Iterative Nearest Neighbour Machine Translation for Unsupervised Domain Adaptation

Hui Huang[1]†, **Shuangzhi Wu**[2], **Xinnian Liang**[3]†, **Zefan Zhou**[4]†,
Muyun Yang[1]‡, **Tiejun Zhao**[1]
[1]Faculty of Computing, Harbin Institute of Technology, Harbin, China
[2]ByteDance AI Lab, Beijing, China
[3]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[4]School of Computer Science and Engineering, Northeastern University, Shenyang, China
22b903058@stu.hit.edu.cn, yangmuyun@hit.edu.cn, tjzhao@hit.edu.cn,
{wufurui, liangxinnian, zhouzefan.zzf}@bytedance.com,

Abstract

Unsupervised domain adaptation of machine translation, which adapts a pre-trained translation model to a specific domain without in-domain parallel data, has drawn extensive attention in recent years. However, most existing methods focus on the fine-tuning based techniques, which is non-extensible. In this paper, we propose a new method to perform unsupervised domain adaptation in a non-parametric manner. Employing only in-domain monolingual data, this method jointly perform nearest neighbour inference on both forward and backward translation directions. The forward translation model creates nearest neighbour datastore for the backward direction, and vice versa, strengthening each other in an iterative style. Experiments on multi-domain datasets demonstrate that our method significantly improves the in-domain translation performance and achieves state-of-the-art results among non-parametric methods.

1 Introduction

Neural machine translation (NMT) has demonstrated impressive performance when trained on large-scale corpora. However, despite the abundance of general-domain parallel data, domain-specific parallel data is not readily available (Chu and Wang, 2018). Therefore, how to adapt a general-domain NMT model via in-domain monolingual data has become the research focus in the community. Since no annotated data is involved, this effort is also generally known as the unsupervised domain adaptation of NMT.

Existing approaches are mostly focused on the data selection and finetuning techniques (Pourdamghani et al., 2019; Aharoni and Goldberg, 2020a; Hu et al., 2019; Zhang et al., 2022). For example, Hu et al. (2019) proposed to induce in-domain lexicon pairs as synthetic data for fine-tuning. Zhang et al. (2022) proposed to use constrained back-translation model to generate synthetic in-domain data for fine-tuning. Despite the progress they have made, the cumbersome fine-tuning process would lead to catastrophic forgetting (Thompson et al., 2019) and decrease the performance on general domain. Besides, a copy of parameters is required for each domain, which is not flexible facing multi-domain scenario. Therefore, Zheng et al. (2021) propose to perform adaptation based on non-parametric nearest neighbour inference, and they introduce an autoencoder task based on target language to enable in-domain data construction with monolingual data. However, adapter layers are still required to be fine-tuned in their method, which is not fully non-parametric.

In this work, we propose an iterative nearest neighbour approach named Iter-*k*NNMT to achieve fully non-parametric unsupervised domain adaptation. Our framework is built based on the recently proposed *k*NN-MT (Khandelwal et al., 2021). We employ two pre-trained general-domain NMT models in both forward and backward directions, and the datastores are constructed by the model in the reverse direction for each other. The forward model performs nearest neighbour inference to the source language sentences, and the results serve as the datastore for the backward model. Then, the backward model performs nearest neighbour inference to the target sentences and generate better datastore for the forward model. This process is iteratively performed, making the most of monolingual data for non-parametric inference.

We evaluate the proposed Iter-*k*NNMT on multi-datasets, including IT, Medical, Law and Koran

† Contribution during internship at ByteDance Inc.

‡ Corresponding Authors.

Findings of the Association for Computational Linguistics: ACL 2023, pages 13294–13301
July 9-14, 2023 ©2023 Association for Computational Linguistics

domains. Experimental results show that, without introducing any extra parameters, we are able to achieve 6 BLEU improvement on in-domain translation, bringing a new state-of-the-art results among non-parametric adaptation methods.

2 Approach

2.1 Preliminary: *k*NN-MT

*k*NN-MT can be formulated as the following two steps, namely datastore construction and nearest neighbour decoding.

