NormNet: Normalize Noun Phrases for More Robust NLP

Minlong Peng, Mingming Sun
Cognitive Computing Lab, Baidu Research, Beijing, China

{pengminlong, sunmingming01}@baidu.com

Abstract

A critical limitation of deep NLP models is their over-fitting over spurious features. Previous work has proposed several approaches to debunk such features and reduce their impact on the learned models. In this work, a normalization strategy is proposed to eliminate the false features caused by the textual surfaces of noun phrases. The motivation for this strategy is that noun phrases often play the role of slots in textual expressions and their exact forms are often not that important for performing the final task. As an intuitive example, consider the expression "x like eating y". There are a huge number of suitable instantiations for x and y in the locale. However, humans can already infer the sentiment polarity of x toward y without knowing their exact forms.

Based on this intuition, we introduce NormNet, a pretrained language model based network, to implement the normalization strategy. NormNet learns to replace as many noun phrases in the input sentence as possible with pre-defined base forms. The output of NormNet is then fed as input to a prompt-based learning model to perform label prediction. To evaluate the effectiveness of our strategy, we conducted experimental studies on several tasks, including aspect sentiment classification (ASC), semantic text similarity (STS), and natural language inference (NLI). The experimental results confirm the effectiveness of our strategy.

1 Introduction

Deep learning has proven quite effective in many NLP tasks (Collobert et al., 2011; Mikolov et al., 2013; Devlin et al., 2019). However, despite their great success and various NLP applications they power, critical limitations persist. A typical issue is their tendency to learn spurious features instead of the true signals of the task (Leino et al., 2019; Sagawa et al., 2020; Wang and Culotta, 2021; Yang et al., 2021b). This often leads to corrosive outcomes, from degraded performance on data in which the features no longer present (Kumar et al., 2019; Gui et al., 2021), to pernicious biases in model decisions (Blodgett et al., 2020), and to overall reduced trust in technology (Han and Tsvetkov, 2021).

This work proposes to address the spurious features caused by the textual surfaces of noun phrases. Here, we mainly consider noun phrases because they are often highly variable and replacing their forms would not make the sentence unreasonable. As a motivating example, consider the task to identify the sentiment polarity of `I` toward `apples` based on the textual expression `I like eating apples`, which belongs to the positive class. Here, "I" and "apples" can be changed to many other forms (e.g., "I" $\rightarrow$ "They", "Many people", etc.) with the resulting sentences still reasonable. If the model is trained on such an example, it may over-fit the spurious correlations between the positive class and `I` or `apples`. These spurious correlations will result in over-fitting, degrading the generalization performance and interpretability of the learned model.

Such a problem can be mitigated by pre-processing the original expression into x `like eating` y before feeding it as input to the learning model and reformulating the task to predict the sentiment of x toward y, where x and y denote two variables. In this way, the processed expression, together with its label, captures abstract knowledge independent of the specific forms of x and y—x is positive toward y no matter x is `I` or `They` and y is `apples` or `bananas`. In addition, such a pre-processing can make the learned model more interpretable and facilitate symbolic learning.

With this consideration, we propose the idea of sentence normalization, which aims to replace as many noun phrases in the input sentence as possible with specifically designed base forms. Figure 1 shows a normalized input sentence for

Findings of the Association for Computational Linguistics: ACL 2023, pages 2171–2183
July 9-14, 2023

an intuitive understanding. To implement the idea, we introduce a pretrained language model (PLM) based network, NormNet. Given an input sentence, NormNet will first identify noun phrases in the sentence. Then, it applies a PLM to evaluate the variability of every noun phrase conditional on its context. Phrases with high variability will be normalized to a base form, ranging from "A" to "Z". The resulting sentence will then be fed as input to the learning model to perform label prediction.

We tested the effectiveness of NormNet on three typical NLP tasks, i.e., Aspect Sentiment Classification (ASC), Semantic Text Similarity (STS), and Natural Language Inference (NLI). The experimental results show that our normalization strategy can improve both the models' in-domain and cross-domain performance. The contribution of this work is three-fold:

- We propose a novel idea of normalization for addressing the spurious features caused by the textual surfaces of noun phrases to deep NLP models.
- We introduce a pretrained language model based network, NormNet, to implement the proposed normalization strategy.
- Experimental studies on ASC, STS, and NLI tasks verify the effectiveness and reasonability of the proposed strategy.

2 Related Work

2.1 Data Augmentation

One of the related techniques to our idea is word-substitution-based data augmentation. This technique randomly replaces words or phrases with other strings, such as synonyms (Fadaee et al., 2017; Kobayashi, 2018), words having the same morphological features (Silfverberg et al., 2017), or words predicted by a pretrained language model (Wu et al., 2019; Wang et al., 2022; Bayer et al., 2022). For instance, give the expression "`I like eating apples.`", the technique may generate augmented expressions: "`They like eating apples.`", "`I like eating bananas.`", "`I love eating apples`", etc. Then, it will generate task labels for these expressions using some heuristics and append the generated samples to the training data set for model training. Such a technique has been found effective for various natural language processing tasks, such as machine translation (Xia et al., 2019), text classification (Feng et al., 2021), and dialogue understanding (Niu and Bansal, 2019).

