MultiVerS: Improving scientific claim verification with weak supervision and full-document context

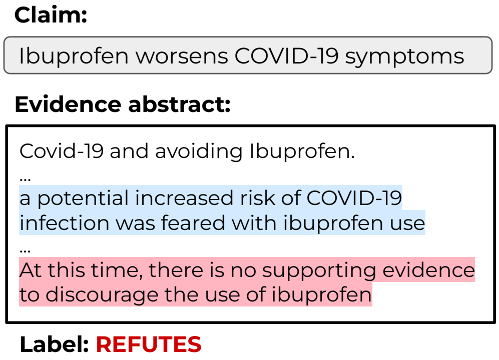
David Wadden[†] **Kyle Lo**[‡] **Lucy Lu Wang**[‡]
Arman Cohan[‡] **Iz Beltagy**[‡] **Hannaneh Hajishirzi**[†‡]
[†] University of Washington, Seattle, WA, USA
[‡] Allen Institute for Artificial Intelligence, Seattle, WA, USA
{dwadden,hannaneh}@cs.washington.edu
{kylel,lucyw,armanc,beltagy}@allenai.org

Abstract

The scientific claim verification task requires an NLP system to label scientific documents which Support or Refute an input claim, and to select evidentiary sentences (or *rationales*) justifying each predicted label. In this work, we present MultiVerS, which predicts a fact-checking label and identifies rationales in a multitask fashion based on a shared encoding of the claim and full document context. This approach accomplishes two key modeling goals. First, it ensures that all relevant contextual information is incorporated into each labeling decision. Second, it enables the model to learn from instances annotated with a document-level fact-checking label, but lacking sentence-level rationales. This allows MultiVerS to perform weakly-supervised domain adaptation by training on scientific documents labeled using high-precision heuristics. Our approach outperforms two competitive baselines on three scientific claim verification datasets, with particularly strong performance in zero / few-shot domain adaptation experiments. Our code and data are available at https://github.com/dwadden/multivers.

1 Introduction

The proliferation of scientific mis- and disinformation on the web has motivated the release of a number of new datasets (Saakyan et al., 2021; Sarrouti et al., 2021; Wadden et al., 2020; Kotonya and Toni, 2020) and the development of modeling approaches (Pradeep et al., 2021; Li et al., 2021; Zhang et al., 2021) for the task of *scientific claim verification*. The goal of the task is to verify a given scientific claim by labeling scientific research abstracts which Support or Refute the claim, and to select evidentiary sentences (or *rationales*) reporting the findings which justify each label.



Claim:

Ibuprofen worsens COVID-19 symptoms

Evidence abstract:

Covid-19 and avoiding Ibuprofen.
...
a potential increased risk of COVID-19 infection was feared with ibuprofen use
...
At this time, there is no supporting evidence to discourage the use of ibuprofen

Label: REFUTES

Figure 1: A claim from the HealthVer dataset, refuted by a research abstract. The sentence in red is a *rationale* reporting 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 treatment for COVID. MultiVerS incorporates the full context of the evidence-containing abstract when predicting fact-checking labels.

A common approach to this task is to first extract rationales from the larger document context, and then make label predictions conditioned on the selected rationales. This "extract-then-label" approach has two important drawbacks, which we aim to address in this work. First, the rationales may lack information required to make a prediction when taken out-of-context; for instance, they may contain acronyms or unresolved coreferences, or lack qualifiers that specify the scope of a reported finding (Figure 1 provides an example). Second, the "extract-then-label" approach requires training data annotated with both sentence-level rationales and abstract-level labels. While sentence-level rationale annotations are costly and require trained experts, abstract-level labels can be created cheaply using high-precision heuristics, e.g., the titles of research papers sometimes make claims that are supported by their abstracts.

Motivated by these challenges, we introduce MultiVerS (**Multi**task **Ver**ification for **S**cience): Given a claim and evidence-containing scientific abstract, MultiVerS creates a shared encoding of the entire claim / abstract context, using the Longformer encoder (Beltagy et al., 2020) to accommodate long sequences. Then, it predicts an abstract-level fact-checking label and sentence-level rationales in a multitask fashion, enforcing consistency between the outputs of the two tasks during decoding. This modeling approach ensures that label predictions are made based on all available context, and enables training on instances derived via weak supervision for which abstract-level labels are available, but sentence-level rationales are not.

In experiments on three scientific claim verification datasets, we find that MULTIVERS outperforms two state-of-the-art baselines, one of which has more than 10x the parameters of our system. In addition, we show that training MULTIVERS on weakly-labeled in-domain data substantially improves performance in the zero / few-shot domain adaptation settings. The ability to achieve reasonable performance given limited labeled data is especially valuable in specialized domains, due to the high cost of collecting expert annotations.

