Exploring Automatically Perturbed Natural Language Explanations in Relation Extraction

Wanyun Cui
Shanghai University of Finance and Economics
`cui.wanyun@shufe.edu.cn`

Xingran Chen
University of Michigan
`chenxran@umich.edu`

Abstract

Previous research has demonstrated that natural language explanations provide valuable inductive biases that guide models, thereby improving the generalization ability and data efficiency. In this paper, we undertake a systematic examination of the effectiveness of these explanations. Remarkably, we find that corrupted explanations with diminished inductive biases can achieve competitive or superior performance compared to the original explanations. Our findings furnish novel insights into the characteristics of natural language explanations in the following ways: (1) the impact of explanations varies across different training styles and datasets, with previously believed improvements primarily observed in frozen language models. (2) While previous research has attributed the effect of explanations solely to their inductive biases, our study shows that the effect persists even when the explanations are completely corrupted. We propose that the main effect is due to the provision of additional context space. (3) Utilizing the proposed automatic perturbed context, we were able to attain comparable results to annotated explanations, but with a significant increase in computational efficiency, 20-30 times faster.

1 Introduction

The application of neural networks in NLP has been a great success. However, the opaque nature of neural network mechanisms raises concerns regarding the reliability of their inferences and the potential for superficial pattern learning. This has led to severe issues with the generalization ability and vulnerability of neural networks. (Jia and Liang, 2017; Rajpurkar et al., 2018).

One common attempt to address the opacity of neural networks is to guide them with explanations. Researchers propose that by connecting the model and the inductive bias from explanations, the reliability of neural network inferences can be improved.

This approach has been successfully applied in relation extraction tasks, as demonstrated in previous studies such as (Srivastava et al., 2017; Hancock et al., 2018; Murty et al., 2020).

Earlier approaches relied on explanations from semantic parsers (Srivastava et al., 2017; Hancock et al., 2018), which incurs a high annotation cost. The recently proposed approach, ExpBERT (Murty et al., 2020), was a breakthrough in its ability to directly incorporate *natural language explanations*. For example, in Fig. 1a, o_1 *and* o_2 *went on a honeymoon* can be used as one explanation to guide the recognition of the spousal relation. ExpBERT with annotated explanations achieves 63.5% accuracy in the spousal relation extraction dataset, while BERT without explanations only achieves 52.9%.
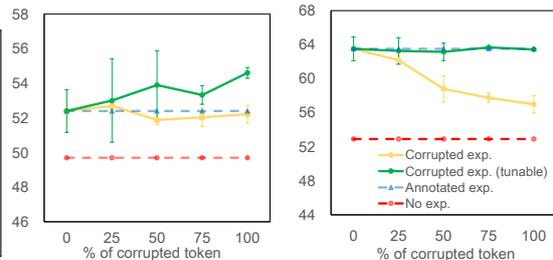
Considering the simple mechanism of ExpBERT, such improvement is quite surprising. ExpBERT simply concatenates explanations with the original text before being encoded by a language model. Based on the success of ExpBERT, one might conclude that text concatenation and pre-trained language models are sufficient for integrating the inductive bias from natural language explanations and guide models to make a sound inference. On the other hand, as exemplified by the history of deep learning, introducing extra inductive biases into neural networks is never trivial. Due to the strong generalization ability of neural networks, introducing inductive biases by humans is often surpassed by simpler models or even random inductive biases (Xie et al., 2019; Touvron et al., 2021; Tay et al., 2021a).

In this paper, starting from investigating the working mechanism of ExpBERT, we study how explanations guide and enhance the model effect. We first propose a simple strategy to control the inductive bias of explanations by the lens of *corrupted explanations*, wherein some words of annotated explanations are replaced by random words. We show an example of the corrupted explanation

3454

(a) Guide model with explanations.　　(b) Disease　　(c) Spouse

Figure 1: Corrupted explanations and their impact under frozen LM setting. Our results show that the random corruption does not result in a decrease in performance on the Disease dataset. While corruptions do lead to a reduction in the effect on the Spouse dataset, we find that even when explanations are 100% corrupted, they still result in an improvement over the baseline without explanations. These findings are an average of 3 runs.

in Fig. 1a, where the word *honeymoon* is replaced by the random word *frog*. Obviously, the explanations will provide less valid information after random corruption.

We show the effect of corrupted explanations in Fig. 1b and Fig. 1c. On both datasets, adding explanations shows a clear improvement to the baseline with no explanation. Surprisingly, however, reducing the inductive bias of explanations has almost no effect on the improvement of explanations on the Disease dataset. On the Spouse dataset, although the improvement decreases as the corruption increases, the effect of 100% corrupted explanations is still better than no explanations. These results suggest that the effect of explanations should not be entirely attributed to their inductive bias.

With comprehensive experiments of corrupted explanations in §3, we identified the following characters of natural language explanations:

- **Sensitivity** The effect of natural language explanations is sensitive to the training style and the downstream dataset. The previously observed improvement in accuracy and data efficiency (Murty et al., 2020) is only applicable for frozen language models. For fine-tunable language models, the improvement becomes less significant and does not generalize to all datasets.

- **Cause** The effect of the natural language explanations comes from the extra context space, rather than their inductive bias. Given enough context space, there is no significant effect decrease over annotated explanations even if they are completely corrupted.