However, such a data augmentation technique can be an expensive process. It will dramatically increase the overall size of the dataset by orders of magnitude. For example, if just substituting 2 words with 10 possible candidates for each sentence of the training data set, the dataset can easily grow by a factor of $10 \times 10 = 100$ (if applied independently). While this may have some benefits in terms of over-fitting, it can also significantly increase data storage costs and training time, which can scale linearly or super-linearly with respect to the training set size.

Instead of explicitly listing all the possible substitutions of a word or phrase through data augmentation, our method seeks to represent the possible substitutions with a consistent form, e.g., representing "`I like eating apples.`", "`They like eating apples.`", and "`I like eating bananas.`" with a consistently form "x `like eating` y.". Compared with the data augmentation technique, this method is much more efficient, with each sentence corresponding to only one normalized sentence.

2.2 Word Normalization

Another related techniques to the idea of our proposed normalization strategy are word-normalization and lemmatization (Schütze et al., 2008), which are two prevalent techniques in NLP to alleviate model over-fitting. They reduce inflectional forms and sometimes derivationally related forms of a word to a common base form based on heuristic rules and morphological analysis. For example, `am, is, are` will be stemmed to a consistent base form `be`.

The idea of our strategy is somehow motivated by the two techniques but has several critical differences. First, the base form of word-normalization and lemmatization is word dependent, making the normalized word vocabulary still large. In comparison, our strategy uses much simpler base forms. Second, word-normalization applies the normalization process to every word independently, ignoring its context. While our strategy uses a pretrained language model to model its context to determine which word or phrase should be normalized.

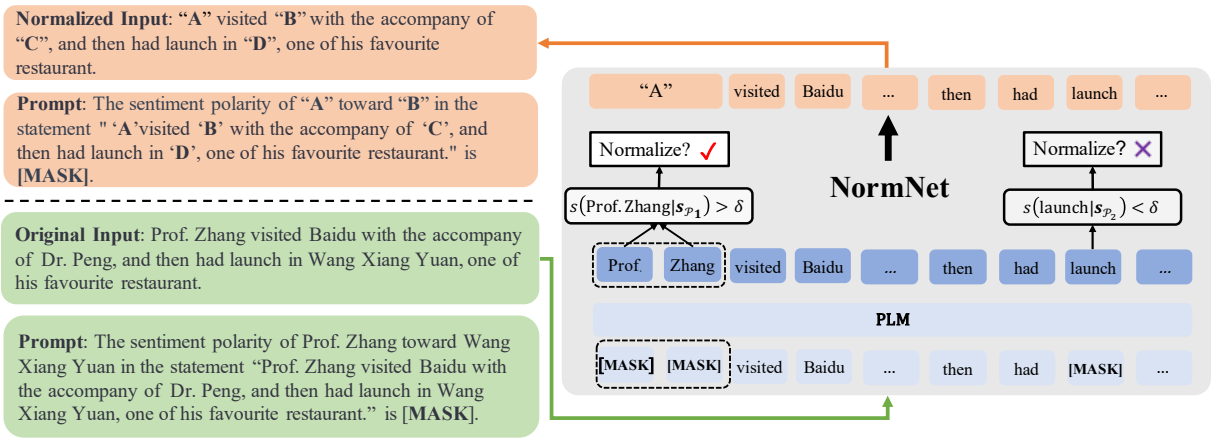

Figure 1: Schema of the proposed approach. It applies the prompt-learning-based method to perform label prediction. In addition, it introduces a NormNet module to simplify the input sentence, i.e., replacing some noun phrases with special tokens ("A"-"Z"). Here, $s_{\mathcal{P}_1}$ and $s_{\mathcal{P}_2}$ denotes the sentence with "Prof. Zhang" and "launch" being masked, respectively.

2.3 Symbolic Learning

A potential application of our strategy is to connect deep learning with symbolic learning. Symbolic learning uses symbols to represent certain objects and concepts, and allows developers to define relationships between them explicitly, e.g., (x `is the father of` y) $\wedge$ (y `is the father of` z) $\Rightarrow$ (x `is the grandfather of` z), with x, y, and z denoting three different variables (Mao et al., 2019). Based on the defined symbolic rules, it builds a rule system to perform the end tasks.

Because symbolic systems learn ideas or instructions explicitly and not implicitly, they are extremely data-efficient, interpretable, and robust to a cleverly designed adversarial attack (Evans and Grefenstette, 2018). However, a critical limitation of symbolic learning is that it requires developers to provide symbolic rules manually and the dominant data in the real world is non-symbolic, e.g., the natural language data. Thus, some recent work proposed to automatically mine rules from natural language data (Evans and Grefenstette, 2018) and applied deep learning to symbolic learning (Zhang and Sornette, 2017).

