LONGCHECKER: Improving scientific claim verification by modeling full-abstract context

Anonymous ACL submission

Abstract

The spread of scientific mis- and dis-information has motivated the development of datasets and models for the task of scientific claim verification. We address two modeling challenges associated with this task. First, existing claim verification systems make predictions by extracting an evidentiary sentence (or sentences) from a larger context, and then predicting whether this sentence supports or refutes the claim in question. This can be problematic, since the meaning of the selected sentence may change when interpreted outside its original context. Second, given the difficulty of collecting high-quality fact-checking annotations in expert domains, there is an unaddressed need for methods to facilitate zero / few-shot domain adaptation. Motivated by these challenges, we develop LONGCHECKER. Given a claim and evidence-containing abstract, LONGCHECKER predicts a fact-checking label and identifies evidentiary sentences in a multi-task fashion based on a shared encoding of all available context. This approach enables LONGCHECKER to perform domain adaptation by leveraging weakly-supervised in-domain data. We show that LONGCHECKER achieves state-of-the-art performance on three datasets, and conduct analysis to confirm that its strong performance is due to its ability to model full-abstract context.

1 Introduction

The task of scientific claim verification requires a system to assess the veracity of a scientific claim against a corpus of documents. The proliferation of mis- and dis-information on the web – particularly as it relates the COVID-19 pandemic (Pennycook et al., 2020; Naeem et al., 2020) – has motivated the release of a number of new datasets for this task (Saakyan et al., 2021; Sarrouti et al., 2021; Wadden et al., 2020; Kotonya and Toni, 2020), accompanied by advances in model performance (Pradeep et al., 2021; Li et al., 2021; Zhang et al., 2021).

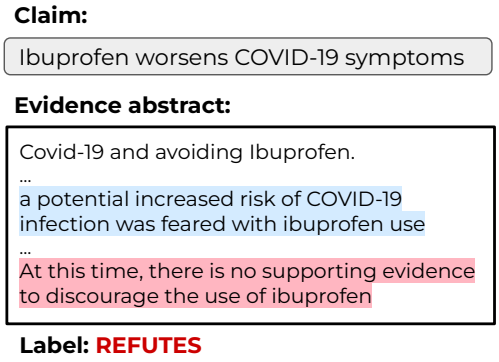

Figure 1: A claim from the HealthVer data set, refuted by a research abstract. The sentence in red is a *rationale*, which reports a finding that REFUTES the claim. However, this finding cannot be interpreted properly without the context in blue, which specifies that the finding applies to ibuprofen as a potential treatment for COVID symptoms. LONGCHECKER incorporates the full context of the evidence-containing abstract when predicting fact-checking labels.

One commonality among existing models is that they verify claims using a pipeline approach. Given a claim and an abstract that may contain evidence, they first extract rationales from the abstract which contain evidence sufficient to entail or contradict the claim, when taken in the context of the abstract. Then, they predict a fact-checking label based on the selected rationales, taken out-of-context. This approach has two important shortcomings. First, the rationales containing evidence may lack information required to make a prediction out-of-context; for instance, they may contain acronyms or pronouns, or lack qualifiers that specify the scope of the finding. Figure 1 provides an example. This challenge has previously been observed in work on scientific literature understanding (Nye et al., 2020), and more generally in the task of sentence *decontextualization* (Choi et al., 2021).

Second, pipeline models require training data annotated with both sentence-level rationales and abstract-level labels. While sentence-level anno-

tations for scientific claim verification are quite costly, abstract-level labels can be created cheaply using high-precision heuristics For instance, the titles of research papers often make claims that are supported by their abstracts. Ideally, models should be able to take advantage of these additional abstract-level labels without requiring that they be paired with sentence-level rationale annotations.

Motivated by these challenges, we develop the LONGCHECKER system: given a claim and evidence-containing abstract, LONGCHECKER encodes the entire claim / abstract context in a single long sequence. The resulting context exceeds the 512-token window common to BERT-style (Devlin et al., 2019) transformer architectures between 12% and 43% of the time, depending on dataset. To accommodate this, LONGCHECKER builds on the Longformer model, which has been successfully applied to related tasks, such as question answering, involving long-document context (Beltagy et al., 2020; Pradeep et al., 2021).

Longformer uses special sentinel tokens to construct globally-contextualized representations of the entire context, and each individual sentence in the abstract. We use the representations of these tokens to predict an abstract-level fact-checking label and sentence-level rationale labels, respectively. We find that this modeling approach improves performance on three datasets for scientific claim verification over two state-of-the-art baselines, one of which has more than 10x the parameters than our system. In addition, it is able to effectively leverage weakly-supervised in-domain data for zero/few-shot domain adaptation, outperforming a state-of-the-art pipeline model trained using heuristically-labeled rationales.

In summary, we make the following contributions: (1) We introduce LONGCHECKER, a multitask system for full-context scientific claim verification, and find that it outperforms two state-of-the-art baselines on three datasets. (2) We propose a set of simple heuristics to assign weak fact-checking labels to a large collection of research abstracts, and find that training LONGCHECKER on these weakly-labeled data improves average zero-shot performance by 24 F1 across our three datasets. (3) We conduct ablations and analysis confirming that LONGCHECKER outperforms existing systems due to its ability to model full-abstract context when making fact-checking predictions.

2 Background: Scientific claim verification

2.1 The scientific claim verification task

We will use the definition of scientific claim verification from the SCIFACT task (Wadden et al., 2020). We provide a brief review of the task and refer the reader to that work for more detail. Some other works have cast scientific claim verification as a sentence-level natural language inference (NLI) task; in §4, we describe how we process these datasets to be compatible with the task as considered in this work.

Task definition Given a claim c and a collection of *candidate abstracts* which may contain evidence relevant to c, the scientific claim verification task requires a system to predict a *label* $y(c, a) \in \{\text{SUPPORTS}, \text{REFUTES}, \text{NEI}\}$[1], which indicates the relationship between c and a for each candidate a. For all abstracts labeled SUPPORTS or REFUTES, the system must also identify *rationales* $R(c, a) = \{r_1(c, a), \ldots, r_n(c, a)\}$, where each $r_i(c, a)$ is a sentence from a that either entails or contradicts the label $y(c, a)$[2]. The rationales may not be self-contained, and may require additional context from elsewhere in the abstract to resolve coreferential expressions or acronyms, or to determine qualifiers specifying experimental context or study population[3]. Examples of this situation are provided in Figure 1 and Appendix A.3.

