Uncovering Hidden Consequences of Pre-training Objectives in Sequence-to-Sequence Models

Tannon Kew[1] and **Rico Sennrich**[1,2]
[1]Department of Computational Linguistics, University of Zurich
[2]School of Informatics, University of Edinburgh
{kew,sennrich}@cl.uzh.ch

Abstract

Some variants of self-supervised denoising objectives for pre-training encoder-decoder language models have been reported to have a negligible impact on downstream performance. Yet the design of these pre-training objectives leads to behavioural differences that can be uncovered with specific manipulations. We reproduce a recently proposed zero-shot control method and find that it is only successful on a subset of models. To understand what causes the difference in its effectiveness, we perform a set of controlled experiments, varying only the pre-training objective, and find unexpected interactions between the pre-training method and downstream controllability of models after fine-tuning. Our results show that different pre-training objectives have consequences that may not be visible in standard downstream evaluation, but which should be taken into account when developing models with controllability in mind.

1 Introduction

Self-supervised denoising objectives have proven extremely powerful for deriving transformer-based pre-trained language models (PLMs) given massive amounts of unlabelled data. These objectives are typically agnostic towards specific downstream tasks and thus do not resemble real-world use cases. Instead, they enable the model to learn optimal parameter initialisations for subsequent fine-tuning on various downstream tasks (Dai and Le, 2015; Erhan et al., 2010). During fine-tuning, the PLM quickly learns new tasks based on the supervised signal provided, rendering pre-training task largely redundant.

Previous work has found performance differences on downstream tasks to be negligible given various denoising pre-training objectives (Lewis et al., 2020; Alajrami and Aletras, 2022).[1] As a result, the choice of which method to apply in pre-training has largely been based on factors such as efficiency (e.g. Raffel et al., 2020; Song et al., 2019). However, given equally well performing pre-training objectives, we find that encoder-decoder PLMs respond drastically differently to post-hoc manipulations after fine-tuning.

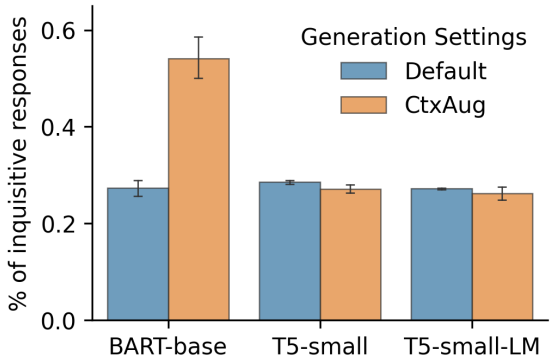

Figure 1: The effect of CtxAug for inquisitive dialogue modelling with off-the-shelf models. In contrast to BART, T5 models exhibit a minimal response to the context code. T5-small-LM refers to the LM-adapted model from Lester et al. (2021a).

Specifically, we investigate the use of context augmentation (CtxAug), proposed by Hazarika et al. (2022), as a zero-shot control method designed to steer a fine-tuned encoder-decoder model towards generating outputs with particular attributes. While they introduce this as a general control mechanism for encoder-decoder transformers, our experiments with BART (Lewis et al., 2020) and two variants of T5 (Raffel et al., 2020; Lester et al., 2021a) show that controllability via context augmentation is predominantly exhibited by BART (Figure 1).

Given this observation, we hypothesise that the success of this zero-shot control method may be highly dependent on a model's pre-training objective. To investigate this hypothesis, we set out to identify exactly what aspects of BART's pre-training allow for CtxAug to work. Our findings

[1]We confirm these findings with our own models in Appendix C.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7010–7022
July 9-14, 2023

suggest that fine-tuned models are capable of exhibiting *vestigial behaviours*[2] which are endowed by their pre-training objectives and allow for interesting and useful post-hoc manipulation methods in downstream applications.

2 Background

2.1 Seq2Seq Pre-training Objectives

To jointly pre-train an encoder-decoder transformer (Vaswani et al., 2017), seq2seq pre-training objectives typically corrupt an input sequence (*noise*) before feeding it to the model and then train the model to recover the original sequence (*denoise*). Usually, this involves span-based masked language modelling (MLM) (Joshi et al., 2020; Devlin et al., 2019a) combined with a standard language modelling objective involving left-to-right prediction (Bengio et al., 2003; Radford et al., 2018). However, popular denoising objectives differ in terms of the extent of corruption applied and the amount that needs to be recovered. For instance, **MASS** (Song et al., 2019) applies MLM to a *single*, randomly selected span of contiguous source tokens and predicts only the noised tokens given their positional information. **T5** (Raffel et al., 2020) randomly selects *multiple* token spans and replaces *each span* with a single unique 'sentinel' mask token. The target sequence then corresponds to a stilted sequence consisting of the masked input spans separated by their respective sentinel tokens. **BART** (Lewis et al., 2020) applies span-based MLM in conjunction with sentence permutation. In stark contrast to the previous approaches, BART is tasked with reconstructing the input sequence *in full* and not just the masked spans, which we refer to as partial reconstruction.