Our strategy can be seen as an combination of symbolic learning and deep learning. It replaces some noun phrases of natural language expressions with symbols and applies deep model to the resulting expressions to perform the end tasks. This can enhance the robustness of the deep model to the noise on the symbolic phrases. In addition, based on the normalized expression and learned model, it may mine symbolic rules with logic mining techniques like Markov Logic Network (MLN) (Richardson and Domingos, 2006) and perform symbolic reasoning with pre-defined rules. We leave this research in future work.

2.4 Prompt-tuning-based Model Learning

Nowadays, most NLP tasks are built on pre-trained language models (PLMs) (Kenton and Toutanova, 2019; Brown et al., 2020). A typical practice to utilize PLMs is adding a task-specific head on top of PLMs, and then fine-tuning the entire model by optimizing task-specific objectives on training data. However, most existing PLMs are trained with language modeling objectives, which usually differ from the learning objectives of downstream tasks. There is a gap between PLMs and downstream tasks, and the performance degradation introduced by the gap is often considerable when the downstream training data set is small.

To overcome the gap between pre-training and downstream tasks, another popular technique for utilizing PLMs has been introduced, which we call *prompt-tuning* in this work. In prompt-tuning, downstream tasks are formalized as language modeling problems by inserting language prompts, and the results of language modeling are heuristically mapped to the solutions of downstream tasks (Schick et al., 2020; Han et al., 2022). As shown in Figure 1, a typical prompt template has the form: "`<Input> <Prompt Words> [MASK].`" (the numbers and positions of each component may change). And there are a set of label words (e.g. "positive" and "negative")

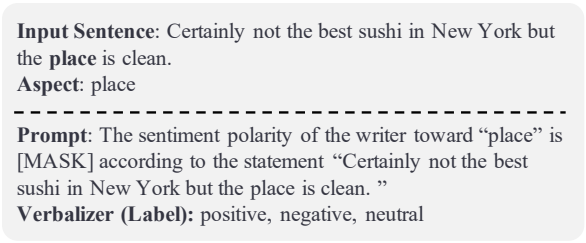

Figure 2: Template design for ASC

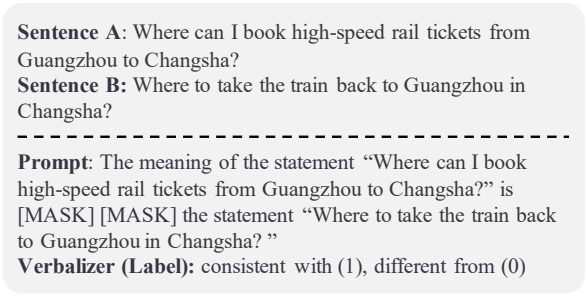

Figure 3: Template design for STS

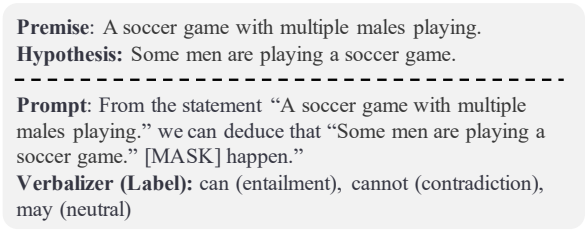

Figure 4: Template design for NLI

serving as the candidate set for predicting [MASK]. By fusing the original input with the prompt template for predicting [MASK] and then mapping predicted words to corresponding labels, prompt tuning converts a classification task into a language modeling task. Compared to the conventional fine-tuning method, prompt-tuning is more similar to the pre-training objectives, thereby helping to better use knowledge in PLMs and often obtaining better performance, especially when the training data set of the downstream task is small (Gu et al., 2022).

3 Methodology

Figure 1 shows the general architecture of the proposed method. It adopts a prompt-tuning-based learning model to perform the end tasks. And compared with the traditional prompt-tuning-based method, it additionally introduces a NormNet to pre-process the input sentence, which replaces some noun phrases in the input sentence with special tokens (we use "A"-"Z" in this work). The resulting sentence is then fed as input to the prompt-tuning-based learning model to perform model training and inference. In the following, we illustrate the detail of the prompt-tuning-based learning model and NormNet, respectively.

3.1 Prompt-tuning-based Learning Model

We adopt the popular prompt-tuning-based method to perform the end tasks. Here, we illustrate the template and verbalizer we used for ASC, STS, and NLI, respectively.