In summary, our contributions are as follows:

1. We introduce MULTIVERS, a multitask system for full-context scientific claim verification. MULTIVERS improves fully-supervised fact-verification performance by an average of 11% on three datasets over two state-of-the-art baselines, with improvements of 14% and 26% in the few-shot and zero-shot settings.
2. We present weak supervision heuristics to assign fact-checking labels to two large scientific datasets, and show that training on these annotations more than doubles zero-shot domain adaptation performance.
3. Through ablations and analysis, we demonstrate that our multitask modeling approach achieves our goals of incorporating full-document context into label predictions, and facilitating zero / few-shot domain adaptation.

2 Background

2.1 The scientific claim verification task

We use the definition of scientific claim verification from the SCIFACT task (Wadden et al., 2020), and provide a brief overview of the task here. Other works have cast scientific claim verification as a sentence-level natural language inference (NLI) task; in §4.1, 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 these situations 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 or 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 metric rewards precision in identifying exactly which sentences contain the evidence justifying the label. In this work, we refer to these two metrics as "abstract" and "sentence" evaluation respectively.

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 *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*

[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) and DeYoung et al. (2020).

setting, models may train on 45 claims from the target dataset. In the *fully-supervised* setting, models may train on all claims from the target dataset.

While most existing work on scientific fact-checking has focused on the fully-supervised setting, some recent work has examined the zero-shot setting. Lee et al. (2021) use language model perplexity as a measure of claim veracity. Wright et al. (2022) generate claims based on citation sentences, and verify each generated claim against the abstracts mentioned in the claim's source citation. Given the high potential impact of fact verification systems for specialized domains, combined with the substantial cost of creating these datasets, we believe that the development of techniques for zero / few-shot domain adaptation represents an important area for continued research.

2.2 Scientific claim verification datasets

Several scientific claim verification datasets have been released in the past few years. COVIDFact (Saakyan et al., 2021) and HealthVer (Sarrouti et al., 2021) verify COVID-19 claims 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) verifies claims about climate change against Wikipedia. In this work, our focus is verifying claims against scientific literature. We therefore perform experiments on the COVIDFact, HealthVer, and SciFact datasets. Preprocessing details and summary statistics for 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. More recently, ARSJoint (Zhang et al., 2021) achieved performance competitive with these two systems.[4]

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 $\widehat{y}(c, f_R(\widehat{R}(c, a))$ based on the claim and predicted rationales, where f_R is a function which creates a representation of the predicted rationales.

While existing models share this general approach, they use different functions f_R to construct rationale representations. For Vert5Erini, rationale selection and label prediction are performed by two separate T5-3B models, and f_R concatenates the text of the selected rationales. As a result, the label predictor may not have access to all context needed to make a correct label prediction. ParagraphJoint and ARSJoint attempt to address this issue by encoding the claim and full abstract (truncating to 512 tokens), and using these representations as the basis for both rationale selection and label prediction. The function f_R consists of self-attention layers over the (globally-contextualized) token representations of the predicted rationales. Thus, ParagraphJoint and ARSJoint can incorporate abstract-level context into label decisions. However, the mechanism by which this occurs is more complex than for our proposed system and requires rationale supervision for all training instances.

3 The MultiVerS model

We propose the MultiVerS model for full-context claim verification. In §3.1, we describe our modeling approach. Rather than predicting rationales $\widehat{R}(c, a)$ followed by the overall fact-checking label $\widehat{y}(c, f_R(\widehat{R}(c, a)))$, we predict $\widehat{y}(c, a)$ directly based on an encoding of the entire claim and abstract, and enforce consistency of $\widehat{R}(c, a)$ with $\widehat{y}(c, a)$ during decoding. A similar idea has been shown to be effective on sentiment analysis and propaganda detection with token-level rationales (Pruthi et al., 2020). In §3.2, we explain how our 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 $\texttt{</s>}_i$:

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

The 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

[4] Recent progress can be found on the SciFact leaderboard.

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.

Multitask 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 feedforward layers followed by a two-way softmax, on top of the globally-contextualized token `</s>`$_i$.

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.

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

$$L = L_{\text{label}} + \lambda_{\text{rationale}} L_{\text{rationale}} \tag{1}$$

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

At inference time, we first predict $\widehat{y}(c, a)$ to be the label with the highest softmax score. If the predicted label is NEI, we predict no rationales. If the predicted label is either SUPPORTS or REFUTES, then we predict rationales as all sentences with an assigned softmax score of greater than 0.5. If no sentences have a rationale softmax over 0.5, then we predict the highest-scoring sentence as the sole rationale. In §6.2, we show that this ability to condition the rationale predictions on the label prediction (as opposed to conditioning the label on the predicted rationales) leads to substantial improvement in the zero-shot domain adaptation setting.