2.2 Context Augmentation for Zero-shot Control

Despite strong generalisation abilities of fine-tuned PLMs, controlling for desirable attributes in generated text remains an active area of research (e.g. Dathathri et al., 2019; Liu et al., 2021; Yang and Klein, 2021; Krause et al., 2021; Pascual et al., 2021) Recently, Hazarika et al. (2022) proposed CtxAug as a means of controlling fine-tuned encoder-decoder LMs in a zero-shot setting. Given an encoder-decoder transformer trained on a downstream task, CtxAug aims to provide additional conditioning context, not included in the original source sequence, to guide the model generation towards a particular attribute. CtxAug encodes a set of phrases or sentences that exhibit a target attribute into an averaged representation $\mathbb{C}$, which is concatenated with the hidden representation of the original source sequence: $\mathbb{C} \oplus enc(x)$. The decoder can then attend to this augmented input context at inference time without any updates to the model's parameters. To ensure that the model does not simply disregard the context code, the authors also propose to manually re-weight the model's cross attention with an attention biasing parameter. In experiments on dialogue modelling, Hazarika et al. (2022) demonstrate that CtxAug can be used to encourage more inquisitive and positive sentiment responses.

3 Experimental Setup

3.1 Pre-training

To investigate the effect of different encoder-decoder pre-training objectives on CtxAug, we use a controlled setup on scaled-down models and datasets, where only the pre-training objective differs. Specifically, we compare the following objectives (depicted in Table 3):

i) **MLM+PS**: span-based MLM combined with sentence permutation (i.e. BART's default pre-training objective);

ii) **MLM**: span-based MLM alone;

iii) **PS**: sentence permutation alone;

iv) **SI$_{\text{PR-MS}}$**: MASS-style span-infilling with partial reconstruction[3];

v) **SI$_{\text{PR-T5}}$**: T5-style span-infilling with running partial reconstruction and sentinel tokens;

vi) **SI$_{\text{FR}}$**: span-infilling with full reconstruction of the input sequence.

Since methods differ in their original works in terms of how spans are selected for masking, we

[2]While there is a substantial body on catastrophic forgetting, where information relevant for a learned task is lost upon training on a new task (McCloskey and Cohen, 1989; Goodfellow et al., 2014), we use *vestigial behaviour* to refer to observable properties that remain after fine-tuning and can be traced back to earlier (pre-)training tasks, in analogy to vestigial structures in biology.

[3]For consistency, our SI$_{\text{PR-MS}}$ differs from the original MASS objective in that we select multiple spans for masking in a given input, while (Song et al., 2019) only select a single span per training example, and we do not perform any random mask replacement.

		Single Objectives (§4.1)							Mixed Objectives (§4.3)		
		No PT	MLM+PS	MLM	PS	SI_{FR}	$SI_{PR\text{-}MS}$	$SI_{PR\text{-}T5}$	$SI_{FR/PR}$ 1:3	$SI_{FR/PR}$ 1:1	$SI_{FR/PR}$ 3:1
inquisitive	default	54.18	35.24	50.39	40.61	50.79	44.87	54.04	47.90	57.84	50.80
	CtxAug	-8.27	**+9.68**	+5.42	-0.20	**+6.37**	**-10.90**	-7.07	+2.42	**+2.82**	+5.51
positive	default	29.24	39.19	29.17	34.52	31.46	35.65	34.00	31.46	35.65	34.00
	CtxAug	**+11.99**	**+7.12**	+5.11	**+15.33**	**+6.71**	**+6.47**	**+13.93**	**+6.71**	**+6.47**	**+13.93**

Table 1: Portion of inquisitive (top) and positive sentiment (bottom) dialogue responses generated under default generation settings (in grey) and the absolute increase/decrease with CtxAug using the appropriate context code. All results are the aggregate of multiple seeded runs. Scores in bold indicate statistically significant differences from the default generations within *all* seeded runs according to a two-tailed unpaired t-test ($p < 0.01$).

unify these based on the approach taken by Lewis et al. (2020) and use a Poisson distribution ($\lambda = 3$).[4] For reference, we also compare to a non-pre-trained (**No PT**) baseline, which is trained from scratch on the downstream task.