Evaluation The SCIFACT task reports four evaluation metrics. We have found that two of these metrics are sufficient to convey the important findings for our experiments: (1) *abstract-level label-only* evaluation computes the model's F1 score in identifying abstracts that SUPPORT and REFUTE each claim. Predicting the correct label $y(c, a)$ is sufficient; models do not need to provide rationales. (2) *Sentence-level selection+label* evaluation computes the point-wise product of the model's F1 score in identifying the rationales $R(c, a)$, with the model's abstract-level label $y(c, a)$; this rewards precision in identifying exactly which sentences contain the evidence justifying the label. We will refer to these two metrics as "abstract" and "sentence" evaluation, respectively.

[1] NEI stands for "Not Enough Info".

[2] This rationale definition is simplified slightly from the one presented in Wadden et al. (2020).

[3] This convention is consistent with related tasks in rationalized NLP for biomedical literature, such as Lehman et al. (2019); DeYoung et al. (2020).

Retrieval settings For *open* scientific claim verification, the system must retrieve candidate abstracts from a corpus of documents. In the *abstract-provided* setting, candidate abstracts for each claim are given as input. We describe the retrieval settings for all datasets in §4.1.

Supervision settings We consider three supervision settings. In the *fully-supervised* setting, models may train on all claims from the target dataset. In the *zero-shot domain adaptation* setting, models may not train on any in-domain fact-checking data, though they may train on general-domain fact-checking data and other available scientific datasets. In the *few-shot domain adaptation* setting, models may train on 45 claims from the target dataset.

Most existing work on scientific fact-checking examines the fully-supervised setting. An exception is Lee et al. (2021), which uses language model perplexity as a measure of claim veracity.

2.2 Datasets

A number of datasets for scientific claim verification have been released in roughly the past year. COVID-Fact (Saakyan et al., 2021) and HealthVer (Sarrouti et al., 2021) verify claims related to COVID-19 against scientific literature. PUBHEALTH (Kotonya and Toni, 2020) verifies public health claims against news and web sources. SCIFACT (Wadden et al., 2020) verifies claims made in citations in scientific papers. CLIMATE-FEVER (Diggelmann et al., 2020) uses Wikipedia to verify claims about climate change. In this work, our focus is on verifying claims against scientific research literature. We therefore perform experiments on the COVID-Fact, HealthVer, and SCIFACT datasets. Additional details on these datasets are included in §4.1.

2.3 Models

Motivated in part by the SCIVER shared task (Wadden and Lo, 2021) and leaderboard, a number of models have been developed for SCIFACT (the focus of the shared task). The two strongest systems on the shared task were VERT5ERINI (Pradeep et al., 2021) and PARAGRAPHJOINT (Li et al., 2021), which we adopt as baselines and describe further in §4.4. More recently, ARSJOINT (Zhang et al., 2021) achieved performance competitive with these two systems.

Pipeline claim verification Given a claim c and candidate abstract a, these models make predictions in two steps. First, they predict rationales $\widehat{R}(c,a) = \{\widehat{r}_1(c,a), \ldots, \widehat{r}_n(c,a)\}$ likely to contain evidence. Then, they make a label prediction $\widehat{y}(c, \widehat{R}(c,a))$ based on the predicted rationales, ignoring the rest of the abstract a. Written another way, they make label predictions by approximating $\widehat{y}(c,a)$ with $\widehat{y}(c, \widehat{R}(c,a))$. We will refer to this approach as the *pipeline approach* to scientific claim verification. Figure 1 demonstrates how this approach can fail when a rationale does not provide all the necessary context required for a prediction.

System details VERT5ERINI uses two separate T5-3B models for the two pipeline components. PARAGRAPHJOINT and ARSJOINT encode the title and full abstract (truncating to 512 tokens to fit within the BERT window), and perform rationale selection and label prediction based on this shared encoding. However, only the encodings of the predicted rationales are used for label prediction.

3 The LONGCHECKER model

In §3.1, we describe our modeling approach. We address the problem of out-of-context rationales raised in §2.3 by making a simple modeling change: instead of approximating $\widehat{y}(c,a)$ with $\widehat{y}(c, \widehat{R}(c,a))$, we predict $\widehat{y}(c,a)$ directly based on an encoding of the entire claim and abstract. In §3.2, we explain how this modeling approach facilitates few-shot domain adaptation using weakly-labeled scientific documents.

3.1 Full-context claim verification

Long-document encoding Given a claim c and candidate abstract a consisting of title t and sentences $s_1, \ldots, s_n$, we concatenate the inputs separated by </s> tokens. The </s> token following each sentence s_i is notated as $</s>_i$:

$$<s>\ c\ </s>\ t\ </s>\ s_1\ </s>_1\ \ldots s_n\ </s>_n$$

This model input sometimes exceeds the 512-token limit common to transformer-based language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019); see Table 1 for details on how frequently this occurs. Therefore, we use the Longformer model (Beltagy et al., 2020) as our encoder. We assign global attention to the <s> token, as well as all tokens in c and all </s> tokens.

Multi-task rationale selection and label prediction Given the full-context Longformer encoding, we predict whether sentence s_i is a rationale via a binary classification head, consisting of two feed-forward layers followed by a two-way softmax, on

top of the globally-contextualized token $\texttt{</s>}_i$. Sentences assigned rationale scores greater than 0.5 are included in $\widehat{R}(c, a)$.

Similarly, we predict the overall fact-checking label $\widehat{y}(c, a)$ by adding a three-way classification head over the encoding of the <s> token. Since the <s> token is trained with global attention, the model makes predictions based on a representation of the entire claim and abstract, rather than only having access to the rationales $\widehat{R}(c, a)$. We refer to the approach taken by LONGCHECKER as the *multi-task* approach to claim verification.

During training, we compute the cross-entropy losses for the label and rationale predictions, and train to minimize the multi-task loss:

$$L = L_{\text{label}} + \lambda_{\text{rationale}} L_{\text{rationale}},$$

where $\lambda_{\text{rationale}}$ is tuned on the dev set.

