# Prompt to be Consistent is Better than Self-Consistent? Few-Shot and Zero-Shot Fact Verification with Pre-trained Language Models

**Fengzhu Zeng** and **Wei Gao**
School of Computing and Information Systems
Singapore Management University
80 Stamford Rd, Singapore 178902
fzzeng.2020@phdcs.smu.edu.sg, weigao@smu.edu.sg

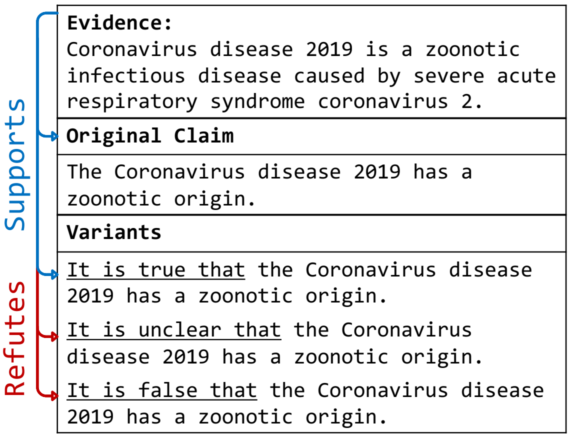
## Abstract

Few-shot or zero-shot fact verification only relies on a few or no labeled training examples. In this paper, we propose a novel method called ProToCo, to Prompt pre-trained language models (PLMs) To be Consistent, for improving the factuality assessment capability of PLMs in the few-shot and zero-shot settings. Given a claim-evidence pair, ProToCo generates multiple variants of the claim with different relations and frames a simple consistency mechanism as constraints for making compatible predictions across these variants. We update PLMs by using parameter-efficient fine-tuning (PEFT), leading to more accurate predictions in few-shot and zero-shot fact verification tasks. Our experiments on three public verification datasets show that ProToCo significantly outperforms state-of-the-art few-shot fact verification baselines. With a small number of unlabeled instances, ProToCo also outperforms the strong zero-shot learner T0 on zero-shot verification. Compared to large PLMs using in-context learning (ICL) method, ProToCo outperforms OPT-30B and the Self-Consistency-enabled OPT-6.7B model in both few- and zero-shot settings.

## 1 Introduction

The problem of misinformation has sparked significant attention on the task of fact verification within the natural language processing (NLP) community. Such task, typically represented by Fact Extraction and VERification (FEVER) benchmark (Thorne et al., 2018), requires models to verify if pieces of evidence *support*, *refute*, or contain *not enough information (NEI)* to validate a given claim.

Fully supervised fact verification has been widely studied and achieved good performance on the data of different domains (Nie et al., 2019; Ma et al., 2019; Wadden et al., 2020; Guo et al., 2022). However, collecting a large set of training data is labor-intensive, time-consuming and costly especially with the constant emergence of new events, such as COVID-19 (Lee et al., 2021; Pan et al., 2021; Saakyan et al., 2021), that may be out-of-domain. Few-shot fact verification is an urgent need but has been paid little attention because its performance is previously not competitive given very few training data (Lee et al., 2021; Zeng and Zubiaga, 2022), not to mention the zero-shot setting without any labeled data available at all.



**Evidence:**
Coronavirus disease 2019 is a zoonotic infectious disease caused by severe acute respiratory syndrome coronavirus 2.

**Original Claim**

The Coronavirus disease 2019 has a zoonotic origin.

**Variants**

It is true that the Coronavirus disease 2019 has a zoonotic origin.

It is unclear that the Coronavirus disease 2019 has a zoonotic origin.

It is false that the Coronavirus disease 2019 has a zoonotic origin.

Figure 1: An illustration of our consistency mechanism for evidence-based fact verification when the evidence supports the claim. The PLM's judgements on the variants should be logically consistent across the different variants of the claim.

In this paper, we try to improve PLMs' capability on factuality assessment for few-shot and zero-shot evidence-based fact verification. In general, consistency in fact verification dictates our assessment on the veracity of a claim based on the evidence given. For example, Figure 1 shows that given the same evidence and three major variants of the claim, the judgement of factuality on the *confirmation variant* "It is true that [claim]" should remain the same as that of the original claim, while the judgement on the *uncertainty variant* "It is unclear

*Findings of the Association for Computational Linguistics: ACL 2023*, pages 4555–4569
July 9-14, 2023 ©2023 Association for Computational Linguistics

that [claim]" and *negation variant* "It is false that [claim]" should be opposite to that of the original claim. The relations (i.e., confirmation, uncertainty and negation) between the claim and its variants naturally constrain what decisions should be made for the variants once the decision on the claim is determined, and vice versa. Such simple consistency constraints with minor adjustments can be generalized to different cases (e.g., when the evidence refutes the claim. See Section 4.3 for detail). Meanwhile, prior studies on consistency in other domains (e.g., knowledge base and question answering (QA)) have shown a strong correlation between PLM's performance and their self consistency (Elazar et al., 2021; Wang et al., 2022), but it is empirically observed that PLMs are insufficient to transfer self-consistency to downstream tasks (Ettinger, 2020; Kassner and Schütze, 2020; Kassner et al., 2021; Elazar et al., 2021). We therefore aim to explicitly impose consistency on PLMs for improving few-shot and zero-shot fact verification performance.