Model We use the BART model architecture, which resembles a standard encoder-decoder transformer with GeLU activation functions. Following Dufter and Schütze (2020) we scale the model down by dividing the size of the hidden layer, intermediate feed forward layers, and the number of attention heads by 12. This results in a hidden size of 64 and intermediate size of 256 and a single attention head.

Data As pre-training data we select the BookCorpus[5] (Zhu et al., 2015; Bandy and Vincent, 2021) due to its stylistic similarities to our downstream task (e.g. dialogues between characters). We perform simple preprocessing, removing preambles and meta data by filtering lines without sentence-final punctuation or lines containing more than 70% punctuation or numbers. We set aside 100 randomly selected books for validation. The resulting corpus contains approximately 72M and 400k sentences for training and validation, respectively. Given our budgeted training setup, the model only sees approximately 65% of the data before reaching the maximum number of update steps. Finally, we train our own BART tokenizer on the training split with a maximum vocabulary size of 4,096.

[4]Here, 0-length spans, which correspond to insertions in the original BART denoising objective are ignored. And contiguous independently masked spans are merged to ensure the we do not have consecutive `[M]` tokens in the input sequence.

[5]We use a version created in September, 2020 (https://github.com/soskek/bookcorpus).

3.2 Fine-tuning & Inference

To measure the impact of CtxAug for zero-shot controlled generation, we follow the experimental setup from Hazarika et al. (2022) and focus on promoting inquisitive and positive responses in knowledge-grounded dialogue generation with the Topical-Chat dataset (Gopalakrishnan et al., 2019). The task is to generate the target dialogue turn given a relevant knowledge snippet k and the dialogue history h^T, where T is the number of turns.

At inference time, we use top-p sampling (p=0.9) with beam size of 4 and a temperature of 0.7. Sequences are generated with a maximum length of 40 tokens. For all experiments, we pre-train and fine-tune with 3 different seeds before performing inference with 5 different seeds. This results in a total of 15 inference runs for each model. To promote inquisitiveness with CtxAug we randomly sample 10 questions from the training data to construct the control code. To promote positive sentiment, we use a limited set of only 5 short phrases. Fine-tuning and inference experiments are performed with Hugging Face's Transformers library (Wolf et al., 2020). We include the full details on training and inference hyperparameters in Appendix A.[6]

4 Results

4.1 Pre-training Objectives for CtxAug

Table 1 shows the effectiveness of CtxAug given the different pre-training objectives considered. For promoting inquisitive responses (top row), BART's original denoising objective (MLM+PS) exhibits the strongest positive response to CtxAug over the default generation setting. Meanwhile, isolating the two independent noising operations used in this objective reveals that sentence permutation (PS)

[6]We make our code available at https://github.com/ZurichNLP/understanding-ctx-aug.

alone is insufficient for CtxAug to succeed. Comparing span-infilling pre-training objectives (SI*), we can observe that the format of the target sequence used during pre-training is crucial. With noising operations being equal, CtxAug for inquisitive responses works effectively only when the model is pre-trained to reconstruct the target sequence in full, while partial reconstruction yields similar results to that of no pre-training (No PT). In contrast, encouraging more positive responses with CtxAug (bottom row) succeeds regardless of the pre-training strategy[7], and even without any pre-training.

This suggests that multiple factors may contribute to the overall effectiveness of CtxAug in practice. Firstly, the fact that models trained from scratch can still leverage CtxAug for positive sentiment suggests that there may be effects arising from correlation of source and target attribute features in the fine-tuning data. In such a case, CtxAug may not generalise to other datasets and tasks. Secondly, and most notably, full reconstruction pre-training objectives support CtxAug more than partial reconstruction objectives.

Reconstructing the corrupted input sequence in full naturally encourages a strong correlation between input and target attributes. This more closely resembles the central mechanism in CtxAug where a vector representing the desired target attribute is 'reconstructed' in the target sequence. Meanwhile, partial reconstruction objectives yield primarily disjointed source and target sequences. This does not necessarily preclude the possibility of inferring relationships between co-occuring attributes over long distances (e.g., sentence-initial subject-verb inversion together with a sentence-final question mark). However, the likelihood of successfully learning these becomes plausible only in scenarios where some co-occurring features remain unmasked and others are reconstructed. This limits the efficacy of CtxAug for promoting inquisitiveness, and possibly other attributes that occur over longer distances, to certain pre-training methods.

