	27.6/23.0	22.4	35.8	34.0/38.6	33.5	34.8	32.5/27.2	31.0	34.1
	Longformer	27.8/21.7	25.3	30.7	42.4/39.3	43.0	41.8	36.2/36.9	36.4	36.0
	Longformer + FactDetect	33.7/25.2	34.0	33.4	41.6/ 55.3	37.4	46.8	34.3/42.0	28.2	43.6
Full	RoBERTa-Large	43.9/28.6	52.0	37.9	48.5/64.6	43.3	55.2	38.1/45.1	30.7	**50.1**
	RoBERTa-Large+FactDetect	45.3 /25.6	61.0	36.0	50.8/66.5	58.1	45.2	**40.6 / 46.0**	35.1	48.2
	Longformer	53.1/35.7	58.1	49.1	54.7/49.3	63.5	49.0	40.4/27.2	**50.2**	33.7
	Longformer + FactDetect	56.2/44.7	59.2	53.0	63.0/65.0	67.2	59.2	40.4/38.4	34.4	40.3
	MULTIVERS	60.6/61.7	59.1	62.0	70.4/72.0	**70.8**	70.0	65.0/62.3	65.3	**64.8**
	MULTIVERS + FactDetect	**62.1/63.0**	**61.5**	**62.7**	**70.8/72.3**	70.3	**71.3**	**65.2**/61.3	**66.2**	64.4

Table 1: Overall performance comparison between different baselines without and with (+FactDetect) multi-task learning incorporating FactDetect. SciFact-Open results are reported in a zero-shot setting. The best results for each dataset are highlighted in bold and the best results within each pair (with and without FactDetect) are underlined.

4.2.2 Zero-shot baselines

LLMs serve as a robust source of knowledge and demonstrate impressive outcomes in various downstream tasks, especially in contexts where zero-shot and few-shot learning are employed. However, the effectiveness of these models heavily depends on the methods used to prompt their responses. Consequently, we evaluate state-of-the-art prompting methods both specific to the claim verification task and general task approaches, and compare them to our FactGen augmented prompting method.

Vanilla: We engage LLMs to assess the truthfulness of claims based on provided evidence and to offer justifications for their verdicts. This process is carried out without integrating any extra knowledge or employing a specific strategy.

Chain of Thought (CoT) (Wei et al., 2022) This popular approach involves breaking down the task into a series of logical steps presented to LLMs via prompts for the given context. We use this approach by providing the claim and evidence as input and instruct it to think step by step and provide explanation before predicting the verdict. We consequently add the *let's think step by step* instruction into the prompt and provide few shot examples where the verdict is given followed by a step by step reasoning explanations.

ProgramFC (Pan et al., 2023) is a newly introduced approach that converts complex claims into sub-claims which are then used to generate reasoning programs using LLMs that are executed and used for guiding the verification. We utilize the closed-book setting of this method with N=1. This approach is built for only two-label datasets where claims are either SUPPORTED or CONTRADICTED by evidence.

We compare these strategies in FlanT5-XXL (Chung et al., 2022), GPT-3.5 (gpt-3.5-turbo checkpoint), GPT-4 (gpt-4 checkpoint), Llama2-13B (Llama-2-13b-chat-hf checkpoint) (Touvron et al., 2023), and Vicuna-13B (Chiang et al., 2023) (vicuna-13b-v1.3 checkpoint). We perform experiments in few-shot promoting ($k = 5$).