Template for ASC. Let $\boldsymbol{s}$ denote the input sentence, $\boldsymbol{a}$ denote the queried aspect, and [MASK] denote the mask placeholder. For performing the Aspect Sentiment Classification task, we apply the hard template with the form: `The sentiment polarity of the writer toward "`$\boldsymbol{a}$`" is [MASK] according to the statement "`$\boldsymbol{s}$`".` It accepts `"positive"`, `"negative"`, and `"neutral"` as the three possible predicted words at the position of [MASK], one mapped to an unique ASC label. We show an intuitive example of this template in Figure 2.

Template for STS. Let $\boldsymbol{s}_a$ and $\boldsymbol{s}_b$ denote the two text expressions of a STS example, and [MASK] denote the mask. For performing the Semantic Text Similarity task, we apply the hard template with the form: `The meaning of the statement "`$\boldsymbol{s}_a$`" is [MASK] [MASK] the statement of "`$\boldsymbol{s}_b$`".` It accepts `"consistent with"` and `"different from"` as the possible predicted phrase at the mask positions, which is then mapped to the label "1" and "0", respectively. We show an intuitive example of this template in Figure 3.

Template for NLI. Let $\boldsymbol{s}_p$ denote the premise expression, $\boldsymbol{s}_h$ denote the hypothesis expression, and [MASK] denote the mask placeholder. For performing the Natural Language Inference task, we apply the hard template with the form: `From the statement "`$\boldsymbol{s}_p$`" we deduce that "`$\boldsymbol{s}_h$`" [MASK] happen.` It accepts `"can"`, `"cannot"`, and `"may"` as the possible predicted words at the position of [MASK], which is then mapped to the NLI label "entailment", "contradiction", and "neutral", respectively. We show an intuitive example of this template in Figure 4.

Original Input: Certainly not the best sushi in New York but the place is clean.

Normalized Input: Certainly not the best sushi in "**A**" but the place is clean.

Aspect: place

Figure 5: Normalized example for ASC

Original Input: Where can I book high-speed rail tickets from Guangzhou to Changsha? [SEP] Where to take the train back to Guangzhou in Changsha?

Normalized Input: Where can I book high-speed rail tickets from "**A**" to "**B**"? [SEP] Where to take the train back to "**A**" in "**B**"?

Figure 6: Normalized example for STS

We finetune a PLM to perform the above tasks. Specifically, at training time, we finetune the PLM to predict the target words at the positions of [MASK]s using the mask prediction objective. At inference time, we applied the finetuned PLM to predict the masked words and accordingly, make label prediction.

3.2 NormNet

The right part of Figure 1 gives a working process of NormNet on an sampled input sentence, and Figure 5, 6, and 7 show a normalized example given by NormNet for ASC, STS, and NLI, respectively. *Note that, for STS and NLI, the two sentences of a single sample are concatenated into a single sentence, separated by [SEP], at this process. After that, the normalized sentence will be separated into two sentences using [SEP].* In general, NormNet involves three steps: **1)** identify noun phrases; **2)** determine which phrases should be normalized; and **3)** normalize the phrases and generate the output sentence.

Specifically, given an input sentence $\boldsymbol{s}$, NormNet first identifies the set of noun phrases, denoted as $\mathcal{N}$, in $\boldsymbol{s}$ using the spaCy chunking tool (Honnibal and Montani, 2017) with the "en_core_web_sm" model. Phrases occurring in different positions but having the same surface form in $\boldsymbol{s}$ correspond to a unique element in $\mathcal{N}$.

Then, for each phrase, $\mathcal{P} \in \mathcal{N}$, it replaces all the occurrences of $\mathcal{P}$ in $\boldsymbol{s}$ with [MASK]s (every constituent token of the phrase is replaced with a [MASK] token). The resulting sentence, $\boldsymbol{s}_{\mathcal{P}}$, is then fed as input to a pretrained masked language model to determine if $\mathcal{P}$ should be normalized. For

Original Input: A soccer game with multiple males playing. [SEP] Some men are playing a soccer game.

Normalized Input : "**A**" with multiple males playing. [SEP] Some men are playing "**A**".

Figure 7: Normalized example for NLI

the purpose, we calculate the mask prediction score of $\mathcal{P}$ given $\boldsymbol{s}_{\mathcal{P}}$:

$$s(\mathcal{P}|\boldsymbol{s}_{\mathcal{P}}) = \frac{-1}{C(\mathcal{P}) \times |\mathcal{P}|} \sum_{i=1}^{C(\mathcal{P})} \sum_{t \in \mathcal{P}_i} \log p(t|\boldsymbol{s}_{\mathcal{P}}; \boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{s}_{\mathcal{P}}$ denotes the sentence $\boldsymbol{s}$ with $\mathcal{P}$ being masked, $C(\mathcal{P})$ denotes the times of occurrence of $\mathcal{P}$ in $\boldsymbol{s}$, $\mathcal{P}_i$ denotes the i-th occurrence of $\mathcal{P}$, and $\boldsymbol{\theta}$ denotes the parameter of the pretrained language model, which is fixed at the process. *Intuitively, a high value of $s(\mathcal{P}|\boldsymbol{s}_{\mathcal{P}})$ indicates either $\mathcal{P}$ does not frequently occur in the background of $s_{\mathcal{P}}$ or the content occurring in the position of $\mathcal{P}$ is highly variable*. In both the cases, fitting the joint distribution of $\mathcal{P}$ and $s_{\mathcal{P}}$ using finite training data will easily result in over-fitting. Motivated by the intuition, we apply the following strategy to determine if $\mathcal{P}$ should be normalized: `If` $s(\mathcal{P}|\boldsymbol{s}_{\mathcal{P}}) > \delta$, `then` $\mathcal{P}$ `in` $\boldsymbol{s}$ `should be normalized, with` δ `being a scalar hyper-parameter.`