- **Parameter search helps** The manual annotations provide a good initialization for that context

- although it can also be obtained via parameter search. If we randomly initialize the extra context and fine-tune it with downstream datasets, it achieves competitive or superior results over annotated explanations.

The above findings motivate us to further investigate and improve natural language explanations. To get rid of the potential entanglement of the existing vocabulary, we further proposed the *perturbed context* as the substitute, which only contains randomly initialized embeddings. We conducted different variants of perturbed contexts to investigate how natural language explanations work. We find that full-rank random contexts achieve competitive results with annotated explanations but are 18-29 times faster.

2 Background and Experimental Setup

2.1 Problem Definition

We consider the relation extraction task, which is frequently used for explanation guidance evaluation. Given $x = (s, o_1, o_2)$, where s is the target sentence, o_1 and o_2 are two entities which are substrings of s, our goal is to predict the relation y between o_1 and o_2.

Additionally, a set of natural language explanations $\mathcal{E} = \{e_1, \cdots, e_n\}$ are annotated to capture relevant inductive bias for this task. This setting follows (Murty et al., 2020). Note that these explanations are designed to capture the global information for all samples in this task, rather than for each example.

For example, for the spousal relationship, *"o_1 and o_2 went on a honeymoon"* is a valid explanation used in ExpBERT. We claim that this explanation constitutes a global inductive bias, and whether

3455

o_1 and o_2 went on a honeymoon will be seen as a feature to determine their spousal relationship for all samples. Similar global feature settings are also used in previous studies (Srivastava et al., 2017; Hancock et al., 2018; Murty et al., 2020).

2.2 Guiding Language Models with Explanations

Introducing annotated explanations Exp-BERT (Murty et al., 2020) is a state-of-the-art model for introducing annotated natural language explanations for relation extraction. In representing the samples $x = (s, o_1, o_2)$, ExpBERT first splices s with each natural language interpretation e_i by a separator $[SEP]$. Then it represents this splice by BERT:

$$\mathcal{I}(s, e_i) = \text{BERT}([CLS], s, [SEP], e_i, [SEP]) \in \mathbb{R}^d \quad (1)$$

where d represents the dimension of the hidden states of BERT. ExpBERT concatenates the n representations (i.e. $\mathcal{I}(s, e_1) \cdots \mathcal{I}(s, e_n)$) as the explanation augmented representation of s. The representation is then classified to the corresponding relation by a MLP classifier.

Introducing corrupted explanations work similarly to ExpBERT, except that a certain fraction of tokens in explanations are replaced by random tokens. The more tokens to be corrupted, the less inductive bias the explanation retains.

No explanation We also compare with the vanilla language model without introducing explanations. We use BERT as the language model by default.

2.3 Training Styles

We consider two training styles: fine-tuning and frozen language models. In **fine-tuning**, all parameters will be dynamically updated through back-propagation. In **frozen language models**, all parameters in the language model BERT are frozen after being pre-trained by an additional corpus (i.e., MultiNLI (Williams et al., 2018)), allowing only tuning the MLP classifier. This setting is used in (Murty et al., 2020).

2.4 Datasets

We follow (Murty et al., 2020) to use three benchmarks: Spouse (Hancock et al., 2018), Disease (Hancock et al., 2018), and TACRED (Zhang et al., 2017). We use the annotated natural language explanations provided by (Murty et al., 2020) for the baselines, except TACRED, whose natural

Table 1: Dataset statistics.

Dataset	Train	Val	Test	#Exp.
Spouse	22055	2784	2680	41
Disease	6667	773	4101	29
TACRED	68124	22631	15509	128

language explanations are not published. Therefore, we manually annotated 128 explanations as in (Murty et al., 2020). The statistics of the datasets is shown in Table 1. More details of the implementations are demonstrated in Appendix A.

3 Characterizing Natural Language Explanations

In this section, we make a thorough experimental study of the characters of natural language explanations. We plot the accuracy of different settings in Fig. 1 (frozen) and Fig. 2 (fine-tunable).

Character 1 *The effect and data efficiency of natural language explanations are sensitive to the training style and the dataset.*

Effect improves for frozen language models. For frozen language models (Fig. 1), introducing annotated natural language explanations significantly improves the effect over models without explanations. This is in line with the finding in (Murty et al., 2020).

The improvement becomes less significant for fine-tuning language models and varies across datasets. However, in the more common setting where all parameters are fine-tuned, the effect of annotated explanations is unstable (Fig. 2). On the Spouse and TACRED datasets, introducing annotated explanations has accuracy improvement, but not on Disease. The improvement is less significant compared to the frozen language models. This challenges the perception in previous work that introducing natural language explanations has significant effects (Murty et al., 2020).