2.1.1 Datastore Construction

Given a pretrained NMT model and an in-domain parallel corpus $(\mathcal{X}, \mathcal{Y})$, *k*NN-MT first constructs a key-value datastore as follows:

$$\mathcal{D}(\mathcal{X}, \mathcal{Y}) = \bigcup_{(x,y)\in(\mathcal{X},\mathcal{Y})} \{(f(x, y_{<i}), y_i), \forall y_i \in y\}$$

where the keys are the mapping representations of all the translation contexts in the training set using the model representation $f(\cdot)$, and the values are corresponding ground-truth tokens, and (x, y) is a parallel sentence pair.

2.1.2 Nearest Neighbour Decoding

During inference, on each step i, *k*NN-MT models the decoding probability $P_{k\text{NN}}(\hat{y}_i|x, \hat{y}_{<i})$ by measuring the distance between query $f(x, \hat{y}_{<t})$ and its k-nearest representations in $\mathcal{D}(\mathcal{X}, \mathcal{Y})$. Denote the retrieved neighbors as $\mathcal{N}^i = \{(h_j, y_j), j \in \{1, 2, ..., k\}\}$, and then a *k*NN distribution over vocabulary is computed as:

$$P_{k\text{NN}}(\hat{y}_i|x, \hat{y}_{<i}) \propto \sum_{(h_j,y_j)\in\mathcal{N}^i} \mathbb{I}_{\hat{y}_i=y_j} \exp\left(\frac{-d(h_j, f(x, \hat{y}_{<i}))}{\tau}\right)$$

where τ is the temperature, and $d(\cdot, \cdot)$ is the L2 distance function. The final probability for the next token is the interpolation of $P_{\text{NMT}}(y_i|x, y_{<i})$ and $P_{k\text{NN}}(y_i|x, y_{<i})$ with a tunable weight λ:

$$P(\hat{y}_i|x, \hat{y}_{<i}) = (1 - \lambda)P_{\text{NMT}}(\hat{y}_i|x, \hat{y}_{<i})+ \lambda P_{k\text{NN}}(\hat{y}_i|x, \hat{y}_{<i})$$

2.2 *k*NN-MT with monolingual data

An effective method to improve domain-specific machine translation with monolingual data is to augment the parallel training corpus with back-translations of target language sentences (Sennrich et al., 2016). In the case of *k*NN-MT, the utilization of monolingual data can follow this style. Specially, given a set of sentences $\mathcal{Y}$ in target language, a pre-constructed NMT model is used to automatically generate their translations $\tilde{\mathcal{X}}$ in source language. Then the datastore can be created based on the synthetic data $(\tilde{\mathcal{X}}, \mathcal{Y})$:

$$\mathcal{D}(\tilde{\mathcal{X}}, \mathcal{Y}) = \bigcup_{(\tilde{x},y)\in(\tilde{\mathcal{X}},\mathcal{Y})} \{(f(\tilde{x}, y_{<i}), y_i), \forall y_i \in y\}$$

This datastore can then be retrieved to interpolate the prediction of forward translation model. Although the source language sentences $\tilde{\mathcal{X}}$ are synthetic, the target sentences $\mathcal{Y}$ are fluent and intact. Therefore, when performing nearest neighbour retrieval, the best tokens can be retrieved largely based on the fitness to target context. As discussed in previous research (Edunov et al., 2018), target fluency is one of the major factors to hinder the performance on domain-specific translation. Therefore, the interpolated probability distribution would be more inclined to the target-domain.

