CLIPText: A New Paradigm for Zero-shot Text Classification

Libo Qin[1], **Weiyun Wang**[2], **Qiguang Chen**[3], **Wanxiang Che**[3]
[1] School of Computer Science and Engineering
[1] Central South University, China
[2] OpenGVLab, Shanghai AI Laboratory, China
[3]Research Center for Social Computing and Information Retrieval
[3]Harbin Institute of Technology, China
lbqin@csu.edu.cn, wangweiyun@pjlab.org.cn,
{qgchen, car}@ir.hit.edu.cn

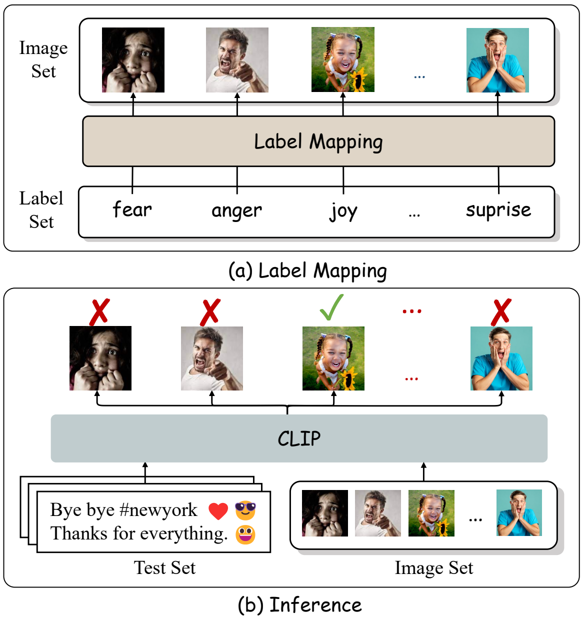
Abstract

While CLIP models are useful for zero-shot vision-and-language (VL) tasks or computer vision tasks, little attention has been paid to the application of CLIP for language tasks. Intuitively, CLIP model have a rich representation pre-trained with natural language supervision, in which we argue that it is useful for language tasks. Hence, this work bridge this gap by investigating a CLIP model for zero-shot text classification. Specifically, we introduce CLIPText, a novel paradigm for zero-shot text classification, which reformulates zero-shot text classification into a text-image matching problem that CLIP can be applied to. In addition, we further incorporate prompt into CLIPText (Prompt-CLIPText) to better derive knowledge from CLIP. Experimental results on seven publicly available zero-shot text classification datasets show that both CLIPText and Prompt-CLIPText attain promising performance. Besides, extensive analysis further verifies that knowledge from CLIP can benefit zero-shot text classification task. We hope this work can attract more breakthroughs on applying VL pre-trained models for language tasks.

1 Introduction

Understanding various modalities is one of the core goals of Artificial Intelligence. To achieve this, vision-and-language (VL) tasks such as visual question answering (Antol et al., 2015) and image caption (Chen et al., 2015) have emerged, aiming to test a system's ability to understand the semantics of both the visual world and natural language. Recently, CLIP (Radford et al., 2021), a cross-modality model pre-trained with 400M noisy image-text pairs collected from the Internet, has gained remarkable success on various VL tasks.

In addition, CLIP shows strong *zero-shot* transfer capabilities on over 30 different existing computer vision (CV) datasets (e.g., image classification (Jia et al., 2021) and object detection (Gu et al., 2021b)). In addition to success on CV tasks, various works begin to explore transferring knowledge of CLIP to other VL modality tasks. For example, Shen et al. (2021) demonstrate that serving CLIP as a strong visual encoder can benefit VL tasks in both pre-training and fine-tuning stage. Song et al. (2022) prove that CLIP can be considered as a strong few-shot learner for VL tasks by providing a comprehensive empirical study on visual question answering and visual entailment (Xie et al., 2019). Nevertheless, while significant recent progress has been made in applying CLIP to other VL and CV modality tasks, the same success has not yet been achieved in language tasks. In this work, we argue that CLIP was pre-trained with natural language supervision, which should be capable of helping



(a) Label Mapping

(b) Inference

Figure 1: Illustration of two steps in CLIPText. CLIPText consists of two steps (1) `Label Mapping`: aiming to map each label into a corresponding image and construct text-image pairs; (2) `Inference`: directly passing the generated text-image pairs into CLIP to obtain the final prediction results.