Finally, we replace each normalized phrase in $\boldsymbol{s}$ with a special token, ranging from "A" to "Z". The resulting input sentence is then fed as input to the learning model for model training and inference. At training time, the computational cost of NormNet is $\mathcal{O}(n_s n_p n_{plm})$, where n_s is the training data size, n_p is the average number of noun phrases in a sentence, and n_{plm} denotes the forward cost of the PLM. At inference time, the computational cost of NormNet is $\mathcal{O}(n_p n_{plm})$.

4 Experiment

4.1 Tasks & Datasets

Aspect Sentiment Classification. The task of aspect sentiment classification (ASC) involves predicting the sentiment polarity of a person toward a given aspect mentioned in the text written by the person. We performed experiments on two datasets for this task: the Multi-Aspect Multi-Sentiment **MAMS** (Jiang et al., 2019) dataset and the Restaurant review (**Rest14**) dataset from SemEval 2014 (Pontiki et al., 2016).

Semantic Text Similarity. Semantic text similarity tasks involve predicting whether two sentences are semantically equivalent or not. We performed experiments on two datasets for this task: the Microsoft Paraphrase corpus (MRPC) (Dolan and Brockett, 2005), and the Quora Question Pairs (QQP) dataset (Chen et al., 2018).

Natural Language Inference. The task of natural language inference (NLI) involves reading a pair of sentences and judging the relationship between them from one of entailment, contradiction or neutral. We evaluate the task on two datasets: the Multi-Genre Natural Language Inference (**MNLI**) dataset (Williams et al., 2018), and the Recognizing Textual Entailment (**RTE**) dataset (Bentivogli et al., 2009). For MNLI, we reported performance on its matched testing set.

For each task, we evaluated the model's *in-domain* and *cross-domain* performance. To evaluate the model's cross-domain performance, we trained the model on the training set of one dataset and tested its performance on the testing set of another dataset of the same task.

4.2 Baselines

We compared our method with three baselines. The first one is the method that applies the conventional finetuning method to perform the end tasks. We call this baseline as **PLMTuning**. The second baseline is the method that directly applies the prompt-tuning-based learning model to perform the task. We call this baseline as **PLMPrompt**. The third baseline is the method that additionally applies the word-substitution-based data augmentation technique based on PLMPrompt. We call this baseline as **PLMPrompt+SubAug**. To make a fair comparison between PLMPrompt+SubAug and our method (referred to as **PLMPrompt+NormNet**), we only performed substitution on phrases that were determined to be normalized by our method for implementing PLMPrompt+SubAug.

4.3 Implementation Detail

Implementation of PLMTuning. To perform the ASC task, we inserted the "[unused1]" token before and after the occurrence of the queried aspect in the input sentence. For instance, given the input sentence `"I like eating apples."` and the query `"apples"`, we reformulated the input sentence to `"I like eating [unused] apples [unused]."` and fed it as input to the learning model to perform model learning and inference. To perform the STS task, we concatenated the two input sentences (denoted as $\boldsymbol{s}_a$ and $\boldsymbol{s}_b$) of a sample into a single sentence connected by the special token "[SEP]": "$\boldsymbol{s}_a$ [SEP] $\boldsymbol{s}_b$" and "$\boldsymbol{s}_b$ [SEP] $\boldsymbol{s}_a$". At inference time, the label of the sample, ($\boldsymbol{s}_a$, $\boldsymbol{s}_b$), was obtained by averaging the prediction results on its two generated inputs. In similar, to perform the NLI task, we concatenated the two input sentences (denoted as Premise $\boldsymbol{s}_p$ and Hypothesis $\boldsymbol{s}_h$) of a sample into a single sentence connected "[SEP]" for the form: "$\boldsymbol{s}_p$ [SEP] $\boldsymbol{s}_h$". For all the tasks, label prediction was built on the final representation of the "[CLS]" token in PLMs. AdamW optimizer (Loshchilov and Hutter, 2018) with linear decay warm-up was applied for model learning. The initial learning rate was set to 2e-5 and the warm-up ratio was set to 0.1. Batch size was set to 32 for MAMS, Rest14, RTE, and MRPC and 64 for QQP and MNLI.