2.3 Iter-*k*NNMT

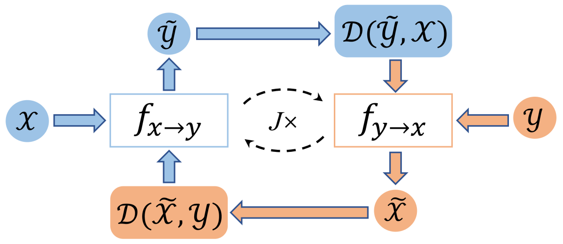
$\tilde{\mathcal{Y}}$
$\mathcal{D}(\tilde{\mathcal{Y}}, \mathcal{X})$
$\mathcal{X}$
$f_{\mathcal{X}\to\mathcal{Y}}$
$J\times$
$f_{\mathcal{Y}\to\mathcal{X}}$
$\mathcal{Y}$
$\mathcal{D}(\tilde{\mathcal{X}}, \mathcal{Y})$
$\tilde{\mathcal{X}}$

Figure 1: Iterative nearest neighbour inference process.

In most applications, if the target monolingual data $\mathcal{Y}$ is accessible, a monolingual source data $\mathcal{X}$ would be accessible, too. To make the most of monolingual data for nearest neighbour inference, we extend the task setting from solely improving the forward NMT model augmented with target monolingual data into a paired one.

As shown in Figure 1, our method runs nearest neighbour inference bidirectionally and refines the datastore iteratively. At each iteration step j:

Method	Data	EN-DE					DE-EN				
		IT	Medical	Law	Koran	Avg.	IT	Medical	Law	Koran	Avg.
basic NMT	-	38.35	39.99	45.48	16.26	35.02	30.69	36.55	41.07	15.56	30.97
Copy-*k*NNMT	half	38.42	39.52	45.18	15.29	34.60	30.87	36.35	40.74	15.55	30.88
BT-*k*NNMT	half	39.82	45.38	51.98	18.96	39.04	31.89	40.56	45.62	20.74	34.70
UDA-*k*NNMT	half	$40.62^{\dagger}$	$44.56^{\dagger}$	$51.32^{\dagger}$	$19.41^{\dagger}$	$38.98^{\dagger}$	$31.95^{\dagger}$	$39.60^{\dagger}$	$45.17^{\dagger}$	$19.48^{\dagger}$	$34.05^{\dagger}$
UDA-*k*NNMT	all	$41.57^{\ddagger}$	$46.64^{\ddagger}$	$52.02^{\ddagger}$	$19.42^{\ddagger}$	$39.91^{\ddagger}$	$33.99^{\dagger}$	$40.75^{\dagger}$	$46.88^{\dagger}$	$20.59^{\dagger}$	$35.55^{\dagger}$
Iter-*k*NNMT	half	40.90	48.06	54.97	20.03	40.99	33.07	42.70	48.11	22.12	36.50
Parallel-*k*NNMT	half	41.33	50.18	56.73	18.46	41.68	34.22	45.27	50.68	22.37	38.14

Table 1: BLEU score of different unsupervised domain-adaptation methods on the four domains. Results with † are re-implemented by us with their released codes and ‡ are taken from their paper.

1) Model $f_{x \rightarrow y}$ performs *k*NN inference on datastore $\mathcal{D}_j(\tilde{\mathcal{X}}, \mathcal{Y})$ to decode the monolingual data $\mathcal{X}$ into $\tilde{\mathcal{Y}}_{j+1}$, which is combined into the datastore $\mathcal{D}_{j+1}(\tilde{\mathcal{Y}}, \mathcal{X})$ for backward translation. 2) Then model $f_{y \rightarrow x}$ performs *k*NN inference on $\mathcal{D}_{j+1}(\tilde{\mathcal{Y}}, \mathcal{X})$ to decode the monolingual data $\mathcal{Y}$ into $\tilde{\mathcal{X}}_{j+1}$, forming datastore $\mathcal{D}_{j+1}(\tilde{\mathcal{X}}, \mathcal{Y})$ for the next iteration step. The newly generated datastore would contain more diverse and fluent source context, serving as a better memory base for the next *k*NN retrieval. Notice no further parameter is introduced during the whole procedure, only the datastore is updated, therefore our method is totally non-parametric.