Findings of the Association for Computational Linguistics: ACL 2023, pages 1077–1088
July 9-14, 2023 ©2023 Association for Computational Linguistics

language tasks. Motivated by this, this work aims to close this gap by studying this research question: *can CLIP benefit language task?*

To this end, we provide a comprehensive investigation on zero-shot text classification task, aiming to studying how to transfer CLIP's zero-shot ability into the language task. Specifically, this work presents CLIPTEXT, a novel paradigm for zero-shot text classification. The key insight is that CLIPTEXT reformulates zero-shot text classification into a text-image matching problem, so that directly applying CLIP to zero-shot text classification can be achieved. As shown in Fig. 1, CLIPTEXT consists of procedure with two steps: (i) `Label Mapping` and (ii) `Inference`. Specifically, the `Label Mapping` step is used for mapping text classification label into a corresponding image, so that the text-image pairs can be constructed. Then, the `inference` step passes the generated text-image pairs into CLIP model, and the label with the highest alignment score is regarded as the prediction result. In addition, inspired by recent progress in prompt methods in natural language processing (Liu et al., 2021; Zhao and Schütze, 2021; Zhu et al., 2022; Hu et al., 2022; Qi et al., 2022), we further present PROMPT-CLIPTEXT by adding an additional semantic prompt word at the beginning of the text in CLIPTEXT, enabling model to better infer language knowledge from CLIP. Compared with previous methods, our method has the following advantages. First, some prior work (Yin et al., 2019) require additional NLI dataset to further train their zero-shot classification model. In contrast, our framework is capable of making full use of the powerful zero-shot capability of the CLIP without any extra pre-training. Second, we present a innovate perspective for zero-shot text classification, which can naturally leverage the additional vision information inferred from CLIP to benefit language tasks. Third, our framework is model-agnostic without any specific network design, thereby it can be easily extended to other VL pre-trained model.

We first evaluate our approaches on the standard zero-shot text classification benchmark (Yin et al., 2019). Experimental results show that CLIPTEXT and PROMPT-CLIPTEXT achieve superior performance. In addition, we further evaluate CLIPTEXT on other four publicly available zero-shot text classification datasets to verify the generalization of CLIPTEXT and PROMPT-CLIPTEXT.

In summary, contributions of this work are:

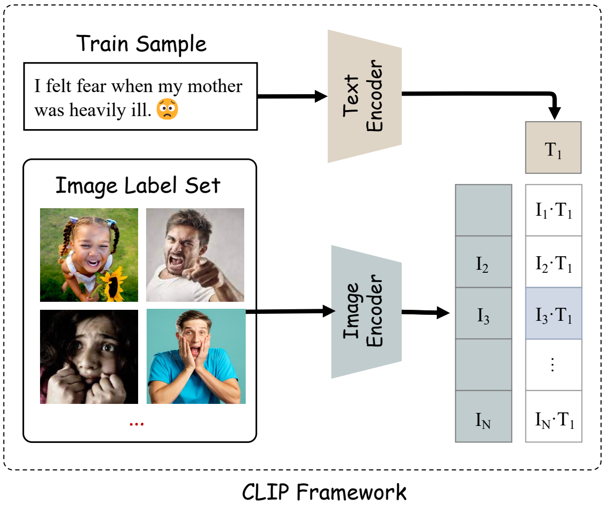


Figure 2: CLIP consists of a text encoder and image encoder, and followed by a dot product operation. The highest alignment score is predicted as the result.

- To our knowledge, this is the first work to investigate how to transfer zero-shot capabilities of CLIP into language tasks. We hope this work will spur more researchers to rethink the role of VL model for language tasks;
- We introduce CLIPTEXT, a novel paradigm for zero-shot text classification by reformulating it as a text-image matching problem. In addition, we further propose PROMPT-CLIPTEXT to better infer knowledge from CLIP to zero-shot text classification;
- Experiments on seven text classification datasets show the effectiveness of our framework. Extensive analysis further verify the generalization and superior of our approach.

To promote the further research, codes are will be publicly available at https://github.com/LightChen233/CLIPText.

2 Preliminaries

2.1 CLIP

CLIP (Contrastive Language-Image Pre-training) (Radford et al., 2021), an efficient and scalable approach to learn visual concepts from natural language supervision, has obtained surprisingly remarkable success on various of zero-shot computer vision tasks (Gu et al., 2021b). Instead of pre-training on traditional high-quality annotated data, CLIP is trained on 400 million noisy web-crawled image-text pairs, which is much easier to collect.

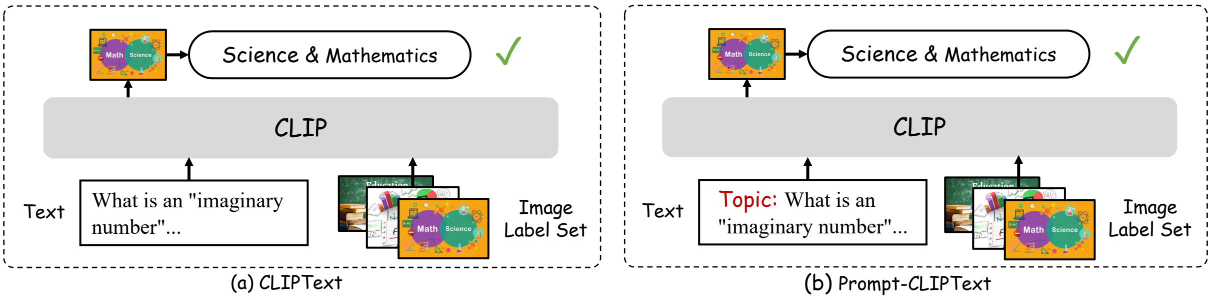


Figure 3: Illustration of CLIPTEXT (a) vs. PROMPT-CLIPTEXT (b). `Topic` stands for hard prompt for the text classification task.