Data efficiency only holds for frozen language models As to data efficiency, previous study (Murty et al., 2020) found that the effect of a model trained on a full amount of data without explanations can be achieved by using a small subset (e.g. 5%) of training data with explanations. However, they only verified this in frozen language models. We analyzed the data efficiency in fine-tunable language models. We vary the proportion of training data on different datasets. The results are shown in Fig. 3. Similar to the accuracy experiments, we find that
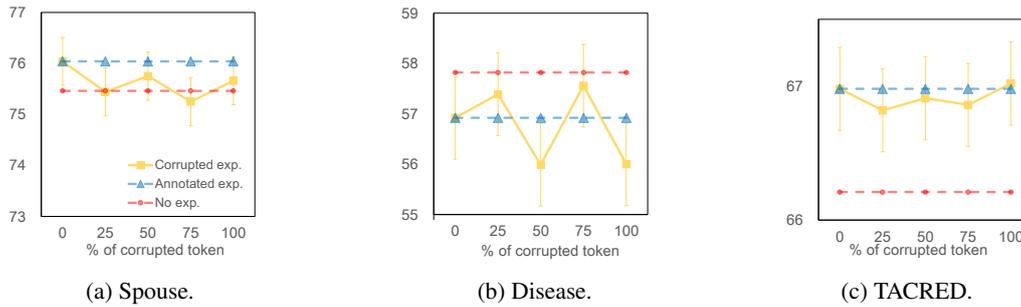
(a) Spouse. (b) Disease. (c) TACRED.

Figure 2: Results for fine-tunable language models. Annotated explanations show unstable and minor improvement over vanilla language models. The effectiveness did not significantly decrease after random corruptions.

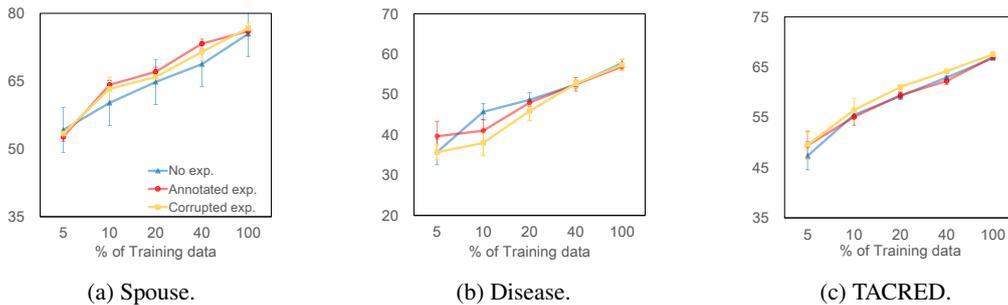(a) Spouse. (b) Disease. (c) TACRED.

Figure 3: For fine-tunable language models, introducing explanations does not improve data efficiency.

the data efficiency of natural language explanations does not generalize to fine-tunable language models. The natural language explanation does not improve the data efficiency, nor does the corrupted explanation.

Character 2 *Reducing the inductive bias of annotated explanations does not significantly decrease the effect.*

We also demonstrate the results of corrupting a certain fraction of tokens in annotated explanations in Fig. 1 and Fig. 2. The results are surprising: after corrupting with random words, the performance does not drop in most cases. Even when we replace 100% of the words in explanations with random words, the results are still competitive with the original explanations. This phenomenon was observed in all three datasets in the fine-tuning setting. Obviously, the 100% corrupted explanations do not provide any valid inductive bias for the model.

Interestingly, for the frozen language models, the results for Spouse and Disease diverge. In Fig. 1b, corrupting the explanations does not reduce the effect of Disease. The results of Spouse in Fig. 1c, however, shows that randomly corrupting the explanations does reduce the effect. Fig. 1c conflicts with other settings. We further investigate this exception below.

Character 3 *Parameter search over the corrupted tokens makes corrupted explanations comparable with annotated explanations in frozen language models.*

In the frozen language models, both the language model and the corrupted explanations are not fine-tunable. The randomly corrupted explanations may not be well-initialized if no optimization is allowed. We expect that the corrupted explanation will still be effective after proper initialization.

To verify this, we slightly modified the training strategy to allow the word embeddings of the corrupted words to be fine-tuned, while still freezing other parameters of the language model. That is, we search for the parameters for the corrupted explanation. Note that, during this process, no annotated explanation is used.

We show the results with tunable corrupted tokens in Fig. 1. We find that while not introducing extra annotations, the parameter search for the corrupted explanation performs competitively (on Spouse) or surpasses (on Disease) the annotated explanations. The results suggest that random initialization does not necessarily perform well. On the other hand, manually annotated explanations serve as a good initialization for the extra context.

4 How Do Explanations Work? An Investigation via Perturbed Context

The effect of corrupted explanations and natural language explanations reminds us to investigate how explanations work through their commonality: they both provide extra contexts. We hypothesize that the real factor at play is the context space provided by (corrupted) explanations, rather than the inductive bias. Hence, in this section, we systematically analyze how explanations work regarding the extra context space. To address this, we present perturbed context.

4.1 Definition of Perturbed Context

Our approach is inspired by the experiments of randomly corrupted explanations in § 3. According to the experiments, inductive biases of the explanations are not important, but rather we need a fine-tunable context to enrich the text representation. Therefore, we propose the perturbed context without any annotation to provide a fine-tunable context. We define a perturbed context e_v in the following form:

$$e_v := o_1 \; [M]_1 \; [M]_2 \; \cdots \; [M]_m \; o_2 \tag{2}$$

where o_1 and o_2 are placeholders for the two entities. We denote the embedding of $[M]_i$ as $emb([M]_i)$. We use $[M]_i$ as the normal token input in language models. We set $m = 4$ in this paper. $emb([M]_i)$ is randomly initialized without semantics.