Candidate abstract retrieval For datasets that require retrieval of candidate abstracts, we rely on the VERT5ERINI (Pradeep et al., 2021) retrieval system, which achieved state-of-the-art performance on the SCIVER shared task (SCIVER used the SCIFACT dataset for evaluation). This model first retrieves abstracts using BM25 (Robertson and Zaragoza, 2009), then refines the predictions using a neural re-ranker based on Nogueira et al. (2020), which is trained on the MS MARCO passage dataset (Campos et al., 2016).

3.2 Training for domain adaptation

Three types of training data are available to train scientific claim verification systems. (1) In-domain fact-checking annotations are the "gold standard", but they are expensive to create and require expert annotators. (2) General-domain fact-checking datasets like FEVER (Thorne et al., 2018) are abundantly available, but generalize poorly to scientific claims (see §5.1). (3) Scientific documents – either unlabled or labeled for different tasks – are abundant, and high precision heuristics can be used to generate document-level fact-checking labels $y(c, a)$ for these data. We describe two such heuristics in §4.2.

Given these three sources, we train LONGCHECKER as follows: we first pretrain on a combination of general-domain fact-checking annotations, combined with weakly-labled in-domain data[4]. Then, we finetune on the target scientific fact-checking dataset. The multi-task architecture of LONGCHECKER is ideally suited to this strategy, since the model can be trained on data with or without rationale annotations. When rationales are not available, we set $\lambda_{\text{rationale}} = 0$ in the loss function and train as usual. By contrast, training a pipeline model requires generating rationale annotations $R(c, a)$, which is relatively low-precision (see §4.2).

[4] "Pretraining" is a slight abuse of terminology. We use "pretraining" as shorthand for "training on the target task with out-of-domain and / or weakly-supervised labels".

4 Experimental setup

We describe our datasets, model training procedure, and baselines.

4.1 Scientific claim verification datasets

We experiment with three scientific claim verification datasets. Table 1 provides a summary of important dataset characteristics. Preprocessing steps and additional statistics for all datasets can be found in Appendix A. HealthVer and COVID-Fact were originally released in an NLI format, pairing claims with (out-of-context) evidentiary sentences. We convert to our task format by identifying the abstracts in the CORD-19 corpus containing these sentences, and label them as rationales.

We use the following terminology: an *atomic* claim makes an assertion about a single property of a single entity, while a *complex* claim may make assertions about multiple properties or entities.

SCIFACT claims (Wadden et al., 2020) were created by re-writing citation sentences occurring in biomedical literature into *atomic* claims, which were verified against the abstracts of the cited documents. REFUTED claims were created by manually negating the original claims. Abstracts that were cited but which annotators judged not to contain evidence were labeled NEI. SCIFACT requires retrieval of candidate abstracts from a corpus.

HealthVer (Sarrouti et al., 2021) consists of COVID-related claims obtained by extracting snippets from articles retrieved to answer questions from TREC-COVID (Voorhees et al., 2020), and verifies them against abstracts from the CORD-19 corpus (Wang et al., 2020). Claims in HealthVer may be *complex*. REFUTED claims occur naturally in the article snippets. HealthVer provides candidate abstracts for each claim, but some of these candidates do not contain sufficient information to support a SUPPORTS/ REFUTES verdict and are labeled NEI.

Dataset	Domain	Claim source	Open	Has NEI	Claim complexity	Negation method	Train claims	Eval claims	> 512 tokens
HealthVer	COVID	TREC-COVID	✗	✓	Complex	Natural	1,622	230	14.9%
COVID-Fact	COVID	Reddit	✗	✗	Complex	Automatic	903	313	12.4%
SciFact	Biomed	Citations	✓	✓	Atomic	Human	1,109	300	27.4%
Fever	Wiki	Wikipedia	-	✓	Atomic	Human	130,644	-	33.2%
PubmedQA	Biomed	Paper titles	-	✓	Complex	Automatic	58,370	-	12.1%
EvidenceInference	Biomed	ICO prompts	-	✓	Atomic	Automatic	7,395	-	42.7%

Table 1: Summary of datasets used in experiments. The top group of datasets are scientific claim verification datasets, and the bottom group are for pretraining. Datasets with a ✓ for "Open" require that candidate abstracts be retrieved from a corpus; those with a ✗ provide candidate abstracts as input. Dataset with a ✓ for "Has NEI" require three-way (SUPPORTS/ REFUTES/ NEI) label prediction, while those with an ✗ are (SUPPORTS/ REFUTES) only. The "> 512 tokens" column indicates the percentage of claim / abstract contexts that exceed 512 tokens.

COVID-Fact (Saakyan et al., 2021) collects claims about COVID-19 scraped from a COVID-19 subreddit, and verifies them against linked scientific papers, as well as documents retrieved via Google search. Claims in COVID-Fact may be *complex*, and candidate abstracts for each claim are provided. All candidates either SUPPORT or REFUTE the claim. Claim negations were created automatically by replacing salient words in the original claims, and as a result the labels $y(c, a)$ are somewhat noisy (see Appendix A).

4.2 Pretraining datasets

We briefly describe our pretraining datasets and the weak supervision heuristics used to construct them. Detailed descriptions of these heuristics can be found in Appendix A.1.

FEVER (Thorne et al., 2018) consists of claims created by re-writing Wikipedia sentences into *atomic* claims, verified against Wikipedia articles.

EVIDENCEINFERENCE (Lehman et al., 2019; DeYoung et al., 2020) was released to facilitate understanding of clinical trial reports, which examine the effect of an *intervention* on an *outcome*, relative to a *comparator* ("ICO" elements). The dataset contains ICO *prompts* paired with (1) labels indicating whether the outcome *increased* or *decreased* due to the intervention, and (2) rationales justifying each label. We use rule-based heuristics to convert these prompts into claims – for instance "[intervention] increases [outcome] relative to [comparator]".

PUBMEDQA (Jin et al., 2019) was released to facilitate question-answering over biomedical research abstracts. We use the PQA-A subset, which is a large collection of biomedical abstracts with "claim-like" titles – for instance, "Vitamin B6 supplementation increases immune responses in critically ill patients." We treat the paper titles as claims and the matching abstracts as the evidence sources.