As shown in Fig. 2 (a), CLIP contains a visual encoder $\mathbb{V}$ and a text encoder $\mathbb{T}$. Specifically, CLIP employs ResNet (He et al., 2016) or ViT (Dosovitskiy et al., 2020) as visual encoder backbone and uses transformer (Vaswani et al., 2017) as text encoder backbone. After text encoder and image encoder acquire text $\mathbb{T}(\text{text})$ and image $\mathbb{V}(\text{image})$ representation, a dot-product function $(\mathbb{V}(\text{image}) \cdot \mathbb{T}(\text{text}))$ is further used for calculating similarity between the given text and image. Specifically, the normalized similarity score of matching image i with text j can be calculated by:

$$\text{score}(i, j) = \frac{\exp(\beta \mathbb{V}(\text{image}_i) \cdot \mathbb{T}(\text{text}_j))}{\sum_{k=1}^{N} \exp(\beta \mathbb{V}(\text{image}_i) \cdot \mathbb{T}(\text{text}_k))}, \quad (1)$$

where β is a hyperparameter; N denotes the number of batch samples.

2.2 Zero-shot Text Classification

To provide an intuitive understanding of zero-shot text classification, we first introduce the classic *supervised text classification* and then describe the key difference between the supervised paradigm and *zero-shot paradigm*.

Supervised Text Classification Paradigm. In traditional *supervised text classification paradigm*, given training data $\mathcal{D}_{\text{train}}$, validation data $\mathcal{D}_{\text{dev}}$, test data $\mathcal{D}_{\text{test}}$, we first leverage $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{dev}}$ to train a model in supervised manner, and then apply the trained model to $\mathcal{D}_{\text{test}}$, which can be denoted as:

$$\mathcal{M} = \text{Train}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}), \quad (2)$$

$$Y = \text{Test}(\mathcal{M}, \mathcal{D}_{\text{test}}), \quad (3)$$

where $\mathcal{M}$ denotes the model trained on $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{dev}}$; Y represents the outputs of $\mathcal{M}$.

Zero-shot Text Classification Paradigm. In contrast to the supervised paradigm, following FitzGerald et al. (2022), zero-shot text classification model $\hat{\mathcal{M}}$ does not require any training process ($\mathcal{D}_{\text{train}}$), and can only access to the dev $\mathcal{D}_{\text{dev}}$ and test set $\mathcal{D}_{\text{test}}$. Model $\hat{\mathcal{M}}$ can be directly applied to test set without any training process ($\mathcal{D}_{\text{train}}$), which is formulated as:

$$\hat{Y} = \text{Test}(\hat{\mathcal{M}}, \mathcal{D}_{\text{test}}), \quad (4)$$

where $\hat{Y}$ represents the outputs of zero-shot text classification.

3 Model

This section illustrates how to solve the zero-shot text classification task with CLIP (see CLIPTEXT (§3.1) and PROMPT-CLIPTEXT (§3.2)).

3.1 CLIPTEXT

We convert the original text-label pairs in text classification into text-image pair to keep the CLIP original structure unchanged, To this end, CLIPTEXT consists of two step:

(i) `Step I Label mapping` (§3.1.1) converts text label into image to build text-image pairs;

(ii) `Step II Inference` (§3.1.2) passes the generated text-image pairs into CLIP to obtain the matching similarity score of each text-image pair and obtain the final zero-shot prediction results.

3.1.1 Step I: Label Mapping

Given test set $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{N}$ (N denotes the data number of test dataset), `label mapping` aims to convert the text label set $\mathcal{V}_{Label}$ into the corresponding semantic alignment image label set $\mathcal{V}_{Image}$ to build the text-image pairs.

In our framework, for each text label $\boldsymbol{y}$, we manually apply the google search engine directly to

Dataset	Type	Label Nums	Labels
Yahoo! Answers	Topic classification	10	`Health, Sports, ..., Politics & Government`
Emotion	Emotion classification	10	`sad, joy, love,..., none`
Situation	Situation classification	12	`search, evac, infra,..., crim., none`
AG's News	News categorization	4	`World, Sports, Business, Sci/Tech.`
Snips	Intent detection	7	`AddToPlaylist,..., SearchCreativeWork`
Trec	Question categorization	6	`NUM, HUM,..., ENTY, DESC`
Subj	Opinion classification	2	`objective, subjective`

Table 1: Statistics of the datasets.

find the corresponding image according to the dev performance[1]:

$$\boldsymbol{v} = \text{LabelMapping}(\boldsymbol{y}). \tag{5}$$

Therefore, with the help of label mapping step, the text-label pairs $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{N}$ can be mapped into text-image pairs $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{v}^{(i)})\}_{i=1}^{N}$ where $\boldsymbol{v} \in \mathcal{V}_{Image}$.