4.3 Main Results

4.3.1 Supervised setup

We first report the results of *supervised* baselines with and without FactDetect incorporated in their training process in Table 1. We experiment with few-shot and full training setups. We observe that incorporating FactDetect into the Longformer and RoBERTa-Large encoders achieves the best performance in all three datasets (in bold). Specifically in the Full training setup, the average improvement in F1 when adding FactDetect to Longformer is 5.8% for HealthVer and 15.2% for SciFact. Longformer + FactDetect in the few-shot setting also improves the F1 score for HealthVer by 21.0%. However, overall we do not see a consistent performance improvement in the few-shot setting which suggests that FactDetect benefits from a larger training dataset. As mentioned earlier, the results of SciFact-Open dataset are reported in a zero-shot setting (with model trained on SciFact training dataset), resulting in lower performance. Additionally, SciFact-Open receives less benefit from FactDetect than other datasets even in the cases where it does improve results. We suspect that this is due to the

Datasets		SciFact		SciFact-Open		HealthVer	
Metrics		F1	F1 /wo NEI	F1	F1 /wo NEI	F1	F1 /wo NEI
FlanT5-XXL	Vanilla	69.0	83.1	67.4	88.6	51.3	61.2
	CoT	53.7	69.2	60.3	84.9	45.1	59.5
	FactDetect	62.5	79.4	54.0	81.2	44.6	63.4
Llama2-13B	Vanilla	19.8	41.0	24.0	39.0	29.0	59.5
	CoT	34.6	44.8	31.0	45.4	44.8	64.3
	FactDetect	39.0	57.0	35.2	38.0	**55.4**	63.9
Vicuna-13B	Vanilla	47.5	58.5	42.8	63.4	35.8	58.7
	CoT	47.3	66.1	52.2	73.4	44.7	54.7
	FactDetect	54.4	69.3	49.0	66.8	40.0	61.0
GPT-3.5	Vanilla	64.5	72.5	63.0	80.4	50.9	68.0
	CoT	69.8	81.8	62.9	84.5	52.1	67.9
	FactDetect	70.6	83.0	55.0	81.4	53.9	68.6
GPT-4	Vanilla	**86.2**	**92.3**	72.9	90.7	47.8	72.1
	CoT	83.2	88.1	**79.0**	96.1	44.1	70.7
	FactDetect	74.3	86.3	70.1	**98.0**	54.0	**75.0**
ProgramFC		–	45.0	–	78.0	–	62.9

Table 2: Using the in-context learning capabilities of LLMs we evaluate the effectiveness of different prompting strategies in 5 LLMs. We report results both with *NOT ENOUGH INFO* (NEI) data samples and without them. The best-performing strategy for each LLM is underlined and overall the best results are highlighted in bold for each dataset.

more complex nature of the dataset, with its having unique claims that are both *supported* and *contradicted* by different evidence sentences. The outcomes is consistent with the top-performing baseline, MULTIVERS. By integrating FactDetect into MULTIVERS, we achieve similar performance, despite the advantage of complete context encoding within this framework. Please note that the reported results were obtained from a single-run experiment.

4.3.2 Zero-shot setup

We additionally evaluate the performance of LLMs for the claim-verification task with FactDetect providing additional context for zero-shot claim verification. We used GPT-3.5 to generate programs for ProgramFC and extracted the verification with FlanT5-XL as described by Chung et al. (2022). We experimented with this model in two-label settings (*supported* and *contradicted*) because the original model is designed in binary verification mode. For a fair comparison, we report binary classification results in all our experiments. The results are reported in Table 2.

We observe that FactDetect substantially improves the performance of Llama2-13B in all three datasets compared to the best-performing baseline with an average performance gain of 14% in the F1 score. Similarly FactDetect shows improvements for GPT-3.5 in SciFact and HealthVer. Interestingly, FlanT5-XXL outperforms other prompting methods in the Vanilla setting. We suspect one main reason for this result is that we directly augment the output of short fact generation into the prompt as a list of sentences and ask the LLM to first extract the most important sentence among them for claim verification. Note that since this approach is fully unsupervised, there is a chance of hallucination which can directly impact the performance of the larger LLMs. This hypothesis also holds for the larger LLMs such as GPT-3.5 and GPT-4. Comparison between ProgramFC and baselines also shows the limited advantage in predicting verdicts in scientific claim verification datasets compared to the general claim verification datasets.