4.2 Duration of Fine-tuning on CtxAug

To investigate how CtxAug is impacted by the duration of fine-tuning, we conduct an ablation study in which we perform inference at regular intervals throughout fine-tuning. Figure 2 depicts how

[7]Appendix B shows that this also holds with publicly available models.

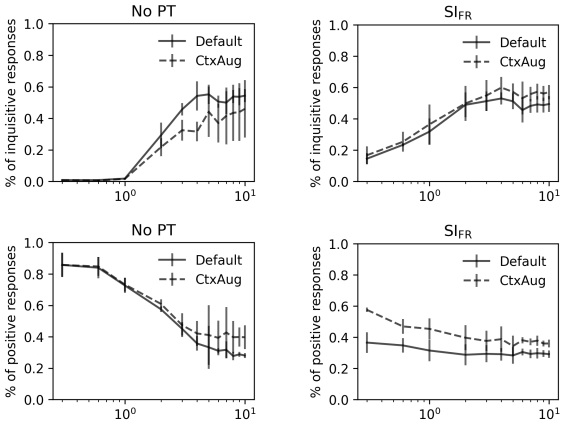

Figure 2: Effect of CtxAug throughout fine-tuning given different pre-training strategies. X-axis values indicate the number of training epochs and are shown on the log scale to better visualise the earliest stages of fine-tuning.

CtxAug behaves relative to the default generation setting as the model learns the downstream task. When starting from randomly initialised parameters, given question control phrases (top left), the model fails to leverage the control code effectively, resulting in degradation in inquisitiveness relative to the default generations settings. For positive sentiment (bottom left), however, we can observe that the fine-tuning data provides a sufficient signal to support CtxAug. In this setting the model starts to effectively make use of the control code after three epochs.

Meanwhile, the SI$_{FR}$ pre-trained model is able to leverage CtxAug at all stages of fine-tuning, highlighting the vestigial behaviour from pre-training. This is most visible when encouraging positive sentiment responses (bottom right), where, in the earliest stages of fine-tuning, we can observe a significant increase in the number of positive sentiment responses generated. As the model adapts to the task, this advantage tapers off, indicating that vestigial behaviours from pre-training weaken over time.

For inquisitive responses (top right), the effect of CtxAug is most noticeable after the first few fine-tuning epochs, suggesting that this type of pre-training objective endows the model with a useful bias that can be effectively exploited by CtxAug. We also note that while the effect is only slight under this condition, it reflects the model's overall tendency to generate responses pertaining to the target attributes in question. As the model learns the task, inquisitiveness naturally increases, while positiveness decreases. Manual inspection confirmed that at the earliest stages of training, models tended

to output generic and positive responses (e.g. "I know!"), which gradually become slightly more varied to include negative responses (e.g. "I don't know that.") and simple questions.

4.3 Mixing Pre-training Objectives

Any encumberment to leveraging interesting and useful post-hoc control techniques such as CtxAug with fine-tuned PLMs may be considered a significant downside of upstream decisions relating to the pre-training objective. Yet in order to scale models and training data, partial reconstruction objectives have been chosen due to their lower computational cost (Raffel et al., 2020). One possible option for striking a desirable balance between pre-training efficiency and downstream flexibility could be to combine different pre-training objectives either within a single pre-training scheme or as a secondary pre-training before fine-tuning (e.g. Lester et al., 2021b). To this end, we experiment with combining SI_{FR} and $\text{SI}_{\text{PR-T5}}$ within a single pre-training scheme, $\text{SI}_{\text{FR/PR}}$, and investigate various mixing ratios: 1:3, 1:1 and 3:1. Table 1 (right) shows that gradually increasing the degree to which the model is tasked with full reconstruction of the noised input improves the effectiveness of CtxAug but even at 75% adoption (3:1), it fails to reach equivalence with using only SI_{FR}.