Implementation of PLMPrompt+SubAug. We performed substitution on phrases that were determined to be normalized by our method, and we applied the BART pretrained model (Lewis et al., 2020) to perform the substitution. Take the input `"I like eating apples."` as an example and suppose that we are to do substitution on `"apples"`. We would feed `"I like eating [MASK]."` as input to BART, which would generate 5 (2 for QQP and MNLI) phrases in the [MASK] position, each one consisting of up to 6 tokens. For each generated phrase, we would place it to the [MASK] position and then feed the resulting sentence to the chunking model to check if it is a noun phrase. If so, the phrase would be preserved as a candidate substitution of `"apples"`. Otherwise, it would be deprecated.

Implementation of NormNet. The pretrained language model of NormNet was implemented by ERNIE-Gram (Xiao et al., 2021), which was explicitly trained on a n-gram mask language model objective. For determining the value of δ, we first extracted named entities from the training set of each dataset using the spaCy NER tool with the "en_core_web_sm" model. We tuned the value of δ so that 70% of the extracted entities would be normalized. This was motivated by the fact that named entities often play the role of slot values in an expression (Louvan and Magnini, 2020) and it can often improve the NER performance with the

In-Domain	Rest14		MAMS	
	Acc	*F1*	*Acc*	*F1*
BERT-base-uncased*	82.74	73.73	78.86	78.01
PLMTuning	84.23	78.22	82.89	82.78
PLMPrompt	84.58	79.74	83.17	83.09
PLMPrompt+SubAug	84.69	79.81	83.25	83.17
PLMPrompt+NormNet	**84.97**	**80.18**	83.28	83.19
Cross-Domain	**Rest14 ↪ MAMS**		**MAMS ↪ Rest14**	
	Acc	*F1*	*Acc*	*F1*
PLMTuning	67.61	67.47	79.44	73.28
PLMPrompt	69.53	69.44	81.16	75.17
PLMPrompt+SubAug	70.24	70.15	82.73	76.08
PLMPrompt + NormNet	**72.21**	**72.09**	**84.01**	**77.65**

Table 1: In-domain and cross-domain performance on ASC. Results of BERT-base-uncased* on Rest14 and MAMS are retrieved from (Yang et al., 2021a) and (Lin et al., 2022), respectively. The PLM of the learning model was implemented by bert-base-uncased.

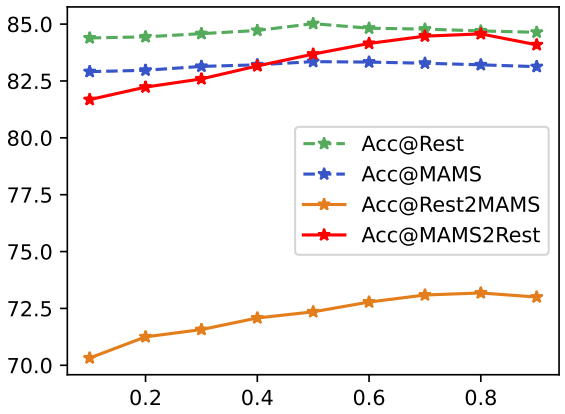

Figure 8: ASC performance by the normalization ratio controlled by α.

augmentation method that replacing an entity of a sentence with other entities of the same type (Dai and Adel, 2020).

All the experiments were run three times and the medium value within the three runs was reported.

4.4 Results on ASC

Table 1 shows the in-domain and cross-domain performance on the ASC task. From the table, we can see that: **1)** Compared with PLMTuning, PLMPrompt achieves slightly better in-domain performance and significantly better cross-domain performance on the two ASC datasets. This verifies the advantage of the prompt-tuning technique over the conventional fine-tuning technique on the two ASC datasets. **2)** PLMPrompt+SubAug achieves a little in-domain and about 1% absolute cross-domain performance improvement over PLMPrompt on the two tested datasets. This verifies the effectiveness of the word-substitution-based data augmentation technique. However, the computational cost of PLMPrompt+SubAug is about 10 times of that of PLMPrompt. **3)** PLMPrompt+NormNet achieves further improvement over PLMPrompt+SubAug, especially in the perspective of cross-domain performance, on the two ASC datasets. This verifies the advantage of our normalization strategy over the augmentation strategy on the two ASC datasets. As a conclusion, the proposed normalization strategy can bring consistent performance improvement to the prompt-tuning-based learning model and does better than the word-substitution-based data augmentation strategy, especially in the perspective of cross-domain performance.