3.1.2 Step II: Inference

Given the generated text-image pairs $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{v}^{(i)})\}_{i=1}^{N}$, CLIP model can get a zero-shot prediction by:

$$\text{Inference}(\boldsymbol{x}, \boldsymbol{v}) = \begin{cases} \max_{\boldsymbol{v} \in \mathcal{V}_{\text{Image}}} \{\mathbb{V}(\boldsymbol{x}) \cdot \mathbb{T}(\boldsymbol{v})\} & \text{if } (\boldsymbol{x}, \boldsymbol{v}) \in \text{Single Label Task}, \\ \{\boldsymbol{v} | \mathbb{V}(\boldsymbol{x}) \cdot \mathbb{T}(\boldsymbol{v}) > t, \boldsymbol{v} \in \mathcal{V}_{\text{Image}}\} & \text{otherwise}. \end{cases} \tag{6}$$

where we select the label with the highest probability as the final prediction result in single label text classification task while we choose the labels greater than the threshold value t in multi-label classification.

3.2 Prompt-CLIPText

Similar to CLIPText, Prompt-CLIPText also contains `Label Mapping` and `Inference` step.

3.2.1 Step I: Label Mapping

Prompt-CLIPText employ the same `label mapping` step to acquire the constructed text-image pairs $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{v}^{(i)})\}_{i=1}^{N}$.

3.2.2 Step II: Inference

Instead of directly passing the $\mathcal{D}_{\text{test}}$ into CLIP, Prompt-CLIPText add an additional semantic prompt word at the beginning of input text $\boldsymbol{x}$ to generate a new prompt-guided text $\hat{\boldsymbol{x}}$ by:

$$\hat{\boldsymbol{x}} = \text{concat}(\texttt{Prompt}, \boldsymbol{x}), \tag{7}$$

where `Prompt` denotes the task-specific hard prompt word for different zero-shot text classification datasets.

Given the updated prompt-guided text-image pairs $\{(\hat{\boldsymbol{x}}^{(i)}, \boldsymbol{v}^{(i)})\}_{i=1}^{N}$, Prompt-CLIPText employ CLIP to obtain the final prediction by:

$$\text{Inference}(\hat{\boldsymbol{x}}, \boldsymbol{v}) = \begin{cases} \max_{\boldsymbol{v} \in \mathcal{V}_{\text{Image}}} \{\mathbb{V}(\hat{\boldsymbol{x}}) \cdot \mathbb{T}(\boldsymbol{v})\} & \text{if } (\boldsymbol{x}, \boldsymbol{v}) \in \text{Single Label Task}, \\ \{\boldsymbol{v} | \mathbb{V}(\hat{\boldsymbol{x}}) \cdot \mathbb{T}(\boldsymbol{v}) > t, \boldsymbol{v} \in \mathcal{V}_{\text{Image}}\} & \text{otherwise}. \end{cases} \tag{8}$$

Take the input text in Fig. 3 for example, the original input text in topic classification dataset $\boldsymbol{x}$ is {*What is an "imaginary number"...*} (Fig. 3 (a)), we insert an additional prompt word `topic:` to generate the prompt-guided text { `topic`*: What is an "imaginary number"...*}) (Fig. 3 (b)). The behind intuition is that prompt in Prompt-CLIPText can be regarded as a inductive prior knowledge to help the CLIP model to better understand the theme of text classification task and thus better transfer knowledge from CLIP to the language task.

Specifically, the prompt word for topic classification, emotion classification, situation classification, intent detection, news categorization, opinion classification and question categorization are `topic`, `interest`, `publication`, `type`, `clarify`, `caption` and `match`, respectively.

4 Experiments

4.1 Experimental Datasets

We first evaluate our approach on three standard zero-shot text classification benchmark, including: (1) **Topic classification**: Yin et al. (2019) choose Yahoo! Answers dataset (Zhang et al., 2015) to

[1] Specifically, after searching for some images based on the query, we first download the top M images returned by google search engine for each label. Then, we calculate the similarity between the image and the corresponding label name and map that label to the image with the highest similarity as an initialization. Finally, for each label, we fix the images corresponding to the remaining N-1 labels (N denotes the number of text label set), and then we try all M images in turn, record the performance of the model in the validation set, and take the image with the highest performance as the mapping image for that label.

evaluate topic classification. It consists of 10 topic categories; (2) **Emotion classification**: The Unify Emotion dataset was released by Bostan and Klinger (2018). It includes 9 emotion types; (3) **Situation classification**: Situation Typing dataset released by Mayhew et al. (2019). It includes 11 situation types.