To train pipeline models on these instances, we create weakly-supervised rationales by selecting the sentences with highest similarity to the claim as measured by cosine similarity of Sentence-BERT embeddings (Reimers and Gurevych, 2019). We use these annotations only when training pipeline models. They are not used by LONGCHECKER. To estimate the precision of rationale labeling heuristic, we predict rationales in the same fashion for our supervised datasets and compute the Precision@1 with which this method identifies gold rationales. The scores are relatively low: 49.4, 48.8, and 43.4 for SCIFACT, COVID-Fact, and HealthVer respectively.

4.3 Model training

Our fully-supervised training procedure consists of pretraining on the three datasets from §4.2, followed by finetuning on one of the target datasets from §4.1. For zero-shot experiments, we perform pretraining only. For few-shot experiments, we pretrain followed by finetuning on 45 target examples.

We found that negative sampling was important to achieve good precision on SCIFACT, which requires document retrieval. We train with 20 negative samples / claim and retrieve 10 abstracts / claim at inference time. For the other datsets, no negative sampling was used.

Additional details including batch sizes, learning rates, number of epochs, etc. can be found in Appendix B.

4.4 Baseline systems

We use PARAGRAPHJOINT and VERT5ERINI as our baseline systems. When making predictions on SCIFACT, we use publicly available model checkpoints available for each system. For HealthVer and

		HealthVer						COVID-Fact						SciFact					
		Abstract			Sentence			Abstract			Sentence			Abstract			Sentence		
Model	Params	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
VERT5ERINI	5.6B	71.3	74.0	72.6	65.6	61.2	63.3	76.6	52.7	62.4	44.8	27.2	33.9	64.0	73.0	68.2	60.6	66.5	63.4
PARAGRAPHJOINT	360M	75.0	68.3	71.5	69.9	60.6	64.9	71.5	68.1	69.8	41.4	40.3	40.8	75.8	63.5	69.1	68.9	54.6	60.9
LONGCHECKER	440M	78.9	76.3	**77.6**	72.0	66.8	**69.3**	77.3	77.3	**77.3**	41.7	45.9	**43.7**	73.8	71.2	**72.5**	67.4	67.0	**67.2**

Table 2: Performance of LONGCHECKER and baselines in the fully-supervised setting. The number of parameters in each model is reported in the "Params" column; VERT5ERINI is roughly 10x larger than the other two systems. We report performance using abstract-level and sentence-level evaluation as defined in §2.1. LONGCHECKER outperforms the baselines on all datasets.

			HealthVer						COVID-Fact						SciFact					
			Abstract			Sentence			Abstract			Sentence			Abstract			Sentence		
Setting	Model	Sci	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero-shot	PARAGRAPH JOINT	✗	66.7	3.0	5.8	33.3	0.9	1.8	73.9	5.4	10.0	39.1	1.7	3.2	59.5	11.3	18.9	48.9	6.2	11.0
		✓	72.3	14.4	24.0	22.9	2.7	4.9	51.3	37.9	43.6	31.5	16.0	21.3	52.9	32.4	40.2	36.4	14.9	21.1
	LONG CHECKER	✗	80.0	0.7	1.3	66.7	0.4	0.7	95.8	14.5	25.2	63.5	6.2	11.2	83.8	14.0	23.9	64.9	6.5	11.8
		✓	60.6	20.5	**30.7**	25.0	4.6	**7.8**	48.8	45.7	**47.2**	32.7	18.5	**23.6**	49.0	44.6	**46.7**	39.0	21.6	**27.8**
Few-shot	PARAGRAPH JOINT	✗	57.2	38.2	45.9	35.0	23.3	28.0	65.7	29.6	40.9	41.1	13.8	20.7	50.0	39.6	44.2	32.1	23.2	27.0
		✓	62.7	41.6	50.0	46.0	29.3	35.8	73.3	60.6	66.3	44.3	30.6	36.2	44.4	51.4	47.6	33.0	35.1	34.0
	LONG CHECKER	✗	56.4	50.8	53.4	35.6	28.4	31.6	74.5	74.5	**74.5**	39.5	45.1	**42.1**	72.4	43.7	54.5	48.8	32.4	39.0
		✓	63.6	47.9	**54.7**	44.0	30.7	**36.1**	71.3	68.1	69.7	40.5	35.1	37.6	76.4	54.0	**63.3**	51.7	40.3	**45.3**

Table 3: Performance of LONGCHECKER and PARAGRAPHJOINT in the zero-shot and few-shot settings. Rows where "Sci" is marked ✓ indicate that the model was pretrained on scientific data. Rows marked ✗ indicate pretraining on FEVER only. The results show that in-domain pretraining improves performance, and that LONGCHECKER outperforms PARAGRAPHJOINT.

COVID-Fact, we use the training code provided by the authors as-is, without adjusting training parameters. Additional details for the baselines can be found in Appendix B.4. In order to compare fairly with the two baselines (which were designed for SCIFACT), we performed model development for LONGCHECKER on SCIFACT as well, and did not modify the training procedure for the other two datasets.

5 Experimental results

We present the results of our experiments. We find that LONGCHECKER exhibits state-of-the-art performance on all datasets and settings, and that training on weakly-supervised scientific data substantially improves zero/few-shot performance.

5.1 Main Results

LONGCHECKER achieves state-of-the-art performance Table 2 shows the fully-supervised performance of LONGCHECKER and the two baselines on our three target datasets. A few trends are apparent. First, LONGCHECKER outperforms the baselines on all datasets, supporting our hypothesis that full-abstract context is often helpful when making labeling decisions (see §6 for further evidence of this). Second, predicting the overall relationship between a claim and abstract is easier than identifying the specific rationales supporting the relationship. Finally, while all models score within roughly six points of each other on HealthVer and SCIFACT, variability is much greater on COVID-Fact. We suspect that this is due to the automatically-generated nature of COVID-Fact negations.

Weakly-labeled in-domain data facilitates few-shot domain adaptation To understand the impact of weakly-supervised in-domain data on model performance in the zero/few-shot settings, we compare the results of pretraining on FEVER, compared to pretraining on all three datasets described in §4.2. Due to the expense of pretraining VERT5ERINI, we use PARAGRAPHJOINT as the baseline for this experiment.