Influence of δ. Here, we study the influence of δ. In the study, we adjusted the value of δ so that 10%-90% of the extracted entities would be normalized. Figure 8 shows the performance of the study. From the figure, we can see that our method performed quite robustly to the variation of δ for in-domain performance. Specifically, on Rest14, the accuracy varied between 84.39 (with 90% of entities normalized) to 85.02 (with 50% of entities normalized). On MAMS, the accuracy varied between 82.91 (with 10% of entities normalized) to 83.35 (with 60% of entities normalized). While the results on the cross-domain Reset14 to MAMS task showed that the accuracy varied from 70.32 (with 10% of entities normalized) to 73.18 (with 80% of entities normalized). On the cross-domain MAMS to Rest14 task, the accuracy varied between 81.68 (with 10% of entities normalized) to 84.57 (with 80% of entities normalized). It is worth noting that our method outperformed PLMPrompt in all settings of δ, which did not perform normalization. An interesting observation is that the when the normalization ratio increases from 0.8 to 0.9, both the in-domain and across-domain performance will slightly decrease. Our explanation to this phenomenon is that the some common entities can help knowledge generalization, like “sunshine" often indicating “positive" polarity. Removing such entities will slightly degrade the performance.

4.5 Results on STS

Table 2 shows the in-domain and cross-domain performance on the STS task. From the table, we can see that: **1)** From the perspective of in-domain

In-Domain	MRPC	QQP
BERT-base-uncased*	81.99	90.27
PLMTuning	82.35	**90.86**
PLMPrompt	84.56	90.27
PLMPrompt+SubAug	85.33	90.41
PLMPrompt+NormNet	**85.87**	90.65
Cross-Domain	**MRPC ↪ QQP**	**QQP ↪ MRPC**
PLMTuning	68.97	70.83
PLMPrompt	69.52	68.54
PLMPrompt+SubAug	70.08	68.83
PLMPrompt + NormNet	**72.11**	**72.32**

Table 2: In-domain and cross-domain performance on STS. The PLM of the learning model was implemented with bert-base-uncased. Results of BERT-base-uncased* are retrieved from (Choshen et al., 2022).

In-Domain	RTE	MNLI
BERT-base-uncased*	59.98	83.73
PLMTuning	63.54	83.56
PLMPrompt	67.17	83.62
PLMPrompt+SubAug	68.23	83.57
PLMPrompt+NormNet	**68.51**	**84.21**
Cross-Domain	**RTE ↪ MNLI**	**MNLI ↪ RTE**
PLMTuning	30.63	18.77
PLMPrompt	30.33	44.04
PLMPrompt+SubAug	30.67	44.28
PLMPrompt + NormNet	**32.48**	**46.57**

Table 3: In-domain and cross-domain performance on NLI. The PLM of the learning model was implemented with bert-base-uncased. Results of BERT-base-uncased* are retrieved from (Choshen et al., 2022).

performance, PLMPrompt performs considerably better than PLMTuning on MRPC but worse on QQP. While from the perspective of cross-domain performance, PLMPrompt achieves about 0.6% absolute improvement over PLMTuning on the cross-domain MRPC to QQP task but about 2.3% absolute decrease on the QQP to MRPC task. Our explanation to this phenomenon is that MRPC is a small dataset. Its only contains about 3.7k training samples, not large enough to completely adopt the language modeling objective to the finetuning objective. In contrast, QQP has about 370k samples. Thus, on QQP, prompt-tuning does not show advantage over the conventional fine-tuning technique. **2)** PLMPrompt+SubAug achieves a little performance improvement on the in-domain MRPC and the cross-domain MRPC to QQP tasks over PLMPrompt. However, on the in-domain QQP task and the cross-domain QQP to MRPC task, the performance improvement introduced by data augmentation is negligible. We believe this phenomenon is also resulting from the size of the training data size. **3)** PLMPrompt+NormNet achieves consistent improvement over PLMPrompt+SubAug on both in-domain and cross-domain tasks. This verifies the effectiveness of our normalization strategy and the advantage of our normalization over the word-substitution-based data augmentation technique.

4.6 Results on NLI

Table 3 shows the in-domain and cross-domain performance on the NLI task. From the table, we can see that: **1)** On the in-domain RTE task, PLMPrompt performs much better than PLMTuning, while on MNLI, the two models perform similarly. This observation is similar to that observed on the STS task, considering that RTE is also a small dataset (only contains about 2.5k training samples) and MNLI is a much larger dataset (contains about 393k training samples). On the two cross-domain tasks, PLMPrompt performs similar to PLMTuning. We think this is because the gap between RTE and MNLI is quite large and it does no matter what kind of tuning method applied. **2)** PLMPrompt+SubAug achieves about 1% absolute improvement over PLMPrompt on the in-domain RTE task and only about 0.1% absolute improvement on the in-domain MNLI task. On the two cross-domain NLI tasks, PLMPrompt+SubAug does not achieve much improvement over PLMPrompt. Here, we give our explanation to the results on the cross-domain RTE to MNLI task. Based on our data analysis, we think this results from the large gap between RTE and MNLI. The generated substitutions do not often occur in MNLI, making the data augmentation not so effective. **3)** On the in-domain RTE and MNLI tasks, PLMPrompt+NormNet achieves about 0.3% and 0.7% absolute improvement over PLMPrompt+SubAug, respectively. While on the cross-domain RTE to MNLI and the MNLI to RTE task, PLMPrompt+NormNet achieves about 1.8% and 2.3% absolute improvement over PLMPrompt+SubAug, respectively. Our explanation to this phenomenon is that our normalization strategy normalizes phrases of different domains into a consistent form, which is somehow equivalent to applying all possible substitutions. This makes our method more effective in the cross-domain