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 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 §6.1). (3) Scientific documents – either unlabeled or labeled for different tasks – are abundant, and high precision heuristics (described in §4.2) can be used to generate document-level fact-checking labels $y(c, a)$ for these data.

We train MULTIVERS as follows: we first pretrain on a combination of general-domain fact-checking annotations, combined with weakly-labeled in-domain data.[5] Then, we finetune on the target scientific fact-checking dataset. The multitask architecture of MULTIVERS is well-suited to this strategy, since the model can be trained on data with or without rationale annotations. When no rationales are available, we set $\lambda_{\text{rationale}} = 0$ in the loss function and train as usual. By contrast, training an "extract-then-label" model on weakly-supervised data requires creating rationale annotations $R(c, a)$, which is quite noisy (see §4.2).

4 Datasets

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 can be found in Appendix A. HealthVer and COVIDFact 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 (Wang et al., 2020) containing these sentences.

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 in SCIFACT (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.

[5] We use "pretraining" as shorthand for "training on the target task with out-of-domain and/or weakly-supervised labels."

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%
COVIDFact	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. Datasets 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.

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), verified 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 justify a Supports / Refutes verdict and are labeled NEI.

COVIDFact (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 COVIDFact 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 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 models requiring rationale supervision, 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). These annotations are not used to train MultiVerS. To estimate the precision of our 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, COVIDFact, and HealthVer respectively.

5 Experimental setup

We describe our model training procedure, the systems against we compare MultiVerS, and our ablation experiments.

5.1 Model training

Our complete training procedure consists of pretraining on the three datasets from §4.2, followed by finetuning on one of the target datasets from

§4.1. We conduct experiments with three different levels of supervision. For *zero-shot* experiments, we perform pretraining only. For *few-shot* experiments, we pretrain followed by finetuning on 45 target examples. For *fully-supervised* experiments, we pretrain and then train on all target data.

Following Li et al. (2021), we found that negative sampling was important to achieve good precision on SciFact, which requires document retrieval. We train with 20 negative samples per claim and retrieve 10 abstracts per claim at inference time. Appendix C.3 shows results without negative sampling. For the other datasets, no negative sampling was used. Additional details including batch sizes, learning rates, number of epochs, etc. can be found in Appendix B.

During model development, we experimented with training on all three target datasets combined before predicting on each one, but found that this did not improve performance; see Appendix C.4.

5.2 Baseline systems

We use ParagraphJoint and Vert5Erini as baselines. Vert5Erini is the largest model, with 5.6B parameters. MultiVerS and ParagraphJoint are comparably-sized, with 440M and 360M parameters, respectively.

In the fully-supervised setting, we compare against both baselines. For prediction on SciFact, we use publicly available model checkpoints as-is. For training on HealthVer and COVIDFact, we use the code provided by the authors, starting from the available checkpoints trained on SciFact. Model hyperparameters (learning rate, batch size, epoch number, etc.) for all systems including MultiVerS were tuned based solely on SciFact and not adjusted further. Additional details can be found in Appendix B.4.

Evaluation in the few-shot and zero-shot settings requires pretraining and finetuning as described in §5.1. Due to the expense of pretraining T5-3B, we do not perform these experiments for Vert5Erini, and compare only against ParagraphJoint (which shows comparable performance in the fully-supervised setting). We pretrain ParagraphJoint on the data described in §4.2.

5.3 Ablations

Since ParagraphJoint and Vert5Erini differ from MultiVerS along a number of important dimensions (e.g. model architecture, number of parameters, and base encoder), we conduct ablations to characterize the performance contributions of three key components of MultiVerS.

Pretraining data We compare the results of three different pretraining strategies. For FeverSci, we pretrain on all available data as described in §5.1. For Fever, we pretrain on Fever only. For *No-Pretrain*, we perform no pretraining.

Base encoder We compare the performance achieved using LongFormer as the encoder for MultiVerS, compared to the results when we swap in RoBERTa but keep other settings identical. We use Longformer-large and RoBERTa-large.

Modeling approach We compare three modeling approaches: (1) The *Multitask* approach is the method used by MultiVerS as described in §3.1. (2) The *Pipeline* approach consists of two separate Longformer modules. The first selects rationales as described in §3.1, but with L_{label} removed from Eq. 1, and the second module predicts a label given the text of the rationales selected by the first module as input. When pretraining on PubmedQA, we train on the rationales chosen by Sentence-BERT as described in §4.2. (3) The *Multitask train / Pipeline inference* (MT / PI) approach takes the model trained using the Multitask approach, and performs inference using the Pipeline approach. Specifically, MT / PI is trained to make label predictions based on full abstracts, but must make test-time label predictions based on predicted rationales only. By contrast, the Pipeline model makes label predictions based on gold and predicted rationales at train and test time, respectively.