5 Related Work

The study of PLMs, their abilities, properties and behaviours, occupies a significant space in today's NLP research (e.g. Rogers et al., 2020; Lialin et al., 2022; Clark et al., 2019). Numerous works have evaluated and compared downstream performance of seq2seq PLMs, covering a wide array of tasks including abstractive summarisation (Blekanov et al., 2022; Zhu et al., 2021; Tang et al., 2022; Fabbri et al., 2021), question answering (Luo et al., 2022), graph-to-text generation (Ribeiro et al., 2021), dialogue modelling (Shin et al., 2022) and text simplification (Štajner et al., 2022), among others. While such comparisons are useful for guiding researchers in selecting the right model for a task and can sometimes reveal interesting differences on certain task-specific data sets, they tend to neglect important differences between PLMs, such as the underlying model size or the type and amount of data used for pre-training. Thus, it remains difficult to explain exactly *why* a particular model performs better or worse on a given task.

Meanwhile, there is a growing body of literature aimed at explaining some of the interesting and often unexpected behaviours observed among large PLMs. In this area, multilinguality has been linked to the duration of fine-tuning (Dufter and Schütze, 2020), and the ability to perform in-context few-shot learning and zero-shot generalisation has been linked to multiple factors. These include model scale (Brown et al., 2020), the types and formatting of demonstrations (Min et al., 2022), memorisation of pre-training data (Xie et al., 2022) and its distributional properties (Chan et al., 2022). The selection of architecture and pre-training objectives have also been found to be influential (Wang et al., 2022). Our work falls into this category and aims to explain which aspects of seq2seq pre-training objectives contribute to the ability to exploit additional conditioning context provided at inference time.

6 Conclusions

As PLMs become increasingly commonplace, so too does the importance of understanding the potential downstream consequences of decisions relating to their design. Our experiments indicate that context augmentation, as a method for zero-shot controlled natural language generation, is susceptible to inductive biases learned in pre-training given different types of control codes. Based on this, we conclude that pre-training objectives that aim to reconstruct a noised input *in full*, similar to BART, are best suited to leverage this technique. Looking forward, we expect that even for seemingly equally effective pre-training objectives, we can identify differences in behaviour, e.g. applicability of control methods, that remain after fine-tuning. In searching for optimal pre-training strategies for PLMs, this opens another dimension that needs to be considered and better understood.

Acknowledgements

We kindly thank Fabian Aiolfi for fruitful discussions throughout this project, as well as the anonymous reviewers for their helpful feedback. This work was facilitated by the infrastructure services provided by S3IT, the Service and Support for Science IT team at the University of Zurich. Rico Sennrich is funded by the Swiss National Science Foundation (project MUTAMUR; no. 176727).

Limitations

Comparing downstream performance of pre-training objectives with large-scale models is prohibitively expensive. Because of this, we employ scaled-down models that closely resemble the architectures and training procedures of popular PLMs. In doing so, we assume that our findings are transferable to some larger publicly available models. As noted by Hazarika et al. (2022), CtxAug offers an interesting alternative to prompting generative LMs that are significantly smaller than those that typically exhibit few- and zero-shot capabilities (Brown et al., 2020). While we provide support for both Hazarika et al. (2022)'s claim and our assumption in preliminary and supplementary experiments with select PLMs (see Section 1 and Appendix B), these experiments are still performed on models of up to 140M parameters. Therefore, we stop short of concluding that our findings generalise to LLMs, which dwarf these models in comparison.

Additionally, the number and types of target attributes that a user may want to control for in various downstream text generation tasks are potentially endless. However, our study focuses on only two possible target attributes, namely, inquisitiveness and positive sentiment, for the task of conversational dialogue modelling. In this way, our work partially serves as a re-implementation and reproduction study, confirming the main findings from Hazarika et al. (2022), but also highlighting limitations.

References

Ahmed Alajrami and Nikolaos Aletras. 2022. How does the pre-training objective affect what large language models learn about linguistic properties? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–147, Dublin, Ireland. Association for Computational Linguistics.

John Bandy and Nicholas Vincent. 2021. Addressing "Documentation Debt" in Machine Learning: A Retrospective Datasheet for BookCorpus. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(null):1137–1155.

Ivan S. Blekanov, Nikita Tarasov, and Svetlana S. Bodrunova. 2022. Transformer-Based Abstractive Summarization for Reddit and Twitter: Single Posts vs. Comment Pools in Three Languages. *Future Internet*, 14(3):69.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. 2022. Data Distributional Properties Drive Emergent In-Context Learning in Transformers. ArXiv:2205.05055 [cs].

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, volume 28. Curran Associates, Inc.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *CoRR*, abs/1912.02164. ArXiv: 1912.02164.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2020. Identifying Elements Essential for BERT's Multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(19):625–660.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *2nd international conference on learning representations, ICLR 2014, banff, AB, canada, april 14-16, 2014, conference track proceedings*.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Interspeech 2019*, pages 1891–1895. ISCA.