4.4 Assessing LLMs for FactGen

Here, we explore the impact of various underlying large language models (LLMs) on the task of claim verification by regenerating short fact sentences using three different LLMs: Mistral-7B[4], GPT-3.5, and Vicuna-13B. The zero-shot experiments were conducted using the same models as in Section 4.3.2 (excluding GPT-4), alongside a supervised experiment involving a Longformer + FactDetect. The findings are depicted in Figure 3.

[4] employed checkpoint: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

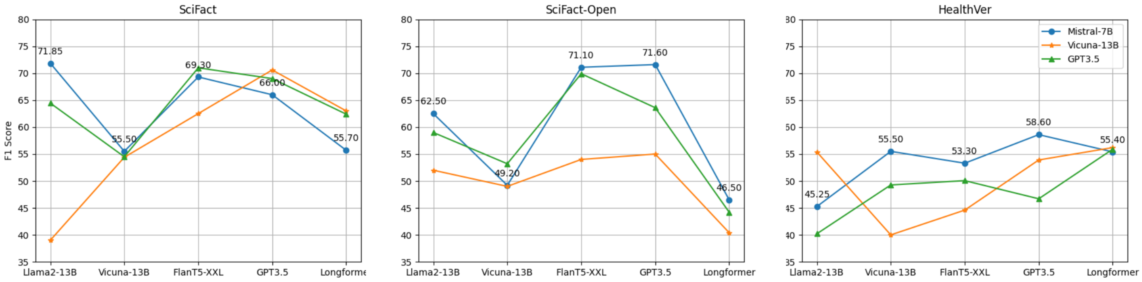

Figure 3: Comparing the F1 Score of three testsets trained (Longformer) or augmented with LLM-based FactGen datasets.

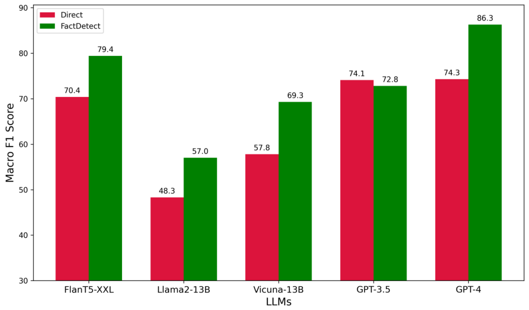

Figure 4: Comparison in Macro F1 score for SciFact between FactDetect-generated short facts and direct generation.

The results indicate that the choice of LLM for generating short facts has a minimal impact on the performance of the supervised model (Longformer+FactDetect). In contrast, the zero-shot experiments exhibited more pronounced performance variations dependent on the LLM utilized for fact generation. Notably, Llama2-13B and GPT-3.5 demonstrated heightened sensitivity to the choice of LLM in the fact generation (FactGen) process. Furthermore, we observed an enhancement in the overall efficacy of the claim verification task when employing Mistral-7B and GPT-3.5 for FactGen. These findings suggest that zero-shot, prompt-based claim verification highly benefits from the utilization of higher-quality LLMs.

4.5 Effectiveness of FactGen

Here, we experiment and compare two short fact generation approaches. The first approach is the Direct approach, where we ask Vicuna-13B to generate short sentences from evidence e (we give 5 examples as few-shot prompting). The second approach is generating short facts using FactDetect. We collect the short sentences for each piece of evidence in a claim-evidence (CE) pair, for the SciFact dataset (dev set) and run experiments in the unsupervised setup. Macro F1 score comparisons between Direct and FactDetect-augmented predictions are given in Figure 4.

Overall, FactDetect-augmented prompts are performing better compared to the Direct approach. These results suggest the usefulness of the three-step approach compared to the baseline sentence generation approach.

5 Error Analysis

To better understand the errors made by FactDetect in the Zero-Shot setting, we manually analyzed the test set predictions where the LLM made incorrect predictions and found that Larger LLMs are more sensitive towards hallucinations.