6 Experimental results

We compare MultiVerS performance relative to our baseline systems, and present ablation results.

6.1 Main Results

Table 2 compares the performance of MultiVerS against ParagraphJoint and Vert5Erini. A few trends are apparent. First, MultiVerS outperforms the baselines on all datasets, with relative improvements — averaged over the three datasets and two evaluation methods — of 26%, 14%, and 11% in the zero-shot, few-shot, and fully-supervised settings respectively. We examine possible causes of this improvement in §6.2. Second, while all models score within roughly six points of each other on HealthVer and SciFact, variability is much greater on COVIDFact. We suspect

		HealthVer						**COVIDFact**						**SciFact**					
		Abstract			Sentence			Abstract			Sentence			Abstract			Sentence		
Setting	**Model**	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero	ParagraphJoint	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
	MultiVerS	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	ParagraphJoint	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
	MultiVerS	63.6	47.9	**54.7**	41.9	31.0	35.7	71.3	68.1	**69.7**	39.5	35.4	**37.4**	76.4	54.1	**63.3**	51.7	40.3	**45.3**
Full	Vert5Erini	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	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
	MultiVerS	78.9	76.3	**77.6**	71.4	67.0	**69.1**	77.3	77.3	**77.3**	41.5	46.1	**43.7**	73.8	71.2	**72.5**	67.4	67.0	**67.2**

Table 2: Performance of MultiVerS and baselines. In the fully-supervised setting, we compare to ParagraphJoint and Vert5Erini, which exhibit comparable performance. In the zero and few-shot settings, we compare to ParagraphJoint only due to the high cost of pretraining Vert5Erini. We report performance using abstract-level and sentence-level evaluation as defined in §2.1.

that this is due to the automatically-generated nature of COVIDFact negations. Third, we observe that HealthVer appears to be the most challenging dataset of the three. Few-shot abstract-level F1 scores for COVIDFact and SciFact are generally within 10 F1 of their fully-supervised values, while the gap is roughly 20 F1 for HealthVer. This may be due to the high complexity of HealthVer claims.

6.2 Ablations

The results of all ablations are shown in Table 3. We report abstract and sentence-level F1 scores in the main text; full results can be found in Table 9 in Appendix C.

In-domain pretraining substantially improves zero / few-shot performance In Table 3a, we compare the performance of models pretrained on FeverSci, Fever, and No-Pretrain. In the zero-shot setting, removing scientific data during pretraining results in a relative performance decrease of 65%, averaged over the three datasets and two evaluation methods. The decrease is driven primarily by very low recall (see Table 9a).

In the few-shot setting, Fever pretraining scores within 4% of FeverSci, while No-Pretrain results in a 39% decrease relative to FeverSci. This suggests that training on a handful of target examples is sufficient to recalibrate a model trained for a different domain, but not to learn the task from scratch. In the fully-supervised setting, Fever pretraining is only slightly worse than FeverSci, while No-Pretrain lags by roughly 9%. Overall, the results indicate that pretraining always helps, and pretraining on weakly-labeled in-domain data helps especially when target data are scarce.

Longformer improves performance on datasets with long documents Table 3b compares the performance of MultiVerS when Longformer and RoBERTa are used as the base encoder. Using Longformer consistently helps on SciFact, but does not help on the other two datasets. This is unsurprising, since 27% of SciFact instances exceed the RoBERTa token limit, compared to less than 15% for the other two datasets (Table 1).

Multitask modeling improves zero / few-shot performance Results comparing our three different modeling approaches are shown in Table 3c. In the zero-shot setting, we find that Multitask performs best, with both MT / PI and Pipeline exhibiting performance drops greater than 50%. The Multitask approach of predicting rationales conditioned on the predicted label leads to improved recall (see Table 9c). Similarly, in the few-shot setting, both Pipeline and MT / PI perform roughly 10% worse than Multitask. Collectively, the results indicate that Multitask makes the best use of the available data when target annotations are limited.

We also find that MT / PI outperforms Pipeline in the zero-shot setting. This supports our intuition from §3.2 that, while training on weakly-supervised *document-level* labels improves zero-shot performance, training on weakly-supervised *sentence-level* rationales (as for Pipeline) leads to worse performance than not training on these rationales (as for MT / PI).