Inspired by the recent success of prompt-enabled PLMs on various few-shot NLP tasks via forming natural language prompts using templates (Radford et al., 2019; Brown et al., 2020; Gao et al., 2021; Liu et al., 2022a), we construct the variants of a given claim by simply altering prompt templates while keeping the claim itself unchanged. Further, we define a factuality-grounded consistency mechanism based on the aforementioned relations between the claim and its variants, and assign the labels (i.e., *support*, *refute*, and *NEI*) satisfying the consistency to the variants, so that we obtain a set of claim-evidence pairs with consistency constraints. To bring such consistency to PLMs, we then use these pairs to fine-tune T-Few (Liu et al., 2022a), a prompt-enabled PLM with a parameter-efficient fine-tuning (PEFT) method by only updating a small number of parameters. We name our method as ProToCo, Prompt PLMs To be Consistent, for improving the consistency of PLMs for few-shot and zero-shot fact verification. Our main contributions can be summarized as follows [1]:

- We design a general factuality-grounded consistency scheme to provide explicit consistency constraints for improving few-shot fact assessment, which is generalizable to zero-shot setting.
- We propose ProToCo, a novel prompt-based consistency training method for improving PLMs on few-shot and zero-shot fact verification.
- Evaluation results on three public fact verification datasets from different domains confirm that ProToCo outperforms the state-of-the-art few-shot baselines by up to 30.4% relative improvement in terms of F1, and also consistently outperforms the strong zero-shot learner T0-3B (Sanh et al., 2022) in zero-shot verification.
- When compared to large PLMs in both settings, ProToCo achieves overall better performance than OPT-30B (Zhang et al., 2022) and significantly outperforms the Self-Consistency-enabled OPT-6.7B model based on Chain-of-Thought (CoT) prompting (Wang et al., 2022).

## 2 Related Work

Existing methods tried to address few-shot fact verification by utilizing the implicit knowledge of PLMs encoded in their parameters without gradients update. Lee et al. (2021) hypothesized that the perplexity of concatenated claim-evidence text sequence evaluated by a language model could benefit claim verification, and used a few training instances to find the threshold of perplexity scores for determining the label of test claim. Zeng and Zubiaga (2022) utilizes PLMs to create a set of representative vectors for each class based on the semantic difference between claim and evidence of a few training instances, which are used to label test claims based on Euclidean distance during inference. However, these models do not update model parameters solely relying on the pre-encoded knowledge of PLMs, which cannot improve the language model itself and may not generalize well in new domains. And they also cannot perform zero-shot task as a few labeled instances are required as the anchors for labeling new instances. Our method aims to update PLMs efficiently to utilize new knowledge in a few examples and enforce model's consistency for improving both few-shot and zero-shot verification.

Recently, several studies worked to generalize PLMs to the target domain by fine-tuning the model with the full training dataset of fact verification from a different domain (Wadden et al., 2020; Saakyan et al., 2021; Schuster et al., 2021; Wadden et al., 2022). Meanwhile, some works targeted to instruct PLMs to generate task-specific training

[1] Code and dataset are available at https://github.com/znhy1024/ProToCo

data used to fully train a classifier for fact verification (Pan et al., 2021; Wright et al., 2022). Such works need a carefully crafted generation policy based on real corpus of the task, and the performance heavily depends on the quality of generated data. These approaches are considered distantly supervised, and significantly differ from ours as they do not aim to build any few-shot or zero-shot model. Unlike these studies, we assume that the language model is minimally aware of fact verification task with only a few task-specific examples, which may be even unlabeled.

In general, PLMs have shown strong few-shot learning ability in various NLP tasks (Brown et al., 2020; Sanh et al., 2022). In-context learning (ICL) uses natural language prompts or instructions to elicit desired output from PLMs without gradient updates (Radford et al., 2019; Brown et al., 2020). However, ICL is hard to deal with many prompted instances (Liu et al., 2022a), sensitive to the prompt design (Liu et al., 2022b; Lu et al., 2022) and performs worse than fine-tuning (Brown et al., 2020; Liu et al., 2022a). An alternative approach is parameter-efficient fine-tuning (PEFT) by updating only a small number of parameters to bridge the gap with standard fine-tuning (Houlsby et al., 2019; He et al., 2022; Mahabadi et al., 2021; Lester et al., 2021; Wei et al., 2022; Ben Zaken et al., 2022; Liu et al., 2022a). Our method utilizes T-few (Liu et al., 2022a), a state-of-the-art PEFT-enabled model, as our backbone to perform the factuality-grounded consistency training.

Previous works evaluate the self-consistency of PLMs by modifying the context of input sentences (Ettinger, 2020; Kassner and Schütze, 2020; Ravichander et al., 2020; Elazar et al., 2021) and empirically show that PLMs are insufficient to transfer self-consistency to downstream tasks. Some works in question answering (QA) prompt large PLMs (e.g., GPT-3 (Brown et al., 2020)) to improve QA accuracy by strengthening the consistency of predicted answers. Wang et al. (2022) prompts PLM to generate multiple explanations and candidate answers and choose the answer that consistently occurs as the prediction. The Maieutic prompting (Jung et al., 2022) designed for True-or-False commonsense QA, and ConCoRD (Mitchell et al., 2022) designed specifically based on self-consistency benchmarks, both of which elicit PLMs to generate distributions for possible candidate answers, followed by a MaxSAT solver (Battiti, 2009) to infer the most probable answer by eliminating contradictory candidates. Both methods are based on different consistency definitions from ours and may not be suitable for the fact verification task.