To further demonstrate the generalization of our method, we take other four publicly available datasets, including: (1) **Intent detection**: We choose a wildly used intent detection benchmark Snips that is collected from the Snips personal voice assistant (Coucke et al., 2018), which contains seven intent labels; (2) **News categorization**: AG's news dataset (Conneau et al., 2017) is the most popular dataset for news categorization, which contains four news types; (3) **Opinion classification**: Subjectivity dataset (Subj) (Pang and Lee, 2005) from with two opinion categories; (4) **Question categorization**: Question dataset (TREC) (Li and Roth, 2002) contains six questions types. Detailed statistics of the datasets are summarized at Table 1.

4.2 Experimental Baselines

We compare the performance of our approach with the following strong zero-shot text classification baselines:

(1) `Majority`: This method directly adopts the most frequent label as output;

(2) `Word2Vec` (Mikolov et al., 2013): This approach first uses the average embedding to represent input text and label, and then applies maximum cosine similarity to obtain the final output;

(3) `ESA` (Chang et al., 2008): This method represents input text and label in the Wikipedia concept vector space, and then acquires final prediction output;

(4) `RTE` (Yin et al., 2019): This method is the entailment-based approach that considers the input text and label as entailment problem. RTE employ train a entailment model based on `bert-base-uncased` with RTE dataset;

(5) `MNLI` (Yin et al., 2019): Similar to RTE, this approach is a `bert-base-uncased` entailment model by pre-training on MNLI;

(6) `FEVER` (Yin et al., 2019): Similar to RTE and MLNL, FEVER is the `bert-base-uncased` model pre-trained on FEVER dataset;

(7) `NSP` (Ma et al., 2021): This method directly use next sentence prediction (NSP) pretraining task of BERT for zero-shot text classification. Specifically, it use the input text and text label as the sentence pair classification;

(8) `NSP (Reverse)` (Ma et al., 2021): Since NSP is not predicting for a directional semantic entailment, Ma et al. (2021) also explore a variant with all pairs reversed and refer it to `NSP (Reverse)`;

(9) `GPT-2` (Radford et al., 2019): We employ generative pre-trained model for zero-shot text classification tasks by directly generating each label output;

For these datasets without reported results, we use the open-source released by Yin et al. (2019) and Ma et al. (2021) to obtain results. All experiments are conducted in GeForce GTX TITAN X, 2080Ti and 3080.

4.3 Experimental Results

Following Yin et al. (2019) and Ma et al. (2021), we report label-wise weighted F1 for emotion and situtation datasets, and accuracy for other datasets.

Experimental results are illustrated at Table 2, we have the following interesting observations:

- Our framework obtains better performance against all baselines. Compared with the previous `NSP-base (Reverse)` model, CLIPTEXT obtains 4.6% improvements on AVG, which verifies our hypothesis that knowledge transferring from CLIP can benefit language task, even better than the knowledge from language itself pre-trained models.

- We do not observe any improvement when we replace `BERT-base` model in `NSP (Reverse)` with `BERT-large`. Besides, CLIPTEXT beats `NSP-large (Reverse)` by 9.1% while using fewer parameters, indicating simply increasing parameters of pre-trained model cannot solve zero-shot text classification.

- We observe that PROMPT-CLIPTEXT can outperform CLIPTEXT on six of seven datasets, which indicates the effectiveness of PROMPT-CLIPTEXT and it can better infer knowledge

Model	Model Size	Yahoo! Answers	Emotion	Situation	AG's News	Snips	Trec	Subj	AVG
Non pre-trained Language Models									
Majority	-	10.0	5.9	11.0	25.0	17.7	-	-	-
ESA (Chang et al., 2008)	-	28.6	8.0	26.0	73.3	63.4	-	-	-
Word2Vec (Mikolov et al., 2013)	-	35.7	6.9	15.6	44.1	63.6	-	-	-
Pre-trained Language Models									
RTE (Yin et al., 2019)	110M	43.8	12.6	37.2	56.7	56.4	27.2†	55.7†	41.4
FEVER (Yin et al., 2019)	110M	40.1	**24.7**	21.0	78.3	69.4	31.8†	56.8†	46.0
MNLI (Yin et al., 2019)	110M	37.9	22.3	15.4	72.4	77.6	33.8†	44.8†	43.5
NSP-BERT-base (Ma et al., 2021)	110M	50.6	16.5	25.8	72.1	73.4	32.4†	48.4†	45.6
NSP-BERT-large (Ma et al., 2021)	350M	43.2†	18.4†	25.7†	70.5†	68.4†	44.8†	42.1†	44.7
NSP-BERT-base (Reverse) (Ma et al., 2021)	110M	53.1	16.1	19.9	78.3	81.3	38.0†	61.8†	49.8
NSP-BERT-large (Reverse) (Ma et al., 2021)	350M	49.7†	19.1†	22.7†	74.4†	63.7†	28.4†	59.1†	45.3
GPT-2 (Radford et al., 2019)	124M	18.7†	12.5†	11.8†	62.3†	18.9†	15.2†	51.4†	27.3
CLIP Text Encoder (Radford et al., 2021)	38M	40.0†	12.5†	30.6†	65.6†	60.8†	37.8†	53.7†	43.0
Pre-trained VL Models - Single Model									
CLIPText	151M	53.6	22.0	37.4	77.0	81.0	41.6	68.0	54.4
Prompt-CLIPText	151M	**53.7**	21.3	**38.8**	**78.4**	**81.8**	**48.4**	**68.5**	55.8
Pre-trained VL Models - Ensemble Model									
CLIPText (Ensemble Model)	151M	55.9	24.7	37.9	77.5	82.9	46.2	69.0	56.3
Prompt-CLIPText(Ensemble Model)	151M	56.1	23.4	39.6	79.4	84.7	51.6	74.1	58.4