In the fully-supervised setting, Multitask performs best on SciFact, while Pipeline slightly outperforms Multitask on HealthVer and COVIDFact. MT / PI performs substantially worse than the other approaches on all datasets. We investigate

	Pretraining	HealthVer	COVIDFact	SciFact
Zero	FeverSci	**30.7 / 7.8**	**47.2 / 23.6**	**46.7 / 27.8**
	Fever	1.3 / 0.7	25.2 / 11.2	23.9 / 11.8
Few	FeverSci	**54.7 / 35.7**	69.7 / 37.4	**63.3 / 45.3**
	Fever	53.4 / 31.9	**74.4 / 42.1**	54.5 / 39.0
	No-Pretrain	39.4 / 27.0	67.8 / 22.6	24.2 / 10.8
Full	FeverSci	**77.6** / 69.1	77.3 / **43.7**	**72.5 / 67.2**
	Fever	77.1 / **70.3**	**77.4** / 43.3	67.9 / 61.7
	No-Pretrain	74.5 / 69.7	69.7 / 36.6	63.3 / 58.4

(a) Effect of pretraining data. In-domain pretraining is very effective in the zero- and few-shot settings. In the zero-shot setting, "No-Pretrain" metrics are not shown since this would correspond to no training at all.

	Encoder	HealthVer	COVIDFact	SciFact
Zero	Longformer	30.7 / 7.8	47.2 / 23.6	**46.7 / 27.8**
	RoBERTa	**34.2 / 9.2**	**48.3 / 26.2**	45.2 / 25.9
Few	Longformer	**54.7** / 35.7	69.7 / 37.4	**63.3 / 45.3**
	RoBERTa	51.2 / **36.9**	**72.1 / 41.0**	50.5 / 34.0
Full	Longformer	77.6 / 69.1	77.3 / **43.7**	**72.5 / 67.2**
	RoBERTa	**78.8 / 72.7**	**78.2** / 43.4	67.6 / 62.3

(b) Effect of base encoder. Longformer improves performance on SciFact, which has the largest fraction of instances exceeding the RoBERTa token limit.

	Approach	HealthVer	COVIDFact	SciFact
Zero	Multitask	**30.7 / 7.8**	**47.2 / 23.6**	**46.7 / 27.8**
	Pipe	3.2 / 0.9	19.0 / 10.5	22.5 / 12.8
	MT / PI	4.5 / 1.8	26.7 / 13.5	28.3 / 17.7
Few	Multitask	**54.7 / 35.7**	**69.7** / 37.4	**63.3 / 45.3**
	Pipe	52.8 / 29.5	68.3 / **38.2**	53.0 / 39.9
	MT / PI	46.7 / 32.3	59.3 / 34.1	56.2 / 41.1
Full	Multitask	77.6 / 69.1	77.3 / 43.7	**72.5 / 67.2**
	Pipe	**78.4 / 69.2**	**77.6 / 47.7**	70.9 / 66.2
	MT / PI	70.6 / 64.3	73.3 / 44.0	60.3 / 57.0

(c) Effect of model architecture. The Multitask approach performs best in the zero- and few-shot settings. We examine the fully-supervised setting in detail in §7.1.

Table 3: Ablations examining the effects of pretraining data, base encoder, and modeling approach. Entries are formatted "{Abstract-level F1} / {Sentence-level F1}".

	Self-contained			**Context-dependent**			
Approach	P	R	F1	P	R	F1	%**Δ**
Multitask	86.1	82.9	84.5	90.3	60.9	72.7	-14.0%
Pipeline	92.4	89.0	90.7	82.4	60.9	70.0	-22.8%
MT / PI	91.8	54.9	68.7	100.0	13.0	23.1	-66.4%
Count	82			46			

Table 4: Performance of the Multitask, Pipeline, and MT / PI modeling approaches on SciFact instances with rationales that are self-contained (can be interpreted in isolation) or context-dependent (must be interpreted in the context of the abstract). Evaluation is performed in the abstract-provided setting. We report abstract-level metrics; sentence-level results are similar. The %Δ indicates the drop in F1 score on context-dependent instances relative to self-contained instances. Multitask suffers the smallest performance loss, while MT / PI suffers the largest.

these findings further in §7.1; our results indicate that Pipeline may, in effect, be trained to make predictions based on insufficient evidence.