We observe that including scientific data during pretraining substantially increases performance, for both models, in both the few-shot and zero-shot settings. For LONGCHECKER in the zero-shot setting, it leads to an average improvement of 24.7 abstract-

Model	Training	Abstract			Sentence		
		P	R	F1	P	R	F1
Pipeline	Target-only	75.3	71.3	73.2	69.5	65.3	67.4
	Full	74.2	71.5	72.8	67.9	63.6	65.7
multi-task	Target-only	68.8	73.5	71.0	65.9	65.9	65.9
	Full	78.9	76.3	**77.6**	72.0	66.8	**69.3**

Table 4: Ablations on the HealthVer test set. The "Pipeline" model uses two separate Longformer models for rationale selection and label prediction, while "multi-task" denotes our final system. "Two-stage" indicates pretraining followed by finetuning on the target dataset, while "Target-only" training uses the target dataset only. Multi-task modeling with two-stage finetuning leads to the best performance.

	Stand-alone		Context-dependent		All	
	Abst	Sent	Abst	Sent	Abst	Sent
VERT5ERINI	**87.8**	75.6	75.2	67.0	**79.7**	70.0
PARAGRAPHJOINT	85.0	**77.4**	73.1	64.0	77.3	68.8
LONGCHECKER	80.5	69.6	**78.4**	**71.0**	79.2	**70.5**
Count	43		85		128	

Table 5: Performance of models on SCIFACT instances with rationales that are "Stand-alone" (can be interpreted correctly out-of-context) and "Context-dependent" (require abstract context to be interpreted correctly). The "All" column shows performance on all instances combined. "Abst" and "Sent" indicate abstract-level and sentence-level F1. LONGCHECKER exhibits the strongest performance on context-dependent rationales.

level F1. For both models, training on FEVER alone appears lead to under-prediction and low recall, suggesting that entailment patterns learned on Wiki-domain text do not generalize readily to scientific literature.

While the improvements are not quite as dramatic in the few-shot setting, scientific data helps in all cases except COVID-Fact with LONGCHECKER. In the fully-supervised setting, pretraining on scientific data no longer made a noticeable difference; we omit these reults for brevity.

LONGCHECKER outperforms PARAGRAPHJOINT in both the few- and zero-shot settings, across all datasets. This is unsurprising, given the relatively low precision of our method for selecting weakly-supervised rtaionales (§4.2), and indicates that the multi-task approach taken by LONGCHECKER may be promising for quickly adapting fact-checking models to new specialized domains or scientific subfields.

Finally, we observe that HealthVer appears to be the most challenging dataset of the three. Few-shot abstract-level F1 scores for COVID-Fact and SCIFACT are generally within 10 F1 of their fully-supervised values, while the gap is a bit over 20 F1 for HealthVer. This may be due to the high complexity of HealthVer claims.

5.2 Ablations

We conduct ablations on the HealthVer dataset to characterize the contributions of the multi-task architecture and two-stage training procedure to the overall performance of LONGCHECKER. First, we compare our multi-task approach to a "pipeline" version of LONGCHECKER, where we use one Longformer model to select rationales, and a second one to make label predictions based on the selected rationales. Second, we compare the performance of LONGCHECKER trained on the target dataset only (no pretraining) with models trained using the full two-stage approach described in §4.3.

The results are shown in Table 4. Interestingly, multi-task learning and two-stage finetuning work the best in combination, but they do not work well separately. This is likely because label prediction is a more difficult task in the multi-task setup, since the model input is much longer. While the model can ultimately achieve better performance, it takes more data to train.

6 Analysis

We collect additional annotations on the SCIFACT test set to characterize the improvements made by LONGCHECKER relative to the baseline systems, and to assess model performance relative to the "upper bound" set by human agreement. For this analysis, we evaluate models in the "abstract-provided" setting.

LONGCHECKER outperforms baselines on instances requiring abstract-level context To determine whether LONGCHECKER's stronger performance is in fact due to its modeling of context missed by previous systems, we collect annotations for 128 claim / evidence pairs from the SCIFACT test set[5]. For each pair, the annotators indicated whether the rationales justifying the fact-checking label were "context-dependent" – i.e. they entailed (or refuted) the claim only when taken in the context of the abstract – or "stand-alone" – i.e. they

[5]These annotations are available at `[anonymized]`.

also entailed the claim when taken in isolation. Examples of "context-dependent" rationales are provided in Figure 1 and Appendix A.3.

The results are shown in Table 5. The majority of annotated instances (85 / 128) were judged to be context-dependent. LONGCHECKER performs roughly the same on stand-alone and context-dependent examples, whereas the two baselines exhibit performance drops of roughly 10 F1 on context-dependent examples. This provides strong evidence that LONGCHECKER's improvements are, in fact, enabled by its multi-task approach. Interestingly LONGCHECKER performs worse than the two baselines on instances where no additional context is required. When a stand-alone rationale is available, it is apparently easier to use it and ignore the surrounding context.

Fact-verification systems approach human performance in the "abstract-provided" setting We assign 151 claim-evidence pairs from SCIFACT for independent annotation by two different annotators. We obtain an estimate of "human-level" performance by treating the first annotator's results as "gold", and the second annotator's results as predictions. The results are shown in Table 6. Existing systems already exceed human agreement for sentence-level evaluation, but not abstract-level, indicating that experts tend to agree on the overall relationship between claim and abstract, but may disagree about exactly which sentences contain the best evidence. This fact constitutes another reason not to rely solely on selected rationales when predicting a fact-checking label: the choice of rationales is itself somewhat subjective.

In addition, these results suggest that one key subtask of scientific claim verification – namely, predicting whether an evidence-containing sentence or short document SUPPORTS or REFUTES a claim – may be nearly "solved" in the setting where (1) the claims are atomic, and (2) roughly 1,000 in-domain labeled claims are available for training.

7 Related work

Related work on scientific claim verification was covered in §2. We briefly discuss some other relevant work. The idea of multi-task label prediction and rationale selection for semi-supervised rationale selection, similar in spirit to LONGCHECKER, was proposed by Pruthi et al. (2020) and applied to sentiment analysis and propaganda detection tasks. A different alternative to supervised rationale selection is to treat rationales as latent variables, as in Lei et al. (2016); Paranjape et al. (2020).

	Abstract			Sentence		
	P	R	F1	P	R	F1
VERT5ERINI	90.7	74.3	81.7	79.6	62.2	69.8
PARAGRAPHJOINT	87.2	64.4	74.1	76.7	55.1	64.1
LONGCHECKER	87.4	75.2	80.9	80.5	70.3	**75.0**
Human	94.8	84.1	**89.1**	67.4	67.4	67.4

Table 6: Performance on SCIFACT in the "abstract-provided" setting. Models exceed human agreement as measured by sentence-level F1, but not abstract-level.