4.2 Variants of Perturbed Contexts

As the extra context is the commonality between annotated explanations and corrupted explanations, we investigate how explanations work w.r.t. extra contexts. We introduce several variants of the perturbed contexts. We are particularly interested in (1) whether the perturbed context should be conditioned on the input x; (2) how the flexibility of the context space affects the results. The flexibility is controlled via factorization. We refer to the implementation of Synthesizer (Tay et al., 2021a), which is a recent study that revisits the inductive biases in attention.

Randomly perturbed contexts We consider the simplest form of the perturbed context, which consists of independent perturbed tokens that are randomly initialized. We set the perturbed contexts to be global and task-specific, rather than sample-specific. The randomly perturbed context is not conditioned on any input tokens.

Let $M \in \mathbb{R}^{m \times d}$ be a randomly initialized matrix. The embeddings of tokens in the randomly perturbed context are defined as:

$$emb_{rand}([M]_i) = M_i \tag{3}$$

The randomly perturbed context has $m \times d$ parameters. These parameters can either be trainable or kept fixed (denoted as **fixed random**).

Conditional perturbed context We also consider constructing perturbed contexts that are conditioned on each sample x. Here we adopt $F_i()$, a parameterized function, for projecting x to the embedding space of the perturbed tokens.

$$emb_{cond}([M]_i) = F_i(x_{pool}) \tag{4}$$

where $x_{pool} \in \mathbb{R}^d$ is the pooled BERT output for x. In practice, we use the multi-layer perceptron as $F(\cdot)$.

Factorized models We investigate the effect of the flexibility of the context. We refer to the factorization method in ALBERT (Lan et al., 2019). We map the original embedding of the perturbed token to a lower dimensional space, and then project it back to the original embedding space. The size of the intermediate space reflects flexibility. Following this idea, we further design the following variants.

Factorized randomly perturbed context We factorize the embedding of the randomly perturbed context by:

$$emb_{f\_rand}([M]_i) = W_{fr} M F_i \tag{5}$$

where $MF \in \mathbb{R}^{m \times l}$ is a lower dimensional embedding matrix, $W_{fr} \in \mathbb{R}^{d \times l}$. We first use MF_i to represent the lower dimensional space of size $l(l < d)$, then we project it back to the normal embedding space of size d.

Factorized conditional perturbed context Similarly, we also factorize the embedding of the conditional perturbed context by:

$$emb_{f\_cond}([M]_i) = W_{fc2}(W_{fc1} F_i(x_{pool})) \tag{6}$$

where $W_{fc1} \in \mathbb{R}^{l \times d}, W_{fc2} \in \mathbb{R}^{d \times l}$.

Mixture explanations We consider combining the perturbed context and manually annotated explanations to see if the two kinds of contexts complement each other. To ensemble the explanations, we add the perturbed context to the annotated explanation list.

Exploiting the perturbed context We use the perturbed context as the context of the original

sentence. Specifically, we use the pre-trained language model BERT to represent the target sample $x = (s, o_1, o_2)$. We construct e_v as a context for s by replacing the placeholders with o_1 and o_2, respectively. Then we use the [SEP] token as a separator to combine s and e_v. In this way, we convert the representation of s into the representation of $s + e_v$. We use the default setting in BERT to represent the sentence pair, i.e., using the final output of the [CLS] token as the representation of sentence pairs:

$$\mathcal{I}(s, e_v) = \text{BERT}([\text{CLS}], s, [\text{SEP}], e_v, [\text{SEP}]) \quad (7)$$

ExpBERT uses n explanations to improve the model, while we found one perturbed context achieves near-optimal results. We will give more experimental evidence in § 4.3 and Appendix C.

4.3 Effect on Fine-Tunable Language Models

We show the effect of different approaches in Table 2. We also compare with the following baselines that also introduce explanations in text understanding: BabbleLabble (Hancock et al., 2018), NeXT (Wang et al., 2020). Since ExpBERT is based on BERT, to make the comparison fair, we also use BERT as the language model by default. In addition, we also conducted experiments on RoBERTa (Liu et al., 2019).

Effect of perturbed contexts The proposed perturbed contexts overall achieve competitive performance with annotated explanations. This further verifies the claim in § 3, that the natural language explanations mainly provide extra context, rather than the specific inductive bias.

Among different variants of perturbed contexts, the fine-tunable random variant achieves the highest accuracy and outperforms the annotated explanations (ExpBERT) by $+1.15\%$ on average. We think this is because the annotated explanations have limited expressiveness and are cognitively biased. Thus a tunable context works better.

Fine-tunable vs fixed Surprisingly, we found that the fixed random variant performs competitively with the fine-tunable one on RoBERTa, and even achieves the highest accuracy in some settings. We think this is because the tunable language models provide sufficient flexibility to complement the flexibility of fixed perturbed contexts.

Global vs conditioned We found that the sample-specific variants (conditional and factorized conditional) have slight performance degradation compared to the global variant. We think this

is because the explanation learned from the training data is prone to contain certain biases. This makes it hard to learn a sample-specific perturbed context generator with high generalization ability. In contrast, learning generalized global perturbed contexts is easier. In terms of flexibility, conditional perturbed tokens are actually a projection of the representation of the original sentence x. This limits its flexibility. This also corroborates our analysis in Appendix B that we need a richer context to enhance the representation.