7 Analysis

7.1 Fully-supervised Pipeline performance

In §6.2, we found that the Pipeline approach (but not the MT / PI approach) performed on par with the Multitask approach in the fully-supervised setting. To understand this finding, we collected detailed annotations for 128 claim / evidence instances from the SciFact test set. For each instance, an annotator indicated whether the annotated rationales were "self-contained" — i.e. sufficient to justify the fact-checking label when taken in isolation, or "context-dependent" — i.e. only sufficient when taken in the context of the abstract. Figure 1 and Table 8 provide examples; see Choi et al. (2021) for a detailed discussion of different forms of context-dependence.[6]

Table 4 compares the performance of the three modeling approaches on instances with self-contained vs. context-dependent evidence. We find that all approaches have lower performance on context-dependent instances relative to self-contained instances, but the size of the performance drop varies widely. The Multitask approach performs 14.0% worse on context-dependent instances, while the Pipeline approach performs 22.8% worse. Most interestingly, MT / PI performs 66.4% worse, driven predominantly by low recall. The MT / PI model frequently (and correctly) predicts that context-dependent rationales are not sufficient to justify a Supports / Refutes decision. These findings suggest that (1) the Multitask approach is, as expected, best at verifying claims with context-dependent evidence, and (2) the Pipeline approach has, in effect, over-fit to context-dependent rationales and learned to make predictions based on insufficient evidence.

[6] Unlike Choi et al. (2021), we do not include the presence of acronyms as "context-dependent," since an acronym can be matched with its expansion based on surface-level textual features. See Appendix C.2 for further analysis of acronyms.

	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
MULTIVERS	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 5: Performance on SCIFACT in the "abstract-provided" setting. Models exceed human agreement as measured by sentence-level F1, but not abstract-level.

7.2 Performance upper bound

To determine an "upper bound" on the achievable performance of scientific fact-checking models, we assign 151 claim-evidence pairs from SCIFACT for independent annotation by two different annotators. We estimate human-level performance by treating the first annotator's results as "gold," and the second annotator's results as predictions. For comparison, we make predictions using MULTIVERS and our two baseline models, with candidate abstracts provided as input. The results are shown in Table 5. 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 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 abstract 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.

8 Related work

Background on scientific claim verification is covered in §2; we discuss other relevant work here. Nye et al. (2020) have previously observed that document-level context is often required to properly interpret scientific findings.

DeYoung et al. (2020) use an "extract-then-label" pipeline for the original EVIDENCEINFERENCE task. Multitask label prediction and rationale selection was proposed by Pruthi et al. (2020) and applied to sentiment analysis and propaganda detection. As in this work, the authors condition on the predicted label when predicting rationales. Another alternative to supervised rationale selection is to treat rationales as latent variables (Lei et al., 2016; Paranjape et al., 2020).

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 adaptation 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 language model pretraining. Here, we focus on training on the target task using out-of-domain and weakly-labeled data.

9 Conclusion

This work points to a number of promising future directions for scientific claim verification. These include applying the approach presented here to develop scientific claim verification models for new scientific sub-domains or other specialized fields given a handful of labeled examples, and extending the task definition to verify claims against longer contexts (e.g. full scientific papers) or larger corpora. Our task formulation also offers an opportunity to study the effects of rationale decontextualization (Choi et al., 2021), especially in cases where models may be making predictions based on insufficient evidence.

In presenting the MULTIVERS system, we addressed two challenges associated with scientific claim verification: incorporating relevant information beyond rationale boundaries by modeling full-document context, and facilitating zero / few-shot domain adaptation through weak supervision enabled by a multitask modeling approach. Our experiments show that MULTIVERS outperforms existing systems across several scientific claim verification datasets. We hope that the task, data, and modeling resources presented in this paper will encourage further work and progress towards the broader goals of identifying and addressing scientific mis- and disinformation.

10 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 disinformation generators trained to "fool" automated fact-checkers. While this risk cannot be ruled out, we believe that the benefits of publishing this work and making our models available to the community to facilitate further research outweigh the risks that this work will be misused by malicious actors.

Acknowledgments

This research was supported by the ONR MURI N00014-18-1-2670, ONR N00014-18-1-2826, DARPA N66001-19-2-4031, NSF (IIS 1616112), NSF Convergence Accelerator Award #2132318, an Allen Distinguished Investigator Award, and the Sloan fellowship. We thank the Semantic Scholar team at AI2, UW-NLP, and the H2lab at UW for helpful comments and feedback. Thanks to Arkadiy Saakyan and Tuhin Chakrabarty for help with COVIDFact, to Mourad Sarrouti for help with HealthVer, and to Xiangci Li and Ronak Pradeep for help with PARAGRAPHJOINT and VERT5ERINI, respectively.

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. In *ACL*.

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*.

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*.

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. In *EMNLP Findings*.

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, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating Scientific Claims for Zero-Shot Scientific Fact Checking. In *ACL*.

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.

COVIDFact The COVIDFact 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. COVIDFact 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[7].

Refuted claims in COVIDFact 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 6.