Long-document encodings for fact verification have been explored by Stammbach (2021), who use Big Bird Zaheer et al. (2020) for full-document evidence extraction from FEVER. Domain adapation for scientific text has been studied in a number of works, including Gururangan et al. (2020); Beltagy et al. (2019); Lee et al. (2020); Gu et al. (2021). In those works, the primary focus is on lagnuage model pretraining. Here, we focus on training on the target task using out-of-domain and weakly-supervised data.

8 Conclusion and future work

In this work, we addressed two weaknesses of existing scientific claim verification systems: modeling abstract-level context, and leveraging weakly-labeled in-domain data for domain adaptation. We developed a modeling framework and weak supervision approach which led to state-of-the-art performance on three datasets, in both the zero/few-shot and fully-supervised setting, and conducted analysis to characterize the source of these improvements.

This work points toward a number of promising future directions for scientific claim verification. These include further research on few-shot domain adaptation, characterization of the performance of fact-checking models when verifying claims against realistic-sized corpora of millions of documents, and extending the approach developed here to contexts beyond scientific research abstracts. Another interesting alternative to the approach taken here would be to explicitly "decontextualize" evidence-containing rationales by filling in missing context, and then make pipeline-style label predictions based on the decontextualized evidence. The reliance of the label predictor on a small collection of decontextualized sentences could lead to the model being more easily interpretable.

9 Ethical considerations and broader impact

One long-term goal of research on scientific claim verification is to build systems that can automatically identify mis- and dis-information, which we believe would be socially beneficial given the current prevalence of mis- and dis-information online.

In the shorter term, this work presents two potential risks. First, automated systems for scientific fact-checking are not mature enough to inform real-world medical decisions. We will include a disclaimer with released software to this effect. Second, bad actors could potentially use this work to develop models trained to "fool" fact-checking systems. While this risk cannot be ruled out, we believe that the benefits of publishing this work outweight the risks that it will be used by malicious actors.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP*.

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

Daniel Fernando Campos, T. Nguyen, M. Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L. Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *NeurIPS*.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *TACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Jay DeYoung, E. Lehman, B. Nye, I. Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*.

T. Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. In *Tackling Climate Change with ML workshop @ NeurIPS*.

Yuxian Gu, Robert Tinn, Hao Cheng, Michael R. Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *ArXiv*, abs/2004.10964.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*.

Neema Kotonya and F. Toni. 2020. Explainable automated fact-checking for public health claims. In *EMNLP*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Nayeon Lee, Yejin Bang, Andrea Madotto, Madian Khabsa, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. In *NAACL*.

Eric P. Lehman, Jay DeYoung, R. Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *NAACL*.

Tao Lei, R. Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. In *EMNLP*.

Xiangci Li, G. Burns, and Nanyun Peng. 2021. A paragraph-level multi-task learning model for scientific fact-verification. In *Workshop on Scientific Document Understanding @ AAAI*.

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*, abs/1907.11692.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Michael Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *ACL*.

Salman Bin Naeem, Rubina Bhatti, and Aqsa Painda Khan. 2020. An exploration of how fake news is taking over social media and putting public health at risk. *Health Information and Libraries Journal*.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *EMNLP*.

Benjamin E. Nye, Jay DeYoung, E. Lehman, A. Nenkova, I. Marshall, and Byron C. Wallace. 2020. Understanding clinical trial reports: Extracting medical entities and their relations. *ArXiv*, abs/2010.03550.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *EMNLP*.

Gordon Pennycook, Jonathon McPhetres, Yunhao Zhang, Jackson G. Lu, and David G. Rand. 2020. Fighting covid-19 misinformation on social media: Experimental evidence for a scalable accuracy nudge intervention. *Psychol Sci*, 31.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with vert5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis @EACL*.

Danish Pruthi, Bhuwan Dhingra, Graham Neubig, and Zachary Chase Lipton. 2020. Weakly- and semi-supervised evidence extraction. *ArXiv*, abs/2011.01459.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

S. Robertson and H. Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*.

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

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *EMNLP*.

Dominik Stammbach. 2021. Evidence selection as a token-level prediction task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER) @ EMNLP*.

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

Ellen M. Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2020. Trec-covid: Constructing a pandemic information retrieval test collection. In *SIGIR*.

David Wadden and Kyle Lo. 2021. Overview and insights from the sciver shared task on scientific claim verification. In *Proceedings of the Second Workshop on Scholarly Document Processing @ NAACL*.

David Wadden, Kyle Lo, Lucy Lu Wang, Shanchuan Lin, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *EMNLP*.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Michael Kinney, Ziyang Liu, William Cooper Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D Wade, Kuansan Wang, Christopher Wilhelm, Boya Xie, Douglas A. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. Cord-19: The covid-19 open research dataset. *ArXiv*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *NeurIPS*.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021. Abstract, rationale, stance: A joint model for scientific claim verification. In *EMNLP*.

A Data processing and statistics

A.1 Data preprocessing

SciFact We use SciFact in its original form, as it was released by the paper authors (Wadden et al., 2020).

HealthVer The HealthVer (Sarrouti et al., 2021) data release available at https://github.com/sarrouti/HealthVer appears in NLI format, pairing claims with evidence-containing sentences; the documents from which the sentences were extracted are not provided. We match evidence-containing sentences to their abstracts in the CORD-19 corpus (Wang et al., 2020) using a simple substring search, after normalizing for capitalization and whitespace differences. Evidence for which no match was found in the corpus is discarded.

We then segment the abstracts into sentences. Any sentence in the abstract with a string overlap of $> 50\%$ with the evidence provided in the original data is marked as a rationale. A small number of claims in HealthVer had both supporting and refuting evidence in the same abstract; we remove these claims as well to conform to our task definition. Modeling conflicting evidence is a promising direction for future work.

COVID-Fact The COVID-Fact data available at https://github.com/asaakyan/covidfact is released in a similar format to HealthVer. Like HealthVer, we perform string search over CORD-19 to identify the abstracts containing evidence, and use the same procedure for assigning rationale labels to sentences from the abstract. COVID-Fact also includes evidence from sources scraped from the web that are not contained in CORD-19, such as news articles. These sources are not provided with the data release; we discard evidence from non-CORD-19 sources[6].