Effect of mixture explanations We evaluate the effectiveness of adding the randomly perturbed context to the annotated explanation list. The results in Table 2 show that mixture explanations have a slight performance degradation over the randomly perturbed context. This indicates that manually annotated explanations do not complement with the randomly perturbed context.

Efficiency Since the traditional approaches require the encoding of n explanations, they have substantial extra training/inference time compared to the vanilla language models. For example, ExpBERT encodes n explanations for each target sentence, which is extremely expensive when n is large. (e.g. TACRED has 128 explanations). Our proposed perturbed contexts, on the other hand, only need to encode a single sentence consisting of the target sentence and the perturbed context. It is almost as efficient as encoding the original sentence. Therefore, the efficiency of our approach is substantially improved compared to the previous work. We show the average training time of different approaches in Table 2. The training time of our approach is almost the same as that of language models without explanations. While having competitive effects, our approach is about $20 - 30$ times faster than ExpBERT which introduces explanations.

4.4 Effect on Frozen Language Models

For a comprehensive comparison, we also compare the effects of different models in the frozen language models. Note that the parameters of the perturbed context are fine-tunable except for the fixed random version. In addition to ExpBERT, we also compare our results with two settings of BERT + BabbleLabel as in (Murty et al., 2020), which uses the outputs of the labeling functions for explanations as features (BabbleLabble-ProgExp), and the encoding of the explanations by ExpBERT as

Model	Annotated	Spouse		Disease		TACRED		Avg. F1
		F1	**Time**	**F1**	**Time**	**F1**	**Time**	
BabbleLabble	Yes	50.1 ± 0.00	-	42.3 ± 0.00	-	-	-	-
NeXT	Yes	-	-	-	-	45.6	-	-
BERT	No	75.5 ± 0.59	1.0×	**57.8 ± 0.90**	1.0×	66.8 ± 0.31	1.0×	66.7
ExpBERT †	Yes	76.0 ± 0.47	28.5×	56.9 ± 0.82	20.1×	67.0 ± 0.14	32.1×	66.6
PC (BERT)								
+ R	No	76.7 ± 1.50	1.1×	57.3 ± 1.57	1.1×	**67.6 ± 0.55**	1.1×	**67.2**
+ Fixed R	No	74.9 ± 0.81	1.1×	56.2 ± 0.75	1.1×	66.9 ± 0.75	1.1×	66.0
+ C	No	75.8 ± 1.13	1.1×	57.3 ± 0.50	1.1×	66.6 ± 0.41	1.1×	66.6
+ FR	No	**77.1 ± 1.48**	1.1×	57.0 ± 0.51	1.1×	67.3 ± 0.25	1.1×	67.1
+ FC	No	75.8 ± 0.16	1.1×	56.6 ± 0.61	1.1×	66.8 ± 0.44	1.1×	66.4
Mixture	Yes	76.3 ± 1.06	28.5×	56.5 ± 1.22	20.1×	66.5 ± 0.69	32.1×	66.4
RoBERTa	No	77.1 ± 1.21	1.0×	58.8 ± 0.77	1.0×	69.2 ± 0.48	1.0×	68.4
ExpRoBERTa †	Yes	75.9 ± 0.77	21.2×	57.0 ± 1.22	21.3×	68.6 ± 0.69	20.6×	67.2
PC (RoBERTa)								
+ R	No	77.6 ± 0.52	1.1×	59.0 ± 0.55	1.1×	**70.1 ± 0.37**	1.1×	68.9
+ Fixed R	No	**78.2 ± 0.65**	1.2×	**59.4 ± 0.86**	1.1×	69.7 ± 0.43	1.0×	**69.1**
+ C	No	77.9 ± 0.56	1.3×	58.5 ± 0.50	1.2×	69.1 ± 0.24	1.2×	68.5
+ FR	No	77.5 ± 0.40	1.2×	57.0 ± 2.06	1.1×	69.3 ± 0.53	1.1×	68.0
+ FC	No	77.7 ± 1.01	1.3×	56.5 ± 1.12	1.2×	69.5 ± 0.15	1.2×	67.9
Mixture	Yes	75.9 ± 0.51	21.5×	57.1 ± 1.08	22.1×	68.7 ± 0.57	22.1×	67.2

Table 2: Results for fine-tunable language models. Without introducing external annotated explanations, the perturbed context achieves competitive results with models with explanations. The efficiency of the perturbed context is significantly higher than its competitors. We denote the perturbed context as **PC**. **R** means fine-tunable random. **C** means conditional. **Fixed R** means fixed random. **FR** means factorized random. **FC** means factorized conditional. Results are averaged over 5 runs. †: We fine-tune all parameters within ExpBERT and ExpRoBERTa.

Model	Spouse	Disease
BabbleLabel-LangExp †	53.6 ± 0.38	49.1 ± 0.47
BabbleLabel-ProgExp †	58.3 ± 1.10	49.7 ± 0.54
BERT-NoExp †	52.9 ± 0.97	49.7 ± 1.01
ExpBERT †	63.5 ± 1.40	52.4 ± 1.23
PC (BERT)		
+ R	**64.7 ± 0.62**	**53.5 ± 0.99**
+ Fixed R	not converged	45.4 ± 1.56
+ FR	60.5 ± 2.59	49.8 ± 0.96
RoBERTa-NoExp	62.2 ± 0.58	53.9 ± 0.32
ExpRoBERTa	65.8 ± 0.95	55.1 ± 0.31
PC (RoBERTa)		
+ R	**66.2 ± 2.18**	**55.7 ± 0.91**
+ Fixed R	41.6 ± 9.31	50.9 ± 0.99
+ FR	66.0 ± 2.60	54.5 ± 0.73

Table 3: Results for frozen language models. The perturbed context outperforms its competitors. †: result from (Murty et al., 2020).

features (BabbleLabble-LangExp). We omit the results of conditional and factorized conditional perturbed contexts, as they need the trainable parameters to learn $F_i()$ and are not suitable for frozen language models. The results are shown in Table 3.