When the information provided by short fact generation doesn't fully align with the evidence and claim, larger language models can detect this mismatch in short sentences and accordingly produce a not enough info (nei) response. Specifically, in instances of misclassification, FlanT5-XXL incorrectly labels claims as (nei) 63% of the time, and GPT-4 does so 68% of the time. Conversely, Llama2-13B and Vicuna-13B incorrectly assign the (nei) label about 10% of the time, more frequently misclassifying other responses as "support". We will tackle this issue in future studies.

6 Conclusion and Future Work

In this work, we propose FactGen, an effective short fact generation technique, for comprehensive and high-quality condensed small sentences derived from evidence. With the relevance-based weak-labeling approach this dataset can be augmented to any state-of-the-art claim verification model as a multi-task learning to train fact detection with FactDetect. The effectiveness of this model has been demonstrated in both fine-tuned and prompt-based models. Our results suggest that FactDetect incorporated claim-verification task in a supervised setting consistently improves performance on average by 10.5% in full training setup.

7 Limitations

A drawback of our method is the reliance on a generative language model, LLMs for producing short fact sentences throughout the entire process. Despite employing Vicuna-13B, which is among the top open-source LLMs available, the factual accuracy and overall quality of the generated content are bounded by the capabilities of this particular model. Consequently, any inaccuracies from the model could impact the effectiveness of the end-to-end claim verification system. Overcoming this obstacle is a crucial direction for future research.

Furthermore, a limitation of zero-shot FactDetect in real-world claim-verification systems is the need to augment the short sentences into the prompt, which is an additional step and can be time-consuming in the claim verification task. However, this problem is mitigated when we fine-tune a claim-verification system with FactDetect in the training phase, and during inference, we just use the claim and evidence as input.

8 Ethics Statement

Biases. We acknowledge the possibility of bias in generated outputs from the trained LLM. However, this is beyond our control.

Potential Risks. Our approach can be used for automated fact-checking. However, they could also be used by malicious actors to manipulate and attack fact-checking models. A possible future direction is to detect such malicious actions before deployment.

Environmental Impact. Training and using LLMs involves considerable computational resources, including the necessity for GPUs or TPUs during training or inference which can have an impact on the environment. However, we trained our datasets on relatively smaller language models with less than 1B parameters and we used LLMs for inference only which has negligible negative effect on the environment.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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

Limeng Cui, Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. Defend: A system for explainable fake news detection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2961–2964, New York, NY, USA. Association for Computing Machinery.

Shih-Chieh Dai, Yi-Li Hsu, Aiping Xiong, and Lun-Wei Ku. 2022. Ask to know more: Generating counterfactual explanations for fake claims. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2800–2810.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims.

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

Shailza Jolly, Pepa Atanasova, and Isabelle Augenstein. 2022. Generating fluent fact checking explanations with unsupervised post-editing. *Information*, 13(10).

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

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

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management*, CIKM '21. Association for Computing Machinery.

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

Xiangci Li, Gully A Burns, and Nanyun Peng. 2021. A paragraph-level multi-task learning model for scientific fact-verification. In *SDU@ AAAI*.

Wenhan Xiong Min-Yen Kan William Yang Wang Liangming Pan, Wenhu Chen. 2021. Zero-shot fact verification by claim generation. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, Online.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004, Toronto, Canada. Association for Computational Linguistics.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. *arXiv preprint arXiv:2106.03794*.

Mourad Sarrouti, Asma Ben Abacha, Yassine M'rabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

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*.

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

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022a. SciFact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. *arXiv preprint arXiv:2310.05253*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Fan Yang, Shiva K. Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D. Ragan, Shuiwang Ji, and Xia (Ben) Hu. 2019. Xfake: Explainable fake news detector with visualizations. In *The World Wide Web Conference*, WWW '19, page 3600–3604, New York, NY, USA. Association for Computing Machinery.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021. Abstract, rationale, stance: a joint model for scientific claim verification. *arXiv preprint arXiv:2110.15116*.