Table 2: Zero-shot Main Results. AVG denotes the average score on all datasets. Results with † are obtained by re-implemented and other results are taken from the corresponding published paper (Chang et al., 2008; Mikolov et al., 2013; Yin et al., 2019; Ma et al., 2021). Results with `BERT-base` denotes that models use `BERT-base` as backbone and with `BERT-large` represents that models use `BERT-large` as backbone. Results with - denotes the missing results from the corresponding published work.

from CLIP to enhance zero-shot text classification.

4.4 Analysis

To better understand our model, we provide comprehensive analysis to answer the following questions:

(1) Whether the vision knowledge from CLIP benefits the language task?

(2) Whether it be better to convert a label to multiple images and then ensemble them?

(3) Why our approach can successfully perform zero-shot text classification?

(4) What is the intuition behind of our approaches?

(5) What is the impact of image selection?

4.4.1 Answer 1: Vision Knowledge inferred from CLIP can Benefit Zero-shot Text Classification

In this section, we investigate whether the vision knowledge inferred from CLIP can benefit zero-shot text classification. To this end, we conduct experiments by directly encoding both text and label by CLIP Text Encoder and calculating the similarity score to predict the final results. We refer it to the `CLIP Text Encoder`.

Table 2 (`CLIP Text Encoder`) illustrates the results. We observe that our framework surpasses `CLIP Text Encoder` by a large margin (54.4% vs. 43.0%), indicating that the image knowledge learned from CLIP text-image matching pretraining benefits zero-shot text classification tasks.

4.4.2 Answer 2: Ensemble Model Boosts Performance

This section investigates the effectiveness of ensemble approach. Specifically, each text label x is converted into two corresponding images and we sum the two text-image alignment scores as the final prediction score.

Table 2 (ensemble) shows the results. We observe that ensemble mode can consistently outperform the single model on the CLIPText and Prompt-CLIPText, which suggests different images can provide different knowledge and views for text, thereby promoting the performance.

4.4.3 Answer 3: Why CLIPTextC Works

To analyze why our approaches work, we provide an intuitive visualization analysis on CLIPText. We choose representations of each text from CLIP text encoder $\mathbb{T}$ and the corresponding image label from CLIP vision encoder $\mathbb{V}$ for visualization.

Fig. 4 shows the t-SNE visualization output, where we observe that the image representation and the corresponding text representation are close

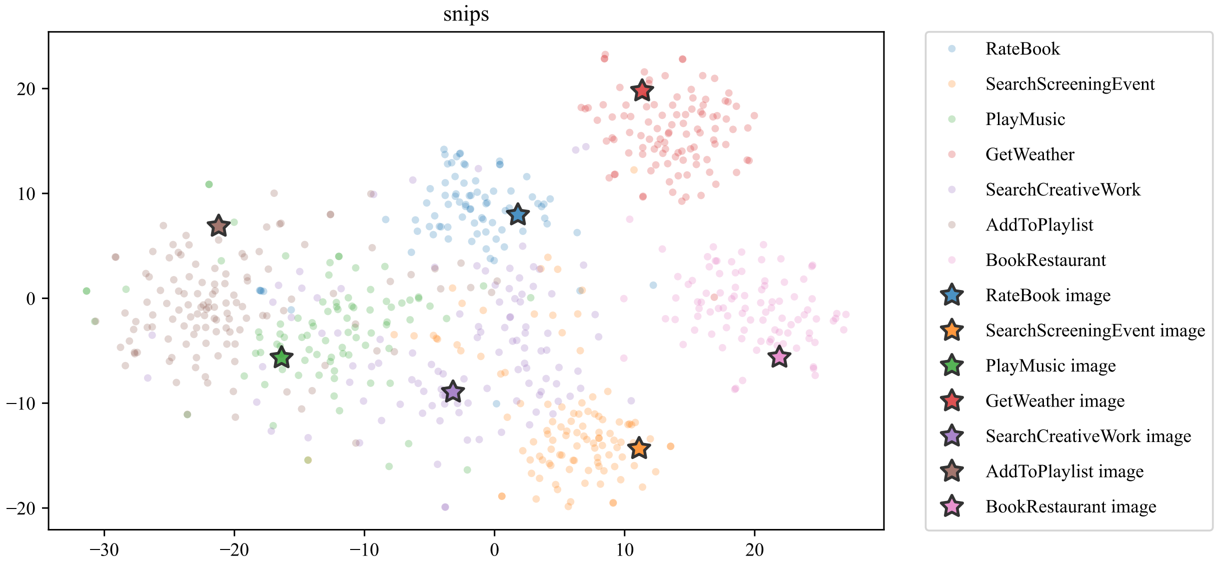


Figure 4: t-SNE visualization of text vectors (dots) from CLIP text encoder $\mathbb{T}$ and image vectors (pentagram) from CLIP image encoder $\mathbb{V}$. The dots in the same color represents text representation with same intent and different colors denote different languages.