The perturbed context still outperforms models with annotated explanations. This demonstrates that extra context space, rather than inductive bias, works for frozen language models. Unlike the results in Table 2, where the fine-tunable and fixed random variants are competitive, the fine-tunable random variant performs clearly better for frozen language models. We think this is because the model requires the perturbed context to provide flexibility via fine-tuning.

4.5 Effect of the Factorization

Results in Table 2 and Table 3 have shown that factorized models have slight performance degradation. To directly investigate the effect of flexibility/factorization, we control the size of the perturbed context of factorized models (i.e. l). We present the results of different ls in different training styles in Fig. 4.

We found that the choice of training style has a significant effect on flexibility. If fine-tuning is allowed, varying the size does not have a significant effect. However, for frozen language models, increasing the size significantly improves the results. This indicates that frozen language models require highly flexible perturbed contexts to enhance the effect. On the other hand, the tunable language model does not rely on the flexibility of the perturbed context. We think the flexibility has already been complemented by the tunable parameters in tunable language models.
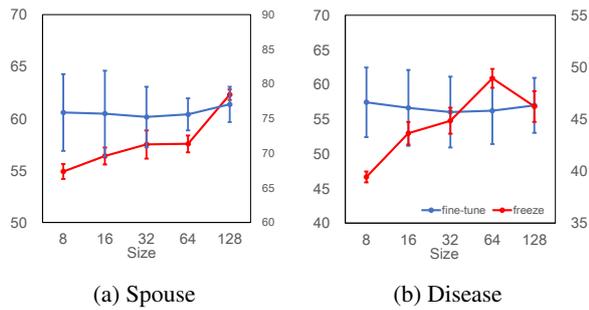
| (a) Spouse | (b) Disease |

Figure 4: Effect of the size of the perturbed context. For frozen language models, the effect is positively correlated with size. However, this does not hold for the fine-tunable language models.

4.6 Summary

Considering all the results in §4.3 4.4 4.5, the perturbed context has competitive or superior performance than annotated explanations if it has sufficient flexibility. The flexibility is obtained by one of the two fashions: (1) tunable perturbed context itself; (2) tunable language models.

5 Related Work

Introducing explanations in models Introducing explanations in text understanding has drawn many research interests. A typical class of methods is to construct explanations for specific domains, and then convert these explanations into features and combine them in the original model (Srivastava et al., 2017; Wang et al., 2017; Hancock et al., 2018). For example, Srivastava et al. (2017) use a semantic parser to transform their constructed explanations into features to apply to downstream tasks. Hancock et al. (2018) use the semantic parser as noisy labels, instead of features. Murty et al. (2020) argues that these semantic parsers can typically only parse low-level statements, therefore they use the language model as a "soft" parser to interpret language explanations, aiming to fully utilize the semantics of the explanations.

Revisiting the value of inductive biases Our paper presents the first proposal to revisit the inductive bias from explanations. Some previous studies worked on revisiting the inductive bias from network architectures (Touvron et al., 2021; Tay et al., 2021b; Liu et al., 2021), and some progress has been made. For example, Tolstikhin et al. (2021) found that, even only using multi-layer perceptrons, competitive scores on image classification benchmarks with CNNs and Vision Transformers (Doso-

vitskiy et al., 2020) could be attained. Guo et al. (2021) found that self-attention can be replaced by two cascaded linear layers and two normalization layers based on two external, small, learnable, shared memories. Melas-Kyriazi (2021) found that applying feed-forward layers over the patch dimension obtains competitive results with the attention layer. These studies all demonstrate that it it nontrivial to integrate valid inductive biases into neural networks.

6 Conclusion

In this paper, we revisit the role of *explanations* for relation extraction. In previous studies, explanations were thought to provide effective inductive bias, thus guiding the model learn the downstream task more effectively. We argue that it is imprudent to simply interpret explanations' effects as inductive bias. We find that the effect of natural language explanations varies across different training styles and datasets. By randomly corrupting the explanations, we found that the effect of explanations did not change significantly as the inductive bias decreased. This suggests that the inductive bias is not the main reason for the improvement. We further propose that the key of explanation for the improvement lies in the fine-tunable context. Based on this idea, we propose perturbed contexts. Perturbed contexts do not require any annotated explanations, while still providing (fine-tunable) contexts like annotated explanations. Our experiments verified that the effectiveness of the perturbed context is comparable to that of annotated explanations, but (1) the perturbed context does not require any manual annotation, making them more adaptable; (2) the perturbed context is much more efficient than that of using annotated explanations.

7 Limitations

This paper lacks a formalized analysis of the relationship among perturbed contexts/pre-training/model generalization. Although we try to analogize pre-training and prompt in Appendix B to explain how the perturbed context works, it lacks a rigorous mathematical description.