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. The dataset is available at `https://evidence-inference.ebm-nlp.com/`. We use templates to convert these prompts to claims. See 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 decrease in [outcome], relative to [comparator]

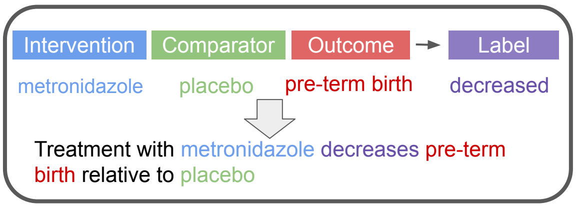


Figure 2: An example showing how an evidence inference prompt (top) can be converted into a claim (bottom) using templates. A refuted claim could be generated by substituting "increases" for "decreases" in the prompt text.

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 converted 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 employ 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 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 7 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 dataset. COVIDFact is the smallest, in part due to the aggressive evidence filtering described in §A.1.

A.3 Examples of context-dependent rationales

Table 8 provides an example of a context-dependent rationale (as defined in §7.1), as well as an example of a rationale with an undefined acronym. The latter occurs when an acronym appears in a rationale but its full expansion does not; an analysis of undefined acronyms is included in

[7] 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.

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 scientific entity.
... 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 6: Automatic negations from COVIDFact. 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
	COVIDFact	299	-	641
	HealthVer	2384	2384	1464
Eval	SCIFACT	113	127	109
	COVIDFact	102	-	215
	HealthVer	374	304	225

Table 7: Evidence distribution by dataset.

Appendix C.2. The code and data release will contain full annotations indicating which of the 128 human-annotated examples described in §7.1 are context-dependent, and which contain undefined acronyms.

A.4 Annotators

In §7, 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 MULTIVERS 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 Lightning 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 COVIDFact). 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 COVIDFact 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 / 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

Category	Example	
Context-dependent	**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.
Undefined 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.

Table 8: Examples from the SciFact dataset showcasing rationales that are context-dependent (top example), or include an undefined acronym (bottom example).

`seed_everything` function in PyTorch Lightning.

All reported results are from a single model run.

B.4 Baselines

VerT5erini For prediction on SciFact, we use the checkpoint available at https://github.com/castorini/pygaggle/tree/master/experiments/vert5erini. For COVIDFact 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 used 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.

C Additional results and analysis

C.1 Full ablation results

In Table 3, we presented F1 scores for ablations comparing pretraining data, model architecture, and encoder used. Table 9 presents the full results, including precision and recall.

C.2 Performance on rationales with undefined acronyms

In §7.1, we examined the difference in performance on instances with self-contained vs. context-dependent evidence. Here, we show the results of evaluation on instances containing an undefined acronym vs. cases without one. We find that undefined acronyms do not pose a challenge for Multitask and Pipeline, but do cause a small performance drop on MT / PI.

C.3 Negative sampling

In §5.1 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 abstract-provided 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.

C.4 Cross-dataset generalization

In §5, 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 MultiVerS on each of our three datasets and then evaluate its performance on the other two. We also train a version of MultiVerS on all three datasets together, and evaluate on each one. Since COVIDFact has no NEI instances, during evaluation we remove

		HealthVer						COVIDFact						SciFact					
		Abstract			Sentence			Abstract			Sentence			Abstract			Sentence		
	Pretraining	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero	FEVERSci	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**
	FEVER	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
Few	FEVERSci	63.6	47.9	**54.7**	41.9	31.0	**35.7**	71.3	68.1	69.7	39.5	35.4	37.4	76.4	54.1	**63.3**	51.7	40.3	**45.3**
	FEVER	56.4	50.8	53.4	34.8	29.4	31.9	74.4	74.4	**74.4**	39.3	45.3	**42.1**	72.4	43.7	54.5	48.8	32.4	39.0
	No-Pretrain	38.5	40.4	39.4	28.5	25.7	27.0	67.8	67.8	67.8	24.9	20.7	22.6	20.0	30.6	24.2	9.5	12.7	10.8
Full	FEVERSci	78.9	76.3	**77.6**	71.4	67.0	69.1	77.3	77.3	77.3	41.5	46.1	**43.7**	73.8	71.2	**72.5**	67.4	67.0	**67.2**
	FEVER	77.5	76.6	77.1	70.8	69.8	**70.3**	77.5	77.3	**77.4**	40.6	46.5	43.3	64.3	72.1	67.9	57.1	67.0	61.7
	No-Pretrain	75.0	74.0	74.5	71.8	67.8	69.7	69.7	69.7	69.7	35.3	38.1	36.6	64.9	61.7	63.3	62.7	54.6	58.4

(a) Effect of pretraining data.