## 3 Problem Definition

Let $\mathbf{C} = \{(x_i, y_i)\}$ be a fact verification dataset, containing training set $\mathbf{C}_{train}$ and test set $\mathbf{C}_{test}$, where each instance consists of the input $x_i$ and ground-truth label $y_i \in \mathcal{Y}$ and $\mathcal{Y} = \{\texttt{Support, NEI, Refute}\}$. Let $x_i = (c_i, e_i)$, and the task aims to predict if the given pieces of evidence $e_i$ supports, refutes or has not enough information to validate the claim $c_i$. In the few-shot setting, we randomly sample $K$ instances *per class* from $\mathbf{C}_{train}$ for training as the class distribution is unknown. As a result, the total number of instances is $3K$ and the few-shot training set is denoted as $\mathbf{C}^{fs}_{train} = \{(x_i, y_i)\}^{3K}$. The zero-shot setting is similar but only uses $x_i$ for each instance and the *unlabeled* training set is given as $\mathbf{C}^{zs}_{train} = \{(x_i)\}^{3K}$. Note that the absence of ground-truth label makes the setting zero-shot (Wright et al., 2022; Zhou et al., 2022). Similar to previous works (Lee et al., 2021; Liu et al., 2022a), we do not assume the availability of development set as it is more realistic in a limit data scenario. Our goal is to generalise a PLM $\mathcal{M}_\theta$ to the unseen test set $\mathbf{C}_{test}$, fine-tuned only using $\mathbf{C}^{fs}_{train}$ or $\mathbf{C}^{zs}_{train}$, where $\theta$ denotes language model parameters.

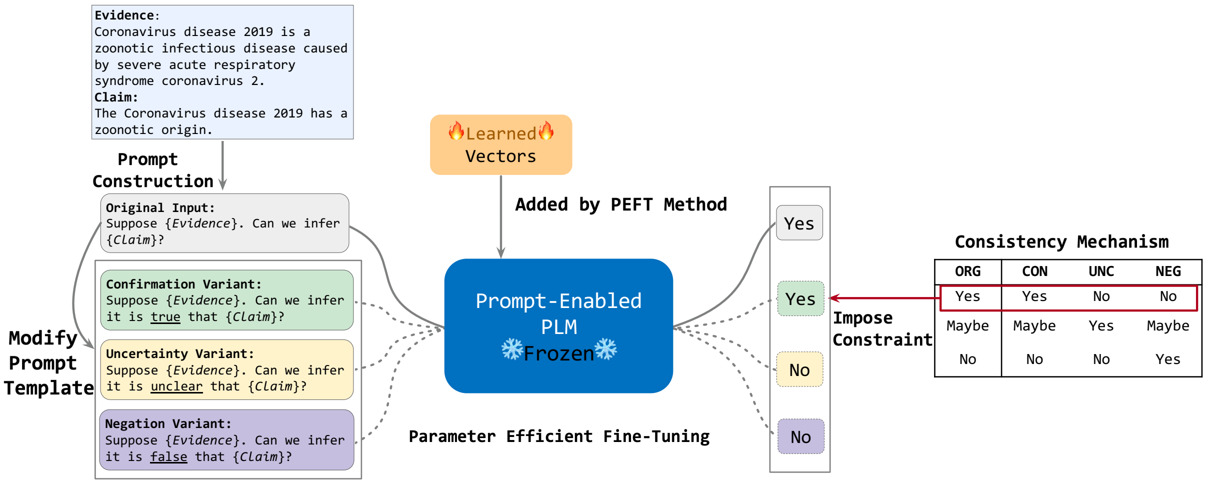
## 4 Methodology

### 4.1 Prompt Construction

Given a labelled instance $(x_i, y_i)$, the input $x_i$ (i.e., $c_i$ and $e_i$) and the label $y_i$ are firstly reformatted as a natural language input and response using a prompt template $\mathcal{T}$, which consists of an input template $\mathcal{T}_x$ and a target template $\mathcal{T}_y$. For example, as shown in Figure 2, the reformatted input $\mathcal{T}_x(x_i)$ is obtained by filling the evidence and claim in their corresponding fields:

$$\texttt{Suppose } \{e_i\}\texttt{. Can we infer } \{c_i\}\texttt{?}$$

and the reformatted label can be $\mathcal{T}_y(y_i) = \texttt{Choices}[y_i]$. Here `Choices` is a prompt-specific target words mapping containing response keys `{Yes, Maybe, No}`, where `Yes` is mapped to `Support`, `Maybe` to `NEI`, and `No` to `Refute`.



**Evidence**:
Coronavirus disease 2019 is a zoonotic infectious disease caused by severe acute respiratory syndrome coronavirus 2.
**Claim:**
The Coronavirus disease 2019 has a zoonotic origin.

**Original Input:**
Suppose {*Evidence*}. Can we infer {*Claim*}?

**Confirmation Variant:**
Suppose {*Evidence*}. Can we infer it is true that {*Claim*}?

**Uncertainty Variant:**
Suppose {*Evidence*}. Can we infer it is unclear that {*Claim*}?

**Negation Variant:**
Suppose {*Evidence*}. Can we infer it is false that {*Claim*}?

**Consistency Mechanism**

| ORG | CON | UNC | NEG |
|---|---|---|---|
| Yes | Yes | No | No |
| Maybe | Maybe | Yes | Maybe |
| No | No | No | Yes |

Figure 2: The architecture of our ProToCo model. Given a claim-evidence pair, confirmation variant (CON), uncertainty variant (UNC) and negation variant (NEG) are created by modifying the prompt template. The original input (ORG) and its variants are used to train the PLM. We use a PEFT method (i.e., $(\mathrm{IA})^3$) to train the PLM, which only updates the parameters of additionally learned vectors while other parameters are frozen. Consistency will be imposed as the constraints on PLM's predictions over the claim and its variants.