Devamanyu Hazarika, Mahdi Namazifar, and Dilek Hakkani-Tür. 2022. Attention Biasing and Context Augmentation for Zero-Shot Control of Encoder-Decoder Transformers for Natural Language Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10738–10748.

Behnam Hedayatnia, Karthik Gopalakrishnan, Seokhwan Kim, Yang Liu, Mihail Eric, and Dilek Hakkani-Tur. 2020. Policy-driven neural response generation for knowledge-grounded dialog systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 412–421, Dublin, Ireland. Association for Computational Linguistics.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to Train BERT with an Academic Budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10644–10652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021b. The Power of Scale for Parameter-Efficient Prompt Tuning. ArXiv:2104.08691 [cs].

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, Online. Association for Computational Linguistics.

Vladislav Lialin, Kevin Zhao, Namrata Shivagunde, and Anna Rumshisky. 2022. Life after BERT: What do other muppets understand about language? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3180–3193, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. 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 6691–6706, Online. Association for Computational Linguistics.

Man Luo, Kazuma Hashimoto, Semih Yavuz, Zhiwei Liu, Chitta Baral, and Yingbo Zhou. 2022. Choose Your QA Model Wisely: A Systematic Study of Generative and Extractive Readers for Question Answering. In *Proceedings of the 1st Workshop on Semi-parametric Methods in NLP: Decoupling Logic from Knowledge*, pages 7–22, Dublin, Ireland and Online. Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The

sequential learning problem. volume 24 of *Psychology of learning and motivation*, pages 109–165. Academic Press.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? *arXiv:2202.12837 [cs]*. ArXiv: 2202.12837.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. ArXiv:1910.10683 [cs, stat].

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. 2022. Dialogue Summaries as Dialogue States (DS2), Template-Guided Summarization for Few-shot Dialogue State Tracking. ArXiv:2203.01552 [cs].

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *International conference on machine learning*, pages 5926–5936.

Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. CONFIT: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5657–5668, Seattle, United States. Association for Computational Linguistics.

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

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? ArXiv:2204.05832 [cs, stat].

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. *arXiv:2111.02080 [cs]*. ArXiv: 2111.02080.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Chenguang Zhu, Ziyi Yang, Robert Gmyr, Michael Zeng, and Xuedong Huang. 2021. Leveraging Lead Bias for Zero-shot Abstractive News Summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1462–1471, Virtual Event Canada. ACM.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A

Benchmarking Platform for Text Generation Models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100, Ann Arbor MI USA. ACM.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE international conference on computer vision (ICCV)*.

Sanja Štajner, Kim Cheng Sheang, and Horacio Saggion. 2022. Sentence Simplification Capabilities of Transfer-Based Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12172–12180.

A Training Details

A.1 Pre-training Hyperparameters

Our scaled down models have approximately 1M parameters and are pre-trained using the open-source Fairseq library (Ott et al., 2019). Following recommendations for budgeted pre-training by Izsak et al. (2021), we use a small batch size of 4,096 tokens and a triangular learning rate schedule which warms up for 2,500 steps and decays to zero with over 250k update steps. We also restrict the maximum sequence length to 256 which is sufficient for our downstream task of dialogue modelling. All other hyperparameters are kept the same as those used by Lewis et al. (2020). Our mini model pre-training takes approximately 6 hours on a single Nvidia K80 GPU (16GB memory).

A.2 Fine-tuning on Topical-Chat

Topical-Chat comprises conversational dialogues between pairs of crowd workers. The crowd workers were provided with reading sets containing different fun facts on eight different topics including sports, pop culture and politics as interesting discussion points. For each target dialogue turn in the dataset, it is assumed that the relevant knowledge snippet is provided as additional context based on previous work from Hedayatnia et al. (2020). Table 2 provides an overview of the dataset's splits.

Split	Items
Train	145,238
Valid	8,986
Test (freq.)	9,065
Test (rare)	9,075

Table 2: Number of items in Topical-Chat for knowledge-grounded dialogue generation.

To fine-tune on Topical-Chat, we followed the setup adopted by Hazarika et al. (2022). Specifically, the input sequence comprises a fixed number of 'bucketed' tokens. 32 tokens are reserved for the knowledge snippet and 25 tokens for each turn in the dialogue history. A <pad> token is used to fill empty positions within each bucket and individual text sequences are truncated if their length exceeds the allocated bucket size. Dialogue history turns are delimited with speaker identifier tokens and the entire input sequence is prepended with a <bos> token. The model is trained for a maximum of 10 epochs with an effective batch size of 20 and a learning rate of $6.25e-5$. The maximum target sequence length is set to 64. Fine-tuning on a single Nvidia K80 GPU (16GB memory) takes around 1.5 to 2.5 hours depending on the model size.