The validation of the perturbed context is limited to relation extraction. Although we show its potential on other applications in Appendix D, the experiments are still primitive. A more systematic evaluation on different NLP tasks is still excepted.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Adaprompt: Adaptive prompt-based finetuning for relation extraction. *arXiv preprint arXiv:2104.07650* (2021).

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettle-moyer. 2018. Ultra-Fine Entity Typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 87–96.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. 2021. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint arXiv:2105.02358* (2021).

Braden Hancock, Martin Bringmann, Paroma Varma, Percy Liang, Stephanie Wang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, Vol. 2018. NIH Public Access, 1884.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. 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)*. 3509–3514.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 897–907.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2021–2031.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).

Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. 2021. Pay Attention to MLPs. *arXiv preprint arXiv:2105.08050* (2021).

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

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-Shot Entity Linking by Reading Entity Descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3449–3460.

Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2017. World knowledge for reading comprehension: Rare entity prediction with hierarchical lstms using external descriptions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 825–834.

Luke Melas-Kyriazi. 2021. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. *arXiv preprint arXiv:2105.02723* (2021).

George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

Shikhar Murty, Pang Wei Koh, and Percy Liang. 2020. ExpBERT: Representation Engineering with Natural Language Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2106–2113.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021).

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 99–110.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 784–789.

Timo Schick and Hinrich Schütze. 2021. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2339–2352.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An Attentive Neural Architecture for Fine-grained Entity Type Classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction (AKBC)*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting Knowledge from Language Models Using Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4222–4235.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 conference on empirical methods in natural language processing*. 1527–1536.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2021a. Synthesizer: Rethinking self-attention for transformer models. In *International Conference on Machine Learning*. PMLR, 10183–10192.

Yi Tay, Mostafa Dehghani, Jai Gupta, Dara Bahri, Vamsi Aribandi, Zhen Qin, and Donald Metzler. 2021b. Are Pre-trained Convolutions Better than Pre-trained Transformers? *arXiv preprint arXiv:2105.03322* (2021).

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601* (2021).

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. 2021. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404* (2021).

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

Sida I Wang, Samuel Ginn, Percy Liang, and Christopher D Manning. 2017. Naturalizing a Programming Language via Interactive Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 929–938.

Ziqi Wang, Yujia Qin, Wenxuan Zhou, Jun Yan, Qinyuan Ye, Leonardo Neves, Zhiyuan Liu, and Xiang Ren. 2020. Learning from Explanations with Neural Execution Tree. In *International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 1112–1122.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38–45.

Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. 2019. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1284–1293.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 35–45.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1441–1451.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*. 78–83.

A Hyperparameters

We use the `bert-base-uncased` and `roberta-base` from Huggingface transformers (Wolf et al., 2020). We set the batch size $= 32$, learning rate $= 2e-5$, and train the model for 5

epochs for all three relation extraction tasks. By default, we set the size of the intermediate space of the factorized models to $l = 32$. 8 NVIDIA RTX 3090Ti GPUs are used to train the models.

Initialization For randomly perturbed context, we empirically found that the initialization of M will affect the results. After some trials, we found that initializing these parameters using a normal distribution with the mean and variance as in the token embeddings of the vanilla BERT is a practical choice.

B   Rationale and Relationship to Prompt Tuning

Our proposed perturbed contexts can be considered as fine-tunable contexts that guide model training. Prompt-tuning is a similar approach using fine-tunable languages. We compare their differences here.

Prompt tuning (Wei et al., 2021; Schick and Schütze, 2021; Shin et al., 2020; Li and Liang, 2021; Brown et al., 2020) utilize the pre-trained masked language modeling task and map the predictions of [mask] to the target label. For example, predicting `good` for the mask in `I love this movie. Overall, this is a` [mask] `movie` will classify it into a positive sentiment. However, for the relation extraction task of interest in this paper, it is difficult to establish the mapping between [mask] and relations by prompt due to the large label space (Chen et al., 2021). Our approach, on the other hand, can be applied to arbitrarily complex sentence classification tasks since the sentence representation is obtained directly from the [CLS] token.

Rationale The rationale for prompt tuning is that the pre-trained masked language model has a strong generalization ability for [mask] prediction. Therefore, the prompt performs well in the few-shot setting. Our perturbed context, on the other hand, exploits the generalization ability of the pre-trained language model for *contextual representation*. That is, given the target $sentence + context$, the language model can efficiently use a richer context to augment the target sentence. Based on the generalization ability for the rich context, given a fine-tunable perturbed context, the language model can automatically learn the optimal perturbed context as the context.

	Spouse	Disease	TACRED
Single	**76.7**	57.3	**67.6**
Multiple	75.1	**57.4**	67.4

Table 4: Comparison results of single/multiple perturbed contexts. Results are averaged over 5 runs. Using multiple perturbed contexts does not show surpassing effects.

C   Effect of Multiple Perturbed Contexts

Although we mainly discuss the scenario of a single perturbed context above, previous work (Murty et al., 2020; Hancock et al., 2018) have used multiple explanations. Therefore, we also validate the effect of using multiple perturbed contexts. Specifically, we formulate $n = 5$ randomly perturbed contexts:

$$e_{vi} := o_1 \ [V]_1^i \ [V]_2^i \ \cdots \ [V]_m^i \ o_2 \text{ for } i = 1 \cdots n \tag{8}$$

Then, we append each perturbed context e_{vi} to the original sentence s and represent these n sentence pairs as in ExpBERT. We concatenate the representations of all sentence pairs to form the resulting feature vector, and use an MLP over it to conduct the classification.