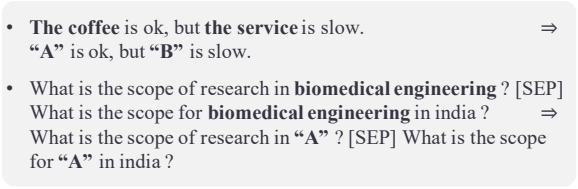

Figure 9: Samples on which PLMPrompt performs incorrectly but our method performs correctly.

scenario.

4.7 Qualitative Study

Here, we selected some samples, on which PLMPrompt made an incorrect prediction but our method made a correct one, and empirically study the reason. Figure 9 shows some of the selected samples. Through case study on these samples, we found that the class conditional distributions, $p(\mathcal{P}|y)$, of the normalized phrases in these samples are usually extreme. For example, "`coffee`" only occurs in positive training samples of Rest14, resulting in a strong connection between "`coffee`" and the positive class label. This may be the reason why PLMPrompt makes an incorrect prediction for "`coffee`" based on the expression "`The coffee are ok , but the service is slow .`", which belongs to the neutral class. In similar, "`biomedical engineering`" occurs 37 times in positive class training data and 7 times in negative class training data of QQP.

5 Conclusion

This work proposes a normalization strategy to overcome the spurious features caused by noun phrase surfaces. Experimental studies on Aspect Sentiment Classification (ASC), Semantic Text Similarity (STS), and Natural Language Inference (NLI) show that the proposed strategy can improve both models' in-domain and cross-domain performance. A potential extension of this work is extending the strategy to other types of phrases.

6 Limitations

We think this work has the following limitations: The **first** limitation is that our method involves additional computation for identifying noun phrases and determining which phrases should be normalized. The **second** limitation is that our method is only performed on noun phrases. Other phrases may also introduce spurious features. Extending our method to other types of phrases is a potential research direction. The **third** limitation is that due to the cost limitation, we did not test on the more powerful GPT-based PLMs, which proves to be more powerful and leads to heated discussions recently.

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs, https://www.kaggle.com/c/quora-question-pairs.

Leshem Choshen, Elad Venezian, Shachar Don-Yehia, Noam Slonim, and Yoav Katz. 2022. Where to start? analyzing the potential value of intermediate models. *arXiv preprint arXiv:2211.00107*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Richard Evans and Edward Grefenstette. 2018. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423.

Tao Gui, Xiao Wang, Qi Zhang, Qin Liu, Yicheng Zou, Xin Zhou, Rui Zheng, Chong Zhang, Qinzhuo Wu, Jiacheng Ye, et al. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. *arXiv preprint arXiv:2103.11441*.

Xiaochuang Han and Yulia Tsvetkov. 2021. Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4398–4409.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. 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 6280–6285.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457.

Sachin Kumar, Shuly Wintner, Noah A Smith, and Yulia Tsvetkov. 2019. Topics to avoid: Demoting latent confounds in text classification. 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 4153–4163.

Klas Leino, Matt Fredrikson, Emily Black, Shayak Sen, and Anupam Datta. 2019. Feature-wise bias amplification. In *International Conference on Learning Representations (ICLR)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Ting Lin, Aixin Sun, and Yequan Wang. 2022. Aspect-based sentiment analysis through edu-level attentions. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 156–168. Springer.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. 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 1317–1323.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Mach Learn*, 62:107–136.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. Promda: Prompt-based data augmentation for low-resource nlu tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14024–14031.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International conference on computational science*, pages 84–95. Springer.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796.

Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-gram: Pre-training with explicitly n-gram masked language modeling for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1702–1715.

Heng Yang, Biqing Zeng, Mayi Xu, and Tianxing Wang. 2021a. Back to reality: Leveraging pattern-driven modeling to enable affordable sentiment dependency learning. *arXiv preprint arXiv:2110.08604*.

Linyi Yang, Jiazheng Li, Pádraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021b. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. 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 306–316.

Qunzhi Zhang and Didier Sornette. 2017. Learning like humans with deep symbolic networks. *arXiv preprint arXiv:1707.03377*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☒ A2. Did you discuss any potential risks of your work?
We perform experiments on common public datasets.

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

☒ 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?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ 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.
No response.

C ☑ Did you run computational experiments?

4

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We used a public model for our experiments.

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?
4.3

☑ 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?
4.3

☑ 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.)?
3.2

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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.