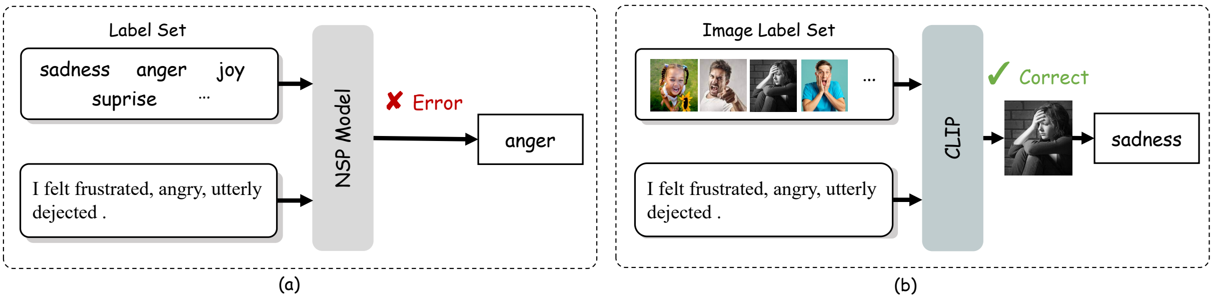


Figure 5: Case study. Red color denotes the wrong prediction while green color represents the correct prediction.

to each other, which demonstrates that the powerful cross-alignment capabilities of CLIP, enabling the model to perform zero-shot text classification.

4.4.4 Answer 4: Qualitative analysis

To intuitively understand our approach, we conduct qualitative analysis by providing a case study in emotion classification task produced by CLIP-Text and NSP (Reverse).

Fig. 5 illustrates the vase study. Given the input text *"I felt frustrated , angry , utterly dejected."*, NSP Reverse model predicts the label angry incorrectly. We suspect that the spurious cues word *angry* in the text confuse the NSP Reverse model to predict angry. In contrast, our approach CLIP-Text predicts the label sadness correctly. This further demonstrates that the rich information in the image can help our model to make a correct prediction compared with single text label in traditional zero-shot text classification model.

4.4.5 Answer 5: Impact of Image Selection

An interesting question arise is what is the impact of image selection in label mapping stage. To answer this question, for each text label, after obtaining M images returned from google search engine, we randomly choose one image from M images as the mapping image. Finally, we try 30 different experiments and obtain the standard deviation.

Results are illustrated in Fig.6, which shows a slightly high standard deviation on each dataset. Therefore, future work can focus on how to automatically select label mapping, which is an interesting and important topic to investigate.

4.4.6 Potential Impact

Recently, CLIP (a powerful vision-and-language (VL) model) has shown remarkable success on various zero-shot VL and compute vision tasks. Inspired by this, our work make the first attempt to investigate how to transfer knowledge of CLIP to language task. To achieve this, we introduce CLIP-

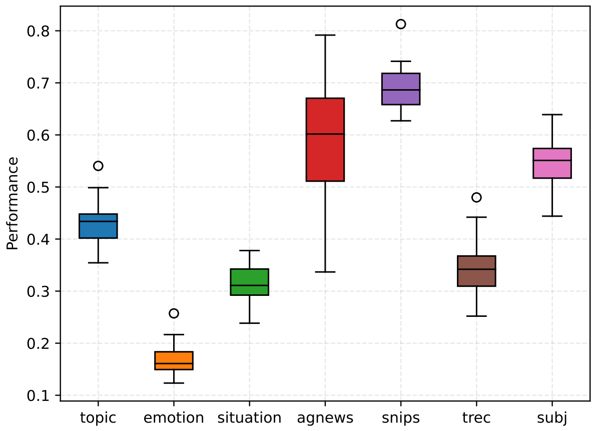


Figure 6: Performance distribution boxplots for each task over 30 random experiments.

TEXT and PROMPT-CLIPTEXT, a novel paradigm for zero-shot text classification by reformulating it into a text-image matching problem. Our work demonstrates CLIP can be a good zero-shot learner in language task and we hope this work will attract more researchers to explore how to better leverage knowledge of VL model to help language tasks.

5 Related Work

In this sectin, we discuss the related work of zero-shot text classification task and application of CLIP.