3 Experiments

3.1 Setup

We use the same multi-domain dataset as Aharoni and Goldberg (2020b) to evaluate the effectiveness of our proposed method. We mainly experiment on the adaptation of four domains of IT, Medical, Law and Koran. To exclude the influence of parallel sentence pairs, we divide the training set into two halves, and fetch the source side of first half and target side of second half, forming two unaligned monolingual in-domain datasets $\mathcal{X}$ and $\mathcal{Y}$, and dev and test set are kept unchanged.

The WMT19 German-English News translation task winner model (Ng et al., 2019) is chosen as our general domain model, and the same setting is applied to train the English-German model[1]. Faiss[2] is used to build the in-domain datastore to carry out fast nearest neighbor search, and 4096 cluster centroids are learned for each domain. We set the hyper-parameter τ as 4 for IT, Medical, Law, and 40 for Koran. The λ is tuned on the in-domain dev sets for different methods, and we use $(4, 8, 16, 24, 32)$ as the value for k.

To avoid the influence of random data partition, all results are the average of 5 runs with different random seeds when partitioning the data.

3.2 Baselines

We mainly compare with the following methods for translation domain-adaptation:

- **Basic NMT** The general-domain model is directly evaluated on the target domain;
- **Copy-*k*NNMT** In-domain datastore is created based on $(\mathcal{Y}, \mathcal{Y})$ for *k*NN-MT inference;
- **BT-*k*NNMT** In-domain datastore is created based on back-translated data $(\hat{\mathcal{X}}, \mathcal{Y})$ for *k*NN-MT inference;
- **Parallel-*k*NNMT** Ground-truth in-domain parallel data $(\mathcal{X}, \mathcal{Y})$ is used to generate the datastore, which can be regarded as the upper bound of the *k*NN retrieval based methods;
- **UDA-*k*NNMT** (Zheng et al., 2021) This method first creates datastore based on $(\mathcal{Y}, \mathcal{Y})$, and then introduces lightweight adapters to map the token-level representation to the ideal representation of translation task. Notice due to the introduction of adapter layers, this method is not fully non-parametric.

3.3 Main Results

As shown in Table 1, our methods can surpass all baselines and achieve the state-of-the-art among non-parametric methods. Especially, we are able to surpass the result of UDA-*k*NN by 1 BLEU

[1] We do not introduce monolingual data as augmentation, only following the setting of their basic model.

[2] https://github.com/facebookresearch/faiss

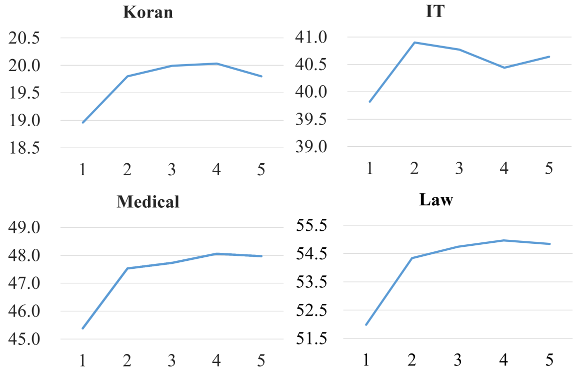

Figure 2: The variation of BLEU score of Iter-*k*NNMT according to the iteration number on EN-DE.

score without the introduction of an extra training phase. Copy-*k*NNMT could hardly bring any improvement. BT-*k*NNMT, which is actually our method without iteration, could introduce notable improvement, but still underperforms our method. We attribute this to the iteration process, where domain-specific knowledge is written into datastore and further refined during the iteration. While not introducing any extra training parameters, the datastore itself is able to memorize and update the domain-related lexical and syntactic knowledge, and this can be actually deemed as a non-parametric learning process.

We also illustrate the variation of BLEU score according the iteration number on EN-DE direction. As shown in Figure 2, the BLEU score increases rapidly in the first two iterations, but the improvement would be marginal or even negative in the following iterations. Therefore, we set the max iteration number as 5 in all experiments.

4 Ablation Studies

4.1 The influence of datastore size

In this section, we want to investigate the influence of the datastore size. To this end, we split the in-domain data into different folds, and fetch two unaligned folds for datastore building.