### 4.2 Inference

We take the text-to-text PLM (e.g., T5 (Raffel et al., 2020)) as $\mathcal{M}_\theta$ since the prompted input and output are text sequences. Let $\mathcal{V}$ be the vocabulary of $\mathcal{M}_\theta$. We denote each $\mathcal{T}_x(x_i)$ as an input sequence of tokens $\mathbf{x}_i$ and $\mathcal{T}_y(y_i)$ as a target sequence of tokens $\mathbf{y}_i = \{t_j \in \mathcal{V}, j \in [1, |\mathbf{y}_i|]\}$ to be generated. Then, the probability of the target sequence is $p_\theta(\mathbf{y}_i \mid \mathbf{x}_i) = \prod_{j=1}^{|\mathbf{y}_i|} p_\theta(t_j \mid \mathbf{x}_i, t_{<j})$, where $p_\theta(t_j \mid \mathbf{x}_i, t_{<j})$ is the probability of each token $t_j$ assigned by the model $\mathcal{M}_\theta$ during autoregressive generation given the input sequence $\mathbf{x}_i$ and the tokens generated prior to $t_j$. Since the sequence $\mathbf{y}_i$ corresponds to the class $y_i$, the predicted score for class $y_i$ given by $\mathcal{M}_\theta$ can be defined as the log-probability normalized by the length of output sequence to avoid possible bias towards length (Liu et al., 2022a):

$$\beta(x_i, y_i, \mathcal{T}) = \frac{1}{|\mathbf{y}_i|} \log p_\theta(\mathbf{y}_i \mid \mathbf{x}_i) \qquad (1)$$

In this way, we obtain the predicted scores of all classes using Equation 1 and use rank classification for inference by following (Liu et al., 2022a). All classes are ranked by the predicted scores and the top-ranked class is taken as the prediction.

### 4.3 The Consistency Mechanism

In this section, we describe how to establish the consistency for fact verification task. Our goal is two-fold: 1) construct a set of variants for a claim corresponding to three basic logical relations between the claim and a variant, i.e, confirmation, uncertainty, and negation; 2) the labels of variants can be unambiguously derived based on the relations above once the label of original claim-evidence pair is known. To this end, we construct the logical variants by modifying the prompt input template $\mathcal{T}_x$, as shown in Figure 2.

Specifically, we prepend "it is {$w$} that" before $c_i$ to get a claim's logical variants, where $w \in \mathcal{V}$ can be an affirmative word (e.g., true), an uncertain word (e.g., unclear), or a negative word (e.g., false), corresponding to the aforementioned relations. Figure 2 shows the consistency constraints that the model should strive to satisfy based on the set of labels assigned to the original claim and its variants given $\mathcal{T}_y(y_i)$. For example, when $\mathcal{T}_y(y_i)$ is Yes or No, the label of the confirmation variant should be same as that of the original claim since they entail each other, while the negation variant should have the opposite label because of their contradiction, and the uncertainty variant is assigned as No since the evidence indicates it is sufficient to draw a certain conclusion. Situation is slightly different when $\mathcal{T}_y(y_i)$ is Maybe since there is not enough evidence to support or refute the confirmation and negation variants, and as a result, both confirmation and negation variants are designated as Maybe while the uncertainty variant is assigned as Yes. With these consistency constraints, we could label the claim variants for each training instance and utilize them for fine-tuning the model.

### 4.4 Training Strategy

It is challenging for few-shot fine-tuning of PLMs as updating a large number of parameters with a few instances may result in unstable performance. Also, there are no labeled instances available for zero-shot fine-tuning. We will introduce how to bring our consistency mechanism into the training of PLMs in both settings.

#### 4.4.1 Parameter Efficient Fine-Tuning (PEFT)

The traditional fine-tuning methods updating all parameters of PLMs are found unstable in the few-shot setting (Zhang et al., 2021; Mosbach et al., 2021; Dodge et al., 2020), and could be computationally expensive. We thus employ PEFT methods (Houlsby et al., 2019; Mahabadi et al., 2021; Lester et al., 2021; Hu et al., 2022; Ben Zaken et al., 2022; Liu et al., 2022a) for more efficient fine-tuning.

We exploit the T-Few recipe which applies a PEFT method called (IA)$^3$ (Liu et al., 2022a) on a zero-shot learner T0 (Sanh et al., 2022) to enable its few-shot ability. The (IA)$^3$ modifies Transformer (Vaswani et al., 2017) via multiplying the keys and values in attention and the intermediate activations of position-wise feed-forward networks by the learned vectors, so that a small number of parameters are introduced for fine-tuning. And T0 has been endowed with a strong zero-shot generalizability by training a LM-adapted T5 (Lester et al., 2021) on a set of datasets covering numerous NLP tasks, where each training instance is reformatted as a natural language input and response using a prompt template.

#### 4.4.2 Loss Functions

With a few training instances, we follow Liu et al. (2022a) by combining several different loss functions to update the new parameters.