5.1 Zero-shot Text Classification Task

Zero-shot text classification allows model directly to make prediction without any training process, which gains increasing attention since it can greatly reduce human annotation efforts. Yin et al. (2019) introduce three zero-shot text classification benchmarks and propose some strong entailment-based baselines to facilitate this line of research. Puri and Catanzaro (2019) introduce a generative language model (e.g., GPT-2) for zero-shot text classification. Ma et al. (2021) explore the powerful zero-shot ability of BERT for zero-shot text classification, which achieves promising performance. Compared with their work, our approaches explore the zero-shot capacities of VL model (CLIP) for zero-shot text classification while their model focus on the natural language understanding models.

5.2 Application of CLIP

CLIP (Radford et al., 2021), a powerful text-image cross-modality pre-trained model, has shown strong zero-shot capability on various downstream tasks. Gu et al. (2021a) apply CLIP to perform open-vocabulary object detection by detecting objects described by arbitrary text inputs rather than in the pre-defined categories. Portillo-Quintero et al. (2021) use CLIP for zero-shot video retrieval. Song et al. (2022) provide a comprehensive investigation on applying CLIP to zero-shot visual question answering and visual entailment. Subramanian et al. (2022) present a strong zero-shot baseline for referring expression comprehension. Su et al. (2022) combine CLIP and off-the-shelf language model for image-grounded text generation, which achieves promising performance. In contrast, our work investigate CLIP into zero-shot text classification and show knowledge from CLIP can benefit language task while their work mainly focusing on zero-shot computer vision or vision-and-language tasks. To the best of our knowledge, we are the first to explore CLIP for zero-shot text classification task.

6 Conclusion

In this work, we studied how to transfer knowledge from CLIP into zero-shot text classification. To this end, we introduced a novel paradigm, CLIPTEXT and PROMPT-CLIPTEXT, for zero-shot text classification by reformulating it as a text-image matching problem. Experimental results demonstrated that CLIP can be a good zero-shot learner for text classification. To the best of our knowledge, this is the first work to apply CLIP for zero-shot text classification task. We hope that our work will motivate further research on transferring knowledge from VL model (e.g., CLIP) to language tasks.

Limitations

We present some limitations of our approach, which can be investigated in the future: (1) Currently, our approaches need to manually choose image for each text label, which may cause the model to be sensitive to the images selected. Though the ensemble method can alleviate this problem to some extent, how to automatically map the text label into the corresponding image is an interesting research question to investigate. (2) Since CLIP was pre-trained on noisy web-crawled data on the Internet, our approaches are limited by pre-training data distribution of CLIP. Therefore, a potential future direction is to further pre-train CLIP on more general downstream task datasets.

Acknowledgements

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 62236004 and 61976072.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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. *arXiv preprint arXiv:2010.11929*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021a. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021b. Zero-shot detection via vision and language knowledge distillation. *arXiv e-prints*, pages arXiv–2104.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Tingting Ma, Jin-Ge Yao, Chin-Yew Lin, and Tiejun Zhao. 2021. Issues with entailment-based zero-shot text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 786–796, Online. Association for Computational Linguistics.

Stephen Mayhew, Tatiana Tsygankova, Francesca Marini, Zihan Wang, Jane Lee, Xiaodong Yu, Xingyu Fu, Weijia Shi, Zian Zhao, and Wenpeng Yin. 2019. Karthikeyan k, jamaal hay, michael shur, jennifer sheffield, and dan roth. *University of pennsylvania lorehlt*.

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.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marín. 2021. A straightforward framework for video retrieval using clip. In *Mexican Conference on Pattern Recognition*, pages 3–12. Springer.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Kunxun Qi, Hai Wan, Jianfeng Du, and Haolan Chen. 2022. Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1910–1923, Dublin, Ireland. Association for Computational Linguistics.

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. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

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

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

Haoyu Song, Li Dong, Wei-Nan Zhang, Ting Liu, and Furu Wei. 2022. Clip models are few-shot learners: Empirical studies on vqa and visual entailment. *arXiv preprint arXiv:2203.07190*.

Yixuan Su, Tian Lan, Yahui Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. 2022. Language models can see: Plugging visual controls in text generation. *arXiv preprint arXiv:2205.02655*.

Sanjay Subramanian, Will Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. 2022. Reclip: A strong zero-shot baseline for referring expression comprehension. *arXiv preprint arXiv:2204.05991*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

☑ A2. Did you discuss any potential risks of your work?
Section Limitation

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

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

B ☑ Did you use or create scientific artifacts?

Section 4 Experiments

☑ B1. Did you cite the creators of artifacts you used?
Section 4 Experiments

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4 Experiments

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 Experiments

☒ 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?
All the data we used are publicly and used safely by previous works.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We will provide the documentation of our code and pre-trained models in our code repo.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4 Experiments

C ☑ Did you run computational experiments?

Section 4 Experiments

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 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?
Section 3 Model and 4 Experiments

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 Experiments

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4 Experiments

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.