Refuted claims in COVID-Fact are generated automatically by replacing words in the original claim. Based on a manual inspection, we found this process to generate a truly refuted claim roughly a third of the time; in most other cases, it generated a claim that was either ungrammatical or for which the provided evidence was irrelevant. A few cases are provided in Table 7.

[6]Upon request, the paper authors did kindly provide us with scraped evidence documents. Unfortunately, we did not have time to re-run our experiments on these additional sources.

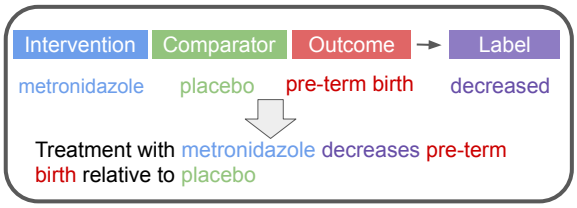

Figure 2: An example showing how an evidence inference prompt (top) can be converted into a claim (bottom) using templates. A refuted claim could be generated by substituting "increases" for "decreases" in the prompt text.

FEVER We use the Fever dataset as-is.

EvidenceInference The EvidenceInference dataset consists of "ICO" (intervention / comparator / outcome) prompts, paired with labels indicating whether the intervention leads to an increase, decrease, or no change in the outcome with respect to the comparator. We use templates to convert these prompts to claims. Figure 2 for an example. Rationale annotations are provided for this dataset. Additional examples of templates are below; the full list will be included in the code release. Refuted claims are generated by swapping "increase" and "decrease" templates.

- **Increase**: [intervention] raises [outcome] relative to [comparator]
- **No change**: [intervention] and [comparator] have very similar effects on [outcome]
- **Decrease**: [intervention] results in a decrase in [outcome], relative to [comparator]

PubMedQA We use the PQA-A subset released at https://pubmedqa.github.io/, which is filtered for "claim-like" titles. We generate negations by identifying titles with the phrases "does not", "do not", "are not", "is not". "Does not" and "do not" are removed and the relevant verbs are modified to have the correct inflection; for instance "smoking does not cause cancer" is convered to "smoking causes cancer". Similarly, "are not" and "is not" are replaced by "are" and "is".

To generate rationales needed to train pipeline models on PubMedQA, we follow the following procedure. First, we encode the claim and all abstract sentences using the `all-MiniLM-L6-v2` model from the Sentence-Transformers package https://www.sbert.net/. Then, we rank abstract sentences by cosine similarity with the claim and label the top-k sentences as rationales, where

Original claim	Automatic negation	Comment
Sars-cov-2 reactive t cells . . . are likely expanded by beta-coronaviruses	Sars-cov-2 reactive t cells . . . are not expanded by beta-coronaviruses	Successful negation
Regn-cov2 antibody cocktail prevents and treats sars-cov-2 . . .	On-cov2 antibody cocktail prevents and treats sars-cov-2 infection . . .	Ungrammatical; "On-cov2" isn't a real thing.
. . . immunity is maintained at 6 months following primary infection	. . . immunity is maintained at 6 weeks following primary infection	Not refuted; The original claim entails the negation. Immunity at 6 months implies immunity at 6 weeks.

Table 7: Automatic negations from COVID-Fact. Some are successful, in the sense that the attempted negation contradicts the original claim. Others are either ungrammatical or are entailed by the original claim.

Fold	Dataset	SUPPORTS	NEI	REFUTES
Train	SCIFACT	508	485	265
	COVID-Fact	299	-	641
	HealthVer	2384	2384	1464
Eval	SCIFACT	113	127	109
	COVID-Fact	102	-	215
	HealthVer	374	304	225

Table 8: Evidence distribution by dataset.

k is randomly sampled from $\{1, 2, 3\}$ with a 4:2:1 frequency ratio (this matches the distribution of k in SCIFACT).

A.2 Dataset statistics

Table 8 provides counts showing the number of claim / evidence pairs with each label (SUPPORTS, REFUTES, NEI), in each of our target datasets. Note that a given claim may be (and often is) paired with more than one abstract containing evidence. HealthVer is the largest datset. COVID-Fact is the smallest, in part due to the aggressive evidence filtering described in §A.1.

A.3 Examples of "context-dependent rationales"

Table 9 provides two examples of cases where abstract-level context is required to understand the relationship between a claim and a rationale reporting a relevant finding.

A.4 Annotators

In §6, we report an analysis based on annotations performed on the SCIFACT dataset. These annotations were performed by students and / or professional annotators associated with the authors' research institutions. Annotators were paid between $15 and $20 / hour.

B Modeling details

B.1 Implementation

We implement LONGCHECKER using PyTorch Lightning (https://www.pytorchlightning.ai/), which relies on PyTorch (https://pytorch.org/).

B.2 Model training

Pretraining For pretraining, we train for 3 epochs on FEVER, EVIDENCEINFERENCE, and PUBMEDQA, with the data randomly shuffled. We train on 4 negative samples (i.e. abstracts containing no evidence) per claim, which we find improves precision. We train on 8 NVIDIA RTX 6000 GPUs with a batch size of 1 / gpu (effective batch size of 8), using a learning rate of $1e-5$, using the PyTorch Lighthing implementation of the AdamW optimizer with default settings. We initialize from a Longformer-large checkpoint pretrained on the S2ORC corpus (Lo et al., 2020).

Finetuning For finetuning, we train for 20 epochs on the target dataset (SCIFACT, HealthVer, or COVID-Fact). For SCIFACT, we train on 20 negative samples claim. To create "hard" negatives – i.e. abstracts that have high lexical overlap with the claim – we create a search index from 500K abstracts randomly selected from the biomedical subset of the S2ORC corpus. For each claim, we obtain negative abstracts by using the VERT5ERINI retrieval system from §3.1 to retrieve the top-1000 most-similar abstracts from this index, removing abstracts that are annotated as containing evidence, and randomly sampling 20 abstracts to be used as negatives during training.

Since HealthVer and COVID-Fact do not have a retrieval step, they do not require negative sampling, and we train on the original datasets as-is.