A.3 Inference on Topical-Chat

At inference time, we use the same hyperparameters for all models. Specifically, we use top-p sampling (p=0.9) with beam size of 4 and a temperature of 0.7. The maximum sequence length is set to 40 tokens. When applying CtxAug we manually re-weight the cross attention distribution using method described in Hazarika et al. (2022). Again, we used the recommend hyperparameter value of 5, which the authors found to provide a good balance between exhibiting the target attribute and maintaining fluency. To account for randomness, we run inference with multiple random seeds, which takes approximately 25 minutes for each experiment setting using a batch size of 120.

To construct the control code, we adopt the same methods as Hazarika et al. (2022). For inquisitiveness, we randomly sample 10 questions from the Topical-Chat training split. These 10 questions are then embedded once to construct the control code that is concatenated with every instance in the test set. Note that the sampling process is dependent on the random seed for each inference run. This means that each seeded inference setting uses a different set of questions to construct the control code. For positive sentiment, we always use the same five phrases defined by Hazarika et al. (2022): "That's awesome", "That's cool", "Oh that is great", "It's great to", "It's wonderful to". Since Hazarika et al. (2022) reported negligible differences between the different sampling strategies for finding control phrases, we refrained from doing

an extensive search over alternative methods and opted to use their recommended settings.

Our main experiments are reported on the Topical-Chat 'frequent' test set, however, we observed similar trends across the board when evaluating on the Topical-Chat 'rare' test set also.

B CtxAug for Positive Sentiment

Encouraging positive sentiment with CtxAug applied to our scaled down models proved successful for all models regardless of the pre-training strategy used. Figure 3 shows that this result also holds with much larger publicly available models, with all differences being statistically significant according to a two-tailed unpaired t-test ($p < 0.01$). Note that the weaker effect of CtxAug for positive sentiment compared to controlling for response inquisitiveness with BART-base agrees with the findings from Hazarika et al. (2022).

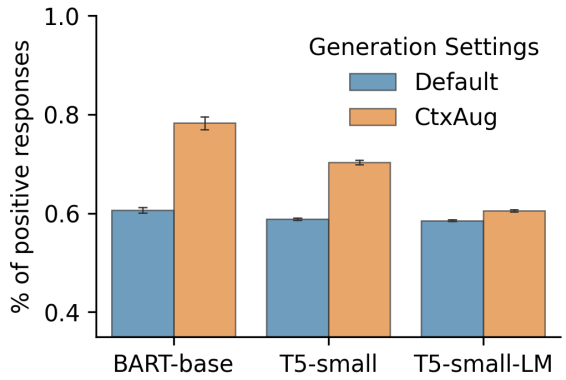

Figure 3: Performance of CtxAug with publicly available models when controlling for positive sentiment in Topical Chat.

C Performance Metrics

Inspecting the results of automatic metrics, we find only negligible differences on downstream performance across different denoising pre-training objectives, supporting previous findings (Lewis et al., 2020; Alajrami and Aletras, 2022; Raffel et al., 2020). Table 5 provides results for commonly used metrics for evaluating dialogue models. Specifically, we report the total number of unique responses generated (Uniq. Resp.), average response length (Resp. len.), perplexity (PPL) as computed by a distilled GPT-2 model[8], the portion of unique unigrams per response (Dist-1), Self-BLEU (Zhu et al., 2018), BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005). The latter three metrics are computed using ground-truth responses as references and are implemented in Hugging Face's Evaluate library[9]. Without pre-training, the difference in performance for all metrics is noticeable.