As shown in Figure 3, on EN-DE direction, our Iter-*k*NNMT can bring consistent improvement and surpass UDA-*k*NNMT among different data scales. Despite the monolingual data being limited, we are able to create various key-value pairs with different *k*NNMT models in different iterations. With different back translations for a single target sequence, the most suitable key-value pair would have more possibility to be retrieved.

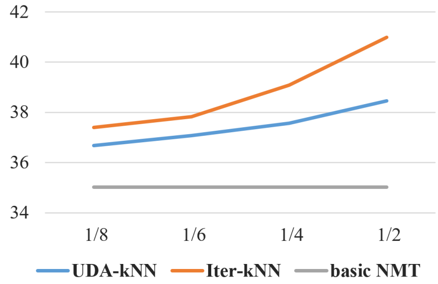

Figure 3: The variation of BLEU score according to different datastore sizes on EN-DE.

4.2 Refinement and Accumulation

To verify that the improvement comes from both the refinement during iteration and the accumulation of datastore, we perform two contrast experiments on EN-DE direction. Firstly, we perform n-best decoding with random sampling on the back translation model, and generate 5 different source sentences for each target sequence, forming an accumulated datastore without refinement. Secondly, we perform iteration without accumulation, only using the newest datastore each time.

Model	Data	Koran	Medical
basic NMT	-	16.26	39.99
UDA-*k*NNMT	full	19.42	46.64
Iter-*k*NNMT	half	20.03	48.06
-accumulation	half	19.22	45.79
-refinement	half	19.79	46.20

Table 2: BLEU scores on EN-DE Koran and Medical domains without refinement or accumulation.

As can be seen in Table 2, both refinement and accumulation play an important role in the Iter-*k*NNMT. While datastore is indeed refined during the iteration, accumulation is necessary to keep the variety and robustness of datastore. While the refined back translation can function as more accurate datastore, the comparably noisy back translation can also improve the robustness of the retrieval. Also, if there is no refinement from iteration, a single back-translation model only has limited decoding space induced by pre-train data, therefore the constructed datastore can also provide limited domain-specific guidance.

5 Conclusion

This paper proposes a simple yet effective method to perform unsupervised domain adaptation of machine translation in a non-parametric manner. We perform nearest neighbour inference on both forward and backward directions, strengthening each other in an iterative manner. While accumulated datastore is more robust and effective than datastore generated in a single pass, the accumulated datastore introduce extra retrieval overhead. In the future, we would investigate how to compress the datastore and improve the decoding efficiency.

Limitations

Due to the lack of research in this area, there is only one direct related paper to our work, which serves as the main baseline in our experiments. We hope we can compare our method with more related works to verify its effectiveness in the future. Also, the domain adaptation problem not only exits in the machine translation filed, but also various generation and understanding NLP tasks, where we should evaluate our method on if we are not limited by time and resource.

Acknowledgements

This work is supported by National Key RD Program of China (2020AAA0108000), National Natural Science Foundation of China (62276077, U1908216), Key RD Program of Yunnan (202203AA080004) and Shenzhen College Stability Support Plan (No. GXWD20220811170358002).

References

Roee Aharoni and Yoav Goldberg. 2020a. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Roee Aharoni and Yoav Goldberg. 2020b. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Nima Pourdamghani, Nada Aldarrab, Marjan Ghazvininejad, Kevin Knight, and Jonathan May. 2019. Translating translationese: A two-step approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3057–3062, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. 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 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

Hongxiao Zhang, Hui Huang, Jiale Gao, Yufeng Chen, Jinan Xu, and Jian Liu. 2022. Iterative constrained back-translation for unsupervised domain adaptation of machine translation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5054–5065, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xin Zheng, Zhirui Zhang, Shujian Huang, Boxing Chen, Jun Xie, Weihua Luo, and Jiajun Chen. 2021. Non-parametric unsupervised domain adaptation for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4234–4241, Punta Cana, Dominican Republic. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

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

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

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

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
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?

Left blank.

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

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?
Left blank.

☑ 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?
Left blank.

☑ 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.)?
Left blank.

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.