Retrieval For SCIFACT, we performed dev set experiments retrieving 10, 20, or 50 abstracts /

Category	Example	
Context (Acronym)	**Claim:**	Hematopoietic stem cells segregate their chromosomes randomly.
	Context:	*we tested these hypotheses in hematopoietic stem cells (HSCs). . .*
	Evidence:	*. . . indicated that all HSCs segregate their chromosomes randomly.*
	Explanation:	HSCs is an acronym for Hematopoietic stem cells.
Context (Coreference)	**Claim:**	Errors in peripheral IV drug administration are most common during bolus administration
	Context:	*OBJECTIVES: To determine the incidence of errors in the administration of intravenous drugs . . .*
	Evidence:	*. . . Most errors occurred when giving bolus doses*
	Explanation:	The evidentiary sentence reporting the finding does not specify the type of error.

Table 9: Examples from the SCIFACT dataset of instances where context from the abstract is required to correctly interpret the rationale.

claim, and found that 10 was the best. We use that in our final experiments.

B.3 Model hyperparameters

No organized hyperparameter search was performed. We consulted with the authors of the Longformer paper for suggestions about good model parameters, and generally followed their suggestions.

The loss function in Section 3.1 requires a weight $\lambda_{\text{rationale}}$. This is set to 15 for all final experiments. We informally experimented with values of 1, 5, and 15; no organized hyperparameter search was performed. We selected the learning rate from the values $[9e-5, 5e-5, 1e-5]$.

We performed all experiments with the same random seed, 76, used by invoking the `seed_everything` function in PyTorch Lightning.

All reported results are from a single model run.

B.4 Baseline training

VERT5ERINI For SCIFACT, we use the checkpoint available at https://github.com/castorini/pygaggle/tree/master/experiments/vert5erini. For COVID-Fact and HealthVer, we follow the instructions in that repository to convert the data to the required format, and train using the available training code as-is, beginning from the available SCIFACT checkpoint. We were unable to get the code to run on GPU; we used a Google Cloud TPU for training and inference.

PARAGRAPHJOINT We use the code available at https://github.com/jacklxc/ParagraphJointModel. For predictions on SCIFACT, we make predictions using the publicly available checkpoint. For the other two target datasets, we use the training code in the repo without modification.

We used PARAGRAPHJOINT as our baseline for zero/few-shot learning experiments, and hence also performed pretraining on PARAGRAPHJOINT. The repository provides code to train on the FEVER dataset, which we used for pretraining with EVIDENCEINFERENCE and PUBMEDQA added to the data.

Domain adaptation results Table 3 shows the results of pretraining experiments performed on LONGCHECKER and PARAGRAPHJOINT. Running this experiment for VERT5ERINI would have involved training T5-3B on large datasets using Google Cloud TPU's. Given the compute required and the comparable performance of PARAGRAPHJOINT, we decided not to run this experiment.

C Additional experimental results

We report additional results not found in the main paper.

C.1 Cross-dataset generalization

In §4, we discussed how the available scientific fact-checking datasets differ in a number of important respects. Here, we explore whether models trained on one system are able to generalize to another despite these differences. We train LONGCHECKER on each of our three datasets and then evaluate its performance on the other two. We also train a version of LONGCHECKER on all three datasets together, and evaluate on each one. Since COVID-Fact has no NEI instances, during evaluation we remove all NEI instances from the other two datasets, and provide the model with evidence-containing abstracts (rather than requiring it to retrieve them).

The results are shown in Table 10. The sentence-level evaluation results (Table 10b) indicate that none of the datasets generalize well to each other

Eval →	HealthVer		COVID-Fact		SciFact	
Train ↓	F1	Δ	F1	Δ	F1	Δ
HealthVer	86.1	0.0	50.2	-24.0	73.4	-15.8
COVID-Fact	50.6	-35.6	74.1	0.0	76.1	-13.1
SciFact	70.5	-15.7	54.6	-19.6	89.2	0.0
Combined	83.0	-3.2	64.3	-9.8	87.8	-1.3

(a) Abstract-level evaluation. SciFact and HealthVer generalize fairly well to each other. COVID-Fact generalizes well to SciFact, but not HealthVer.

Eval →	HealthVer		COVID-Fact		SciFact	
Train ↓	F1	Δ	F1	Δ	F1	Δ
HealthVer	74.2	0.0	28.0	-12.6	39.7	-32.4
COVID-Fact	14.6	-59.5	40.6	0.0	41.6	-30.6
SciFact	20.5	-53.7	33.9	-6.7	72.1	0.0
Combined	71.4	-2.8	39.8	-0.9	70.5	-1.6

(b) Sentence-level evaluation. None of the datasets generalize particularly well to each other. HealthVer generalizes better to SciFact than vice versa.

Table 10: The rows and columns indicate the training and evaluation datasets, respectively. The δ values indicate the loss in performance from evaluating on a dataset different from the one the model was trained on. The "Combined" row indicates training on all datasets combined.

in their ability to identify rationales. The situation is better for abstract labeling (Table 10a). SciFact and HealthVer each generalize reasonably well to each other, but not to COVID-Fact. COVID-Fact generalizes well to SciFact, but not to HealthVer. In general, SciFact appears the "easiest" dataset to generalize to; this could be explained by the fact that SciFact claims were written to be atomic and therefore simple to verify.

Finally, a model trained on all datasets combined manages to achieve reasonable performance across all three datasets, though falling short of the performance of models trained specifically for each individual dataset.

C.2 Negative sampling

In §4.3 we described how, for SciFact, we trained on 20 negative abstracts per claim. The effect of training on these additional negative samples is shown in Figure 11. In the oracle abstract setting, negative sampling is not very beneficial. However, when the model must select evidence from retrieved abstracts, precision drops off dramatically without negative sampling. This is worth noting since it suggests that performance reported when models are provided with "gold" candidate abstracts may not offer an accurate estimate of the accuracy these systems would achieve when deployed in a real-world setting, which could require systems to verify claims over hundreds of thousands of documents.

Retrieval	Neg. sample	Abstract			Sentence		
		P	R	F1	P	R	F1
Oracle	✗	81.9	85.6	**83.7**	69.5	69.7	69.6
	✓	85.2	75.2	79.9	79.0	70.3	**74.4**
Open	✗	38.9	80.6	52.5	35.4	65.1	45.9
	✓	73.8	71.2	**72.5**	67.4	67.0	**67.2**

Table 11: Effect of negative sampling on SciFact.