- **Standard cross-entropy loss** encourages $\mathcal{M}_\theta$ to assign higher probability $p_\theta(\mathbf{y}_i \mid \mathbf{x}_i)$ to the correct target sequence $\mathbf{y}_i$ given the input sequence $\mathbf{x}_i$:

$$\mathcal{L}_i^{\text{lm}} = -\frac{1}{|\mathbf{y}_i|} \log p_\theta(\mathbf{y}_i \mid \mathbf{x}_i) \tag{2}$$

- **Classification task loss** is based on cross-entropy. Given the predicted scores $\beta(x_i, y_i, \mathcal{T})$ assigned by PLM, the probability of predicting class $y_i$ can be calculated as:

$$q_\theta(y_i \mid x_i) = \frac{\exp\left(\beta(x_i, y_i, \mathcal{T})\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\beta(x_i, y', \mathcal{T})\right)} \tag{3}$$

and the loss for the task is:

$$\mathcal{L}_i^{\text{cls}} = -\log q_\theta(y_i \mid x_i) \tag{4}$$

- **Unlikelihood loss** forces incorrect target sequences to be assigned with lower probabilities (Welleck et al., 2020):

$$\mathcal{L}_i^{\text{ul}} = -\frac{\sum_{k=1,k\neq i}^{|\mathcal{Y}|-1} \sum_{j=1}^{|\mathbf{y}_k|} \log(1 - p_\theta(t_j \mid \mathbf{x}_i, t_{<j}))}{\sum_{k=1,k\neq i}^{|\mathcal{Y}|-1} |\mathbf{y}_k|} \tag{5}$$

The total loss for fine-tuning our backbone model T-Few is a sum of the above three losses: $\mathcal{L} = \sum_{i=1}^{3K} \mathcal{L}_i^{\text{lm}} + \mathcal{L}_i^{\text{cls}} + \mathcal{L}_i^{\text{ul}}$.

#### 4.4.3 Few-Shot and Zero-Shot Training

In the few-shot setting, we first fine-tune the model with the original labeled instances as a warm-up, and then continue the fine-tuning with the created variants and the logically consistent labels which are derived from the claim following the proposed consistency mechanism (see Section 4.3).

Given no labeled instance in the zero-shot setting, we directly fine-tune the model with the variants using the following strategy: at each training step, the prediction of the original instance by the PLM is used to assign pseudo labels to its variants based on the proposed consistency mechanism. To some extent, this training strategy provides a regulation to PLM and guide it to update the prediction on the original instance. Note that such method is still zero-shot since what is considered in training is only the determined logical relations between the claim and its variants and no ground-truth information is exploited.

## 5 Experiments and Results

### 5.1 Experimental Setup

#### 5.1.1 Datasets

We use three public fact verification datasets from different domains. Their statistics are shown in Table 1. **FEVER** (Thorne et al., 2018) provides manually crafted claims by altering factual sentences from Wikipedia. The claims are classified as `Support`, `Refute` or `NEI` by annotators. This dataset only provides gold evidence for

| Dataset | label | Supports | Refutes | NEI |
|---|---|---|---|---|
| FEVER | Train | 80,035 | 29,775 | 35,639 |
| | Test | 3,333 | 3,333 | 3,333 |
| SciFACT | Train | 332 | 173 | 304 |
| | Test | 124 | 64 | 112 |
| VitaminC | Train | 124,864 | 71,108 | 52,981 |
| | Test | 17,306 | 9,907 | 7,268 |

Table 1: Statistics of three datasets used for evaluation.

the `Support` and `Refute` classes. To provide evidence for instances in the `NEI` class, we randomly sample a sentence from Wikipedia for each claim using uniform sampling method by following Thorne et al. (2018). **SciFACT** (Wadden et al., 2020) is a fact verification dataset which consists of expert-written scientific claims by re-writing citation sentences occurring in biomedical literature. We choose the sentence from the cited abstract with the highest TF-IDF similarity to the claim for the `NEI` class following Wadden et al. (2020). **VitaminC** (Schuster et al., 2021) is a challenging dataset with cases requiring models to identify subtle factual changes. It is created by utilizing Wikipedia revisions that alter a factual statement to create claim-evidence pairs, where the instances for each revision are made contrastive, i.e., they contain evidence pairs that are nearly identical in content, but one supports the claim while the other contradicts it. VitaminC has three classes similar to FEVER and provides real or synthetic revisions. We only use instances from real revisions as the synthetic does not include the `NEI` class.

### 5.1.2 Baselines

We compare ProToCo to the following few-shot baselines: **Majority** simply assigns the most frequent class of the training set to all instances; **RoBERTa-L** (Liu et al., 2019) is a pre-trained RoBERTa-large model with a feed-forward classifier fine-tuned on top of it; **GPT2-PPL** (Lee et al., 2021) uses a few labeled instances to find the threshold of perplexity scores based on the GPT-2 language model (Radford et al., 2019) for determining claim class labels; **SEED** (Zeng and Zubiaga, 2022) utilizes PLMs to obtain semantic difference vectors between claims and their evidence and average them to create representative vectors for each class, which are used to label instances based on Euclidean distance during inference.

We also compare ProToCo to zero-shot baselines: **T0** (Sanh et al., 2022) is a strong zero-shot learner which is created by training LM-adapted T5 (Lester et al., 2021) on datasets covering multiple tasks, where each training instance is converted as prompted input and output; **T-Few** (Liu et al., 2022a) additionally pre-trains the new parameters introduced by $(\mathrm{IA})^3$ based on T0.

### 5.1.3 Experimental Settings

For few-shot fact verification, we report 4-shot experiments as the main result. We also conduct $K$-shot experiments for $K = \{1, 2, 4, 8, 16\}$ reported as supplementary results. For zero-shot experiments, we randomly sample 30 instances per class from each training set for fine-tuning. Note that no labels are used in this setting. For fair and robust comparison, we sample the training instances based on four random seeds and report the mean performance of macro-F1 and standard deviation over these four splits in all experiments. The seeds and data splits are kept the same across different models.