		HealthVer						COVIDFact						SciFact					
		Abstract			Sentence			Abstract			Sentence			Abstract			Sentence		
	Encoder	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero	Longformer	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**
	RoBERTa	59.5	24.0	**34.2**	25.4	5.6	**9.2**	49.3	47.3	**48.3**	35.2	20.9	**26.2**	45.5	45.0	45.2	34.4	20.8	25.9
Few	Longformer	63.6	47.9	**54.7**	41.9	31.0	35.7	71.3	68.1	69.7	39.5	35.4	37.4	76.4	54.1	**63.3**	51.7	40.3	**45.3**
	RoBERTa	55.0	47.9	51.2	39.0	35.0	**36.9**	72.5	71.6	**72.1**	39.7	42.5	**41.0**	59.0	44.1	50.5	36.8	31.6	34.0
Full	Longformer	78.9	76.3	77.6	71.4	67.0	69.1	77.3	77.3	77.3	41.5	46.1	**43.7**	73.8	71.2	**72.5**	67.4	67.0	**67.2**
	RoBERTa	77.8	80.0	**78.8**	73.4	72.0	**72.7**	78.2	78.2	**78.2**	40.8	46.3	43.4	67.1	68.0	67.6	62.7	61.9	62.3

(b) Effect of base encoder.

		HealthVer						COVIDFact						SciFact					
		Abstract			Sentence			Abstract			Sentence			Abstract			Sentence		
	Approach	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero	Multitask	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**
	Pipe	58.8	1.7	3.2	29.4	0.5	0.9	67.3	11.0	19	57.4	5.8	10.5	80.6	13.1	22.5	72.2	7.0	12.8
	MT / PI	60.9	2.3	4.5	41.7	0.9	1.8	78.5	16.1	26.7	57.7	7.6	13.5	80.9	17.1	28.3	75.5	10.0	17.7
Few	Multitask	63.6	47.9	**54.7**	41.9	31.0	**35.7**	71.3	68.1	**69.7**	39.5	35.4	37.4	76.4	54.1	**63.3**	51.7	40.3	**45.3**
	Pipe	56.3	49.7	52.8	32.6	27.0	29.5	69.4	67.2	68.3	40.6	36.0	**38.2**	54.8	51.4	53.0	43.7	36.8	39.9
	MT / PI	67.0	35.9	46.7	44.5	25.3	32.3	72.6	50.2	59.3	40.2	29.7	34.1	85.3	41.9	56.2	54.7	33.0	41.1
Full	Multitask	78.9	76.3	77.6	71.4	67.0	69.1	77.3	77.3	77.3	41.5	46.1	43.7	73.8	71.2	**72.5**	67.4	67.0	**67.2**
	Pipe	78.7	78.1	**78.4**	70.2	68.3	**69.2**	79.9	75.4	**77.6**	48.2	47.2	**47.7**	68.5	73.4	70.9	64.5	68.1	66.2
	MT / PI	77.6	64.8	70.6	70.0	59.5	64.3	77.7	69.4	73.3	43.6	44.4	44.0	80.5	48.2	60.3	70.5	47.8	57.0

(c) Effect of model architecture.

Table 9: Full ablation results.

all NEI instances from the other two datasets, and evaluate in the abstract-provided setting.

The results are shown in Table 12. The sentence-level evaluation results (Table 12b) indicate that none of the datasets generalize well to each other in their ability to identify rationales. The situation is better for abstract labeling (Table 12a). SciFact and HealthVer each generalize reasonably well to each other, but not to COVIDFact. COVIDFact 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.

Approach	**No undefined acronym**			**Undefined acronym**			**%Δ**
	P	R	F1	P	R	F1	
Multitask	88.1	73.8	80.3	86.0	77.1	81.3	1.2%
Pipeline	89.9	77.5	83.2	88.6	81.2	84.8	1.9%
MT / PI	97.1	42.5	59.1	85.0	35.4	50.0	-15.4%
Count	80			48			

Table 10: Performance of different modeling approaches on instances with vs. without an undefined acronym. We perform evaluation on the same data as reported in Table 4.

Retrieval	Neg. sample	Abstract			Sentence		
		P	R	F1	P	R	F1
Abstract-provided	✗	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.

Eval →	HealthVer		COVIDFact		SciFact	
Train ↓	F1	Δ	F1	Δ	F1	Δ
HealthVer	86.1	0.0	50.2	-24.0	73.4	-15.8
COVIDFact	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. COVIDFact generalizes well to SciFact, but not HealthVer.

Eval →	HealthVer		COVIDFact		SciFact	
Train ↓	F1	Δ	F1	Δ	F1	Δ
HealthVer	74.2	0.0	28.0	-12.6	39.7	-32.4
COVIDFact	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 12: Cross-dataset generalization performance. 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.