The results of multiple randomly perturbed contexts are shown in Table 4. We found that the improvement using multiple randomly perturbed contexts is not significant. We consider that this is because a single perturbed context already provides enough fine-tunable context.

D   Perturbed Contexts as Augmented Context? Application beyond Relation Extraction

Notice that our proposed perturbed contexts are actually fine-tunable contexts added to the original sample, which does not correspond to the semantics of any actual explanation. Therefore, it is natural to think that these perturbed contexts can be used not only as an alternative to annotated explanations for relation extraction, but also for broader applications.

It may be obvious that adding external relevant context can improve the representation of the target text. This idea has been verified on several tasks such as reading comprehension (Long et al., 2017), entity linking (Logeswaran et al., 2019), and even image classification (Radford et al., 2021). One typical class of the external context is the knowl-

edge description of entities. The model will jointly represent the target text and the descriptions of the entities within it to enhance the text representation.

In this section, we made a preliminary attempt at two fundamental tasks that involve entity knowledge: open entity typing (OpenEntity (Choi et al., 2018)) and word sense disambiguation (WSD (Raganato et al., 2017)). We study the effect of replacing knowledge-related contexts with perturbed contexts.

D.1 Tasks

Entity typing Given a sentence s, the goal of entity typing is to classify an entity ent in s. For example, for the sentence *Paris is the capital of France.* and the target entity *Paris*, the model is required to classify *Paris* into *Location*. We propose to use perturbed context as text augmentation for entity typing. To address this, we construct the randomly perturbed context in the form of $e_v := [V]_1 \cdots [V]_m \ ent \ [V]_{m+1} \cdots [V]_{2m}$. Then, we refer to Eqn.(7) to classify the augmented text by BERT.

WSD Given a sentence $s = w_1, \cdots, w_n$, and a polysemy word $w_t(1 \leq t \leq n)$ with candidate senses $\{c_1, c_2, ..., c_m\}$, WSD aims to find the sense for w_t. For instance, for the sentence *Apple is a technology company.* and the target polysemy word *Apple*, the model needs to recognize whether it refers to a fruit or a technology company. To augment the model effectiveness on WSD, we construct the perturbed context and model similar to the entity typing task. The only difference is, we follow (Huang et al., 2019) to use the final hidden state of the target word to conduct the classification.

D.2 Setup

Baselines For entity typing, We consider two types of baselines: the first type directly fine-tunes the target task. These baselines include BERT and NFGEC (Shimaoka et al., 2016). The second type first train the model over the joint corpus of the text and the external knowledge, and then fine-tune it on the target task. These baselines include Know-BERT (Peters et al., 2019) and ERNIE (Zhang et al., 2019). KnowBERT enhances contextualized word representations with attention-based knowledge integration using WordNet (Miller, 1995) and Wikipedia. ERNIE integrates knowledge through aligning entities within sentences with correspond-

	Dev	Test Datasets				
	SE07	SE2	SE3	SE13	SE15	**All**
BERT†	61.1	69.7	69.4	65.8	69.5	68.6
Our BERT	64.8	73.1	71.7	68.2	73.6	71.2
PC (BERT) + R	**66.2**	**74.6**	**72.5**	**68.4**	**74.0**	**72.0**

Table 5: Results on WSD datasets. †: results from (Huang et al., 2019).

Model	Joint pre-train	**P**	**R**	**F₁**
BERT ‡	No	76.37	70.96	73.56
Our BERT	No	75.98	73.42	74.68
NFGEC ‡	No	68.80	53.30	60.10
ERNIE ‡	Yes	78.42	72.90	75.56
KnowBERT ‡	Yes	**78.60**	**73.70**	**76.10**
PC (BERT) + R	No	77.42	72.95	75.12

Table 6: Results on OpenEntity. ‡: results from (Zhang et al., 2019).

ing facts in Wikidata (Vrandečić and Krötzsch, 2014). For WSD, we consider vanilla BERT as our baseline.

Hyperparameters We choose the same hyperparameters as in the relation extraction tasks, except that we train our model for 10 epochs for OpenEntity and 6 epochs for WSD, respectively. For WSD, we refer to the previous setting of training/valid/test splits (Zhong and Ng, 2010; Iacobacci et al., 2016).

D.3 Results

The results of the randomly perturbed context on WSD and OpenEntity are shown in Table 5 and Table 6, respectively. Our proposed approach still outperforms the baselines without joint pre-training. Even compared with baselines that use joint pre-training for knowledge integration, the performance degradation is not significant. This indicates that, to some extent, the perturbed context enhances the representation of texts that require entity knowledge. This shows the potential of our approach in different scenarios. We leave it to future work to explore the effects of the perturbed context on more tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Final section.

☒ A2. Did you discuss any potential risks of your work?
Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

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

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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.
Left blank.

C ☑ Did you run computational experiments?

In appendix

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
parameters same as BERT/RoBERTa.

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

☑ 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?
Our main experiments are run repeatedly for 3-5 times.

☑ 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.)?
We report in the appendix that we use hugging face framework.

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.