We use the original source code[2] of T-Few (Liu et al., 2022a) with its released pre-trained checkpoint of 3B parameters as our backbone model. Following the T-Few paper, we randomly sample a prompt template from the Public Pool of Prompts (P3) (Bach et al., 2022) for each instance at *each* training and inference step to increase the diversity and variability of prompts used. We set training steps as 1,500, batch size as 4, and learning rate as $1 \times 10^{-4}$ for both few-shot and zero-shot settings[3].

For fine-tuning the RoBERTa-L model, we follow Lee et al. (2021) using $2 \times 10^{-5}$ as learning rate and 32 as batch size, and train it for 10 epochs. We use the original code of GPT2-PPL[4] and conduct experiments using GPT2-base as the backbone following the original setting. Additionally, we also present the results of GPT2-PPL with a larger backbone GPT2-xl[5]. We reproduce SEED following the original implementation details in the

[2] https://github.com/r-three/t-few

[3] For all methods, we use the number of shots as batch size if the training size is less than the batch size.

[4] https://github.com/HLTCHKUST/Perplexity-FactChecking

[5] To deal with 3-way classification, we separate support and unsupported classes first, and then separate NEI and refutes classes from the predicted unsupported class, following the assumption that misinformation has higher perplexity (Lee et al., 2021)

| Datasets | Few-shot Methods | | | | | | | | Zero-shot Methods | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Majority | RoBERTa-L | GPT2-PPL$_{\text{base}}$ | GPT2-PPL$_{\text{xl}}$ | SEED$_{\text{nli}}$ | SEED$_{\text{mpnet}}$ | T-Few | ProToCo | T0-3B | T-Few | ProToCo |
| SciFACT | 0.195 | 0.210<br>(0.09) | 0.326<br>(0.04) | 0.348<br>(0.06) | 0.355<br>(0.05) | 0.273<br>(0.07) | 0.382<br>(0.05) | **0.498**<br>(0.03) | 0.315<br>(0.05) | 0.305<br>(0.03) | **0.331**<br>(0.02) |
| FEVER | 0.167 | 0.169<br>(0.01) | 0.293<br>(0.04) | 0.329<br>(0.10) | 0.501<br>(0.07) | 0.352<br>(0.02) | 0.851<br>(0.06) | **0.891**<br>(0.03) | 0.446<br>(0.03) | 0.433<br>(0.01) | **0.479**<br>(0.00) |
| VitaminC | 0.223 | 0.146<br>(0.02) | 0.303<br>(0.04) | 0.327<br>(0.04) | 0.313<br>(0.05) | 0.306<br>(0.04) | 0.489<br>(0.09) | **0.520**<br>(0.05) | 0.373<br>(0.02) | **0.400**<br>(0.00) | 0.386<br>(0.00) |

Table 2: Results of different few-/zero-shot fact verification methods in 4-shot and 0-shot settings on three datasets. We report the macro-F1 averaged over 4 trials with randomly selected training samples from the datasets using different seeds. The best results are in bold while the second results are underlined. The standard deviation is in (.).

paper (Zeng and Zubiaga, 2022) with BERT$_{\text{nli}}$[6] as its base model which was fine-tuned on NLI tasks. Furthermore, we report the results of SEED using the pre-trained model all-mpnet-base-v2[7] as backbone since it provides the best quality of sentence embeddings in all pre-trained models of sentence transformers (Reimers and Gurevych, 2019) [8]. We use the code and pre-trained checkponit with 3B parameters of T0 from Hugging Face Transformers[9]. All the experiments use a server with 4 NVIDIA Tesla-V100 32GB GPUs.

### 5.2 Few-Shot Result

The results of few-shot fact verification are reported in Table 2. We have the following observations.

**Firstly**, given very few labeled instances, RoBERTa-L does not always improve few-shot performance, which is consistent with the empirical finding that traditional fine-tuning of PLMs is unstable in the few-shot setting (Zhang et al., 2021; Mosbach et al., 2021; Dodge et al., 2020).

**Secondly**, with the designs for few-shot learning on PLMs, both versions of GPT2-PPL and SEED achieve much better performance than the majority class and RoBERTa-L, without any gradient update. With different backbone models, GPT2-PPL$_{\text{xl}}$ outperforms GPT2-PPL$_{\text{base}}$ due to its larger model size, while SEED$_{\text{mpnet}}$ lags far behind SEED$_{\text{nli}}$ possibly because the base model BERT$_{\text{nli}}$ fine-tuned on NLI task can be more readily adapted to the fact verification task compared to the base model all-mpnet-base-v2, which was fine-tuned on the sentence matching task. On SciFACT and FEVER datasets, SEED$_{\text{nli}}$ with the semantic difference vector outperforms GPT2-PPL$_{\text{xl}}$ that predicts labels based on a perplexity score. However, SEED$_{\text{nli}}$ is less advantageous on VitaminC as the semantic vector becomes less likely to identify the subtle factual differences in the contrastive instances.

**Thirdly**, our backbone model T-Few clearly outperforms both versions of GPT2-PPL and SEED, which indicates that only relying on the implicit knowledge of PLMs without parameter update is insufficient for few-shot fact verification. Also, compared to RoBERTa-L, the obtained improvements on all datasets shows the PEFT method (IA)$^3$ helps address the instability issue of traditional fine-tuning methods on PLMs under few-shot setting.

**Lastly**, ProToCo with consistency training leads to consistent gains on all datasets, considerably improving T-Few by 30.4%, 6.3% and 4.7% on SciFACT, VitaminC and FEVER, respectively, which demonstrates the effectiveness of imposing the consistency constraints on model training.