[8] https://huggingface.co/distilgpt2

[9] https://huggingface.co/docs/evaluate/index

Model	Noised Input	Target
MASS (Song et al., 2019)	I like [M] [M]. [M] [M] [M] in 1989.	[P] [P] The Simpsons [P] It was released [P] [P] [P]
T5 (Raffel et al., 2020)	I like [M1]. [M2] in 1989.	[M1] The Simpsons [M2] It was released [M]
BART (Lewis et al., 2020)	[M] in 1989. I like [postcards].	I like The Simpsons. It was released in 1989.
MLM+PS	[M] in 1989. I like [postcards].	I like The Simpsons. It was released in 1989.
MLM	I like postcards. [M] in 1989.	I like The Simpsons. It was released in 1989.
PS	It was released in 1989. I like The Simpsons.	I like The Simpsons. It was released in 1989.
$SI_{PR\text{-}MS}$	I like [M] [M]. [M] [M] [M] in 1989.	[P] [P] The Simpsons [P] It was released [P] [P] [P]
$SI_{PR\text{-}T5}$	I like [M1]. [M2] in 1989.	[M1] The Simpsons [M2] It was released [M]
SI_{FR}	I like [M]. [M] in 1989.	I like The Simpsons. It was released in 1989.

Table 3: General-purpose seq2seq denoising objectives used for pre-training. The bottom section depicts the pre-training objectives used in our experiments for comparison with those used in publicly available models. [M] and [P] indicate mask and pad tokens, respectively, while words appearing in square brackets indicate a token selected randomly from the vocabulary, following the 80/10/10 mask, replace, keep strategy used in the original MLM objective (Devlin et al., 2019b).

Knowledge snippet:	Daniel Radcliffe voiced the cartoon parody of Twilight's Edward Cullen on The Simpsons episode Treehouse of Horror XXI.
Speaker A:	Yep me either. I saw the 70's show was made in the UK and was cancelled after only 10 shows.
Speaker B:	Wow, I guess they didnt love it like people did here. Did you realize that in the first 400 episodes of the SImpsons Homer had 188 jobs. I thought he always worked at the plant.
Speaker A:	Oh wow that's a lot of jobs. I had no idea.
Speaker B:	Me neither, that kind of shocked me. Do you remember the Treehouse of Horror xxi from the Simpsons?
Speaker A:	I do not remember that. Was it a good episode?
Target:	It had Daniel Radcliffe voicing Edward Cullen.

Table 4: Example of the knowledge-grounded dialogue task in Topical-Chat.

	No PT	MLM+PS	MLM	PS	SI_{FR}	$SI_{PR\text{-}MS}$	$SI_{PR\text{-}T5}$
Uniq. Resp.	0.57(±0.06)	0.76(±0.01)	0.67(±0.02)	0.68(±0.01)	0.68(±0.02)	0.7(±0.01)	0.7(±0.02)
Resp. len.	13.47(±0.53)	15.75(±0.19)	16.21(±0.27)	15.41(±0.34)	15.99(±0.07)	15.41(±0.33)	16.5(±0.34)
PPL	50.87(±6.95)	59.09(±2.4)	54.91(±3.66)	57.26(±2.51)	53.71(±1.09)	60.62(±3.27)	58.64(±0.98)
Dist-1	0.91(±0.0)	0.92(±0.0)	0.93(±0.0)	0.93(±0.0)	0.93(±0.0)	0.92(±0.01)	0.93(±0.0)
Self-BLEU	0.86(±0.01)	0.74(±0.0)	0.79(±0.01)	0.78(±0.01)	0.79(±0.01)	0.78(±0.01)	0.78(±0.0)
BLEU	0.01(±0.0)	0.03(±0.0)	0.03(±0.0)	0.03(±0.0)	0.03(±0.0)	0.03(±0.0)	0.04(±0.0)
ROUGE-1	0.16(±0.01)	0.2(±0.0)	0.21(±0.0)	0.2(±0.0)	0.21(±0.0)	0.2(±0.0)	0.21(±0.0)
METEOR	0.11(±0.0)	0.15(±0.0)	0.15(±0.0)	0.15(±0.0)	0.15(±0.0)	0.15(±0.0)	0.16(±0.0)

Table 5: Performance metrics for dialogue modelling with Topical-Chat evaluated on the 'frequent' test set. Results are averaged from 3 different pre-trained/fine-tuned models initialised with different seeds, each with 5 different seeded runs for inference.

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 do not foresee any risks stemming from the contribution in this paper.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
4

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

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Data and model training artefacts used in this study are either open-source or were previously made publicly available.

☒ 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)?
Artefacts created in this study relate to small-scale pre-trained language models. We do not foresee an intended use for these models outside of this study.

☒ 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 new data was collected for this study.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
For all data artefacts used, we cite the original works in which they were presented and which provides information about their coverage of domains, languages, linguistic phenomena, etc.

☑ 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.
3.1

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

C ☑ Did you run computational experiments?

3

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

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

☒ 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.)?
Where applicable, this information is included in the relevant Github repository that will be made available with the paper.

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.