### 5.3 Zero-Shot Result

We examine the effectiveness of ProToCo in zero-shot setting, where it only uses a small number of *unlabelled* instances for training. The zero-shot result is also given in Table 2.

We can see that ProToCo performs better than T0-3B on all the datasets, achieving improvements by 7.4%, 5.1% and 3.5% F1 on FEVER, SciFACT and VitaminC, respectively. And our consistency training also improves T-Few by 10.6% and 8.5% in FEVER and SciFACT, respectively. However, ProToCo performs slightly worse than T-Few on VitaminC. Given the contrastive construction approach of VitaminC dataset, we conjecture that

[6] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

[7] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[8] https://www.sbert.net/docs/pretrained_models.html#model-overview

[9] https://huggingface.co/bigscience/T0

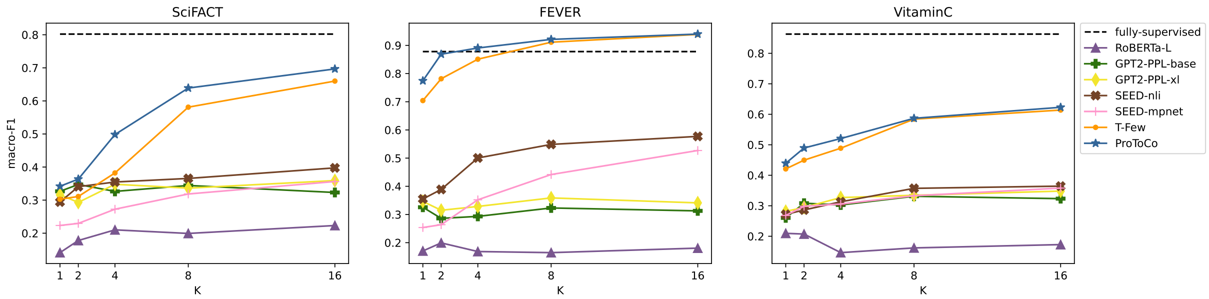


Figure 3: The performance comparison under different the number of shot $K$. For all $K$ tested, ProToCo consistently outperforms all the baselines. We report the results of fully-supervised models using oracle evidence as a reference: the results of RoBERTa-large model from Pradeep et al. (2021) and Pan et al. (2021) on SciFACT and FEVER, respectively; the result of ALBERT-xlarge model (Lan et al., 2020) on VitaminC is obtained by evaluating test set using the provided checkpoint and original code from Schuster et al. (2021).

| Dataset | ICL (OPT-30B) | | ProToCo (T-Few) | |
|---|---|---|---|---|
| | zero-shot | 1-shot | zero-shot | 1-shot |
| SciFACT | 0.332<br>- | 0.324<br>(0.08) | 0.331<br>(0.02) | 0.342<br>(0.05) |
| FEVER | 0.347<br>- | 0.442<br>(0.03) | 0.479<br>(0.00) | 0.774<br>(0.04) |
| VitaminC | 0.340<br>- | 0.284<br>(0.08) | 0.386<br>(0.00) | 0.439<br>(0.04) |

Table 3: Comparison between ProToCo (T-Few) and ICL (OPT-30B). Only an instruction is provided to OPT-30B in zero-shot setting. In few-shot setting, both task instruction and 3 training instances (1 shot) are provided.

this is might be because consistency training alone may not be able to effectively enhance the base model's ability to distinguish the contrastive instances without any supervision signals or prior adversarial training for the base model. One possible solution to address the issue is to use a stronger base model, which is pre-trained with adversarial data, to better capture the subtle differences in the contrastive instances. We will leave this to future work.

### 5.4 Impact of Shots Number

Figure 3 illustrates the comparison between few-shot baselines and ProToCo as the number of shots $K$ increases. ProToCo consistently outperforms the few-shot baselines at all $K$ on the three datasets. The curves of both versions of SEED and GPT2-PPL models quickly saturate compared to ProToCo and changing to a larger backbone cannot bring much improvements in GPT2-PPL method as $K$ increases, suggesting that fine-tuning PLMs is necessary for improving few-shot performance for new knowledge to be learnt.

Interestingly, the improvement of ProToCo over T-Few becomes clearly smaller as $K$ increases on FEVER and VitaminC that are based on Wikipedia data (as Wikipedia-like data might be seen during PLM pre-training), but on the scientific domain dataset SciFACT, consistency training still can lead to a modest improvement even when $K$ reaches 16 shots and is inclined to grow continuously. This indicates the consistency training is especially helpful when the PLMs knows little about the type of data in Scientific domain.

As $K$ increases, ProToCo continues to narrow the gap with the fully-supervised model that was fine-tuned on the full training set. On FEVER, only using 4 labeled instances per class, it already outperforms the fully-supervised model. On VitaminC, however, the trend suggests that its performance is not likely to catch up with the fully-supervised model. Our analysis shows that the chance is low to be able to sample contrastive instances into such limited number of shots of training data. As a consequence, the contrastive nature of this dataset might be underrepresented by the sampled instances, potentially limiting the model from effectively learning such features. We believe that using more training instances or a base model pre-trained on contrastive data might boost ProToCo's performance on VitaminC, but we will leave this to future work.

### 5.5 Comparison to ICL of Large PLMs

We compare ProToCo to ICL of relatively large PLMs in both few-shot and zero-shot settings. Specifically, we compare to OPT (Zhang et al.,

| Method | Datasets | | |
|---|---|---|---|
| | SciFACT | FEVER | VitaminC |
| SelfconCoT<br>(OPT-30B) | 0.289<br>(0.04) | 0.358<br>(0.07) | 0.258<br>(0.06) |
| ProToCo<br>(T-Few) | 0.342<br>(0.05) | 0.774<br>(0.04) | 0.439<br>(0.04) |

Table 4: Comparison with Self-Consistency Chain-of-Thought (Wang et al., 2022) using 3 training instances. The evaluation of FEVER and VitaminC are based on a random subset of test set given limited resources.

2022) with 30B parameters[10] – 10 times larger than ProToCo, which is an open-source large causal language model with similar performance as GPT-3 (Brown et al., 2020). Results in Table 3 show that ProToCo achieves much higher F1 score compared to the few-shot ICL with OPT-30B. Compared to the zero-shot ICL with OPT-30B, ProToCo clearly outperforms ICL on FEVER and VitaminC datasets and performs equally well on SciFACT. This confirms the effectiveness of ProToCo in both settings and demonstrates how the consistency training method enables a smaller PLM to compete with the ICL method using a much larger PLM on fact verification task.

### 5.6 Comparison to Self-Consistency Models

We compare ProToCo with the Self-Consistency Chain-of-Thought (SelfconCoT) method (Wang et al., 2022), which samples multiple outputs from a language model and returns the most consistent answer in the set. We implement the SelfconCoT method following the details described in (Wang et al., 2022) and use OPT with 6.7B parameters as its base model and sample 20 outputs for each instance[11]. Experiments are conducted with 3 training instances (1-shot) and evaluated on the full test set of SciFACT, and a random subset of the test set in FEVER and VitaminC given the limited resources we have.

Results in Table 4 show that ProToCo significantly outperforms SelfconCoT on all datasets, despite the fact that the latter has 2 times more parameters, suggesting that PLM with our consistency training is more suitable for fact verification task. Additionally, using the same hardware, ProToCo is considerably more efficient than SelfconCoT, taking around 6 hours to finish 4 runs of experiments thanks to the PEFT method, while SelfconCoT needs around 14 hours.

## 6 Conclusion and Future Work

We propose a model called ProToCo to improve few- and zero-shot fact verification based on consistency training of PLMs. Experiments on three public datasets show that ProToCo achieves promising fact verification performance by outperforming the existing few- and zero-shot baselines, the in-context learning on large PLMs, and the self-consistency chain-of-thought method. Our method also outperforms fully-supervised model on FEVER dataset.

In the future, we will explore few- and zero-shot solutions for other stages of fact-checking, e.g., evidence retrieval and justification generation, and combine them with ProToCo. We also plan to conduct experiments to evaluate the performance and level of consistency of larger language models (e.g., GPT-3 (Brown et al., 2020), InstructGPT (Ouyang et al., 2022) and LLaMA (Touvron et al., 2023)) on the fact verification task, when the computing resources are available.

## 7 Limitations

While ProToCo works well with our consistency training for improving fact verification under few-shot and zero-shot settings, our work has some limitations. Due to limited resources, currently we were unable to conduct comparison with larger PLMs and examine if extremely large models have already developed the similar or better level of consistency for fact verification on their own. In addition, our experiments show that consistency training brings improvements in both settings using only gold evidence. However, the retrieved evidence in real-world setting can be noisy and incomplete. That said, the performance of ProToCo on non-oracle evidence requires further study. To utilize consistency constraints, ProToCo still needs to fine-tune the PLMs. Also, in zero-shot setting, the labels of logical variants are assigned with the predictions of the original claim by the base model, which could be inaccurate and thus affect the consistency training.

[10] https://huggingface.co/docs/transformers/model_doc/opt

[11] Given the high cost of GPT-3.5 API and unavailability of checkpoints of PaLM (Chowdhery et al., 2022), we have opted to utilize OPT as the base model and chosen the largest checkpoint OPT-6.7B that can be accommodated with our compute resources.

## References

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Roberto Battiti. 2009. *Maximum satisfiability problem*, pages 2035–2041. Springer US, Boston, MA.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

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

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models:

Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ouyu Lan, Xiao Huang, Bill Yuchen Lin, He Jiang, Liyuan Liu, and Xiang Ren. 2020. Learning to contextually aggregate multi-source supervision for sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2134–2146, Online. Association for Computational Linguistics.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans and virtual*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2561–2571, Florence, Italy. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1022–1035.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. 2022. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6859–6866.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online. Association for Computational Linguistics.

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*, pages 94–103, online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *CoRR*, abs/2203.11171.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland. Association for Computational Linguistics.

Xia Zeng and Arkaitz Zubiaga. 2022. Aggregating pairwise semantic differences for few-shot claim verification. *PeerJ Comput Sci*, 8:e1137.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Prompt consistency for zero-shot task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2613–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Section 7*

☒ A2. Did you discuss any potential risks of your work?
*No risk.*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Abstract and Section 1*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*Section 5.1*

☑ B1. Did you cite the creators of artifacts you used?
*Section 5.1 and References*

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*The artifacts used in this paper are publicly available and free.*

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Section 5.1*

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Not applicable. Left blank.*

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Not applicable. Left blank.*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Section 5.1*

### C ☑ Did you run computational experiments?

*Section 5*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Section 5*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Section 5.1.3*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Section 5.2, 5.3, 5.4, 5.5 and 5.6*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Section 5.1.3, 5.6*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*Not applicable. Left blank.*

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*Not applicable. Left blank.*

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*Not applicable. Left blank.*

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Not applicable. Left blank.*

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*Not applicable. Left blank.*