Disfluency Generation for More Robust Dialogue Systems

Benjamin Marie
4i Intelligent Insights
Tecnoincubadora Marie Curie,
Parque Científico y Tecnológico Cartuja
Leonardo da Vinci, 18, 41092 Sevilla, Spain
b.marie@4i.ai

Abstract

Disfluencies in user utterances can trigger a chain of errors impacting all the modules of a dialogue system: natural language understanding, dialogue state tracking, and response generation. In this work, we first analyze existing dialogue datasets commonly used in research and show that they only contain a marginal number of disfluent utterances. Due to this relative absence of disfluencies in their training data, dialogue systems may then critically fail when exposed to disfluent utterances. Following this observation, we propose to augment existing datasets with disfluent user utterances by paraphrasing fluent utterances into disfluent ones. Relying on a pre-trained language model, our few-shot disfluent paraphraser guided by a disfluency classifier can generate useful disfluent utterances for training better dialogue systems. We report on improvements for both dialogue state tracking and response generation when the dialogue systems are trained on datasets augmented with our disfluent utterances.

1 Introduction

Disfluencies are common interruptions in the flow of speech. In English, it is estimated that disfluencies account for 20% of the words (Tree, 1995) and that there is a 50% probability that a sentence of 10-13 words will be disfluent (Shriberg, 1994). A probability that increases for longer sentences.

Since disfluencies are ubiquitous, they can have a significant impact on natural language processing (NLP) tasks. Previous work has largely addressed *disfluency detection* and studied the impact of disfluencies in various NLP tasks (Johnson and Charniak, 2004; Wang et al., 2010). Disfluency detection is a critical component of any NLP framework using speech transcriptions as input.

Disfluencies can mislead components of a dialogue system: natural language understanding (NLU), dialogue state tracking (DST), and response generation. On the other hand, disfluent utterances are usually absent in the publicly available dialogue datasets used for the research and development of dialogue systems. They are either removed, after disfluency detection or have never existed, for instance, in dialogue datasets made from non-spoken texts. The datasets on which dialog systems are trained and evaluated are often heavily curated. The dialogue systems trained on such datasets may then not be robust enough in real-world applications for which disfluent utterances are common.

In this paper, we propose to augment existing training datasets with disfluent paraphrases to train a more robust dialogue system. In contrast to previous work on disfluency generation, our disfluent paraphraser only requires a very limited amount of training data that makes it applicable to a wide range of scenarios.

Our contributions are as follows:

- An analysis exposing the near absence of disfluent utterances in dialogue datasets and their impact on the robustness of dialogue systems.
- A framework to generate disfluent paraphrases.
- More accurate and more robust dialogues engines trained on our augmented datasets.
- A binary disfluency classifier, model[1] and code[2], for dialogue utterances

2 Disfluency in Dialogue

Disfluencies are usually categorized as in Table 1. We can assume that depending on its category, a disfluency will not have the same impact on dialogue systems. For instance, "repair" and "restart" categories have more potential to mislead a system than "filled paused" since they may impact a large

[1] research.4i.ai/models/BERT_disfluency_cls

[2] research.4i.ai/code/BERT_disfluency_cls

Findings of the Association for Computational Linguistics: ACL 2023, pages 11479–11488

repair	I'm watching the **football... I mean the** basketball game
restart	**I would like...** I can't go there
filled pause	It was **uh** 3 days ago
interjection	**Well** I was there
repetition	He read **this** this book

Table 1: Examples of different types of disfluencies. Tokens in bold are disfluent.

portion of an utterance. The example in Table 2 illustrates how a "repair" disfluency can impact the main modules of a dialogue system, with an error made by the NLU module on the slot values that propagates to the response generation.

To verify our assumption that most dialogue datasets used for research are not disfluent, we created a disfluency classifier (Section 3.2) applied to publicly available dialogue datasets commonly used for training and evaluating dialogue systems. The classification results are presented in Table 3. We observe that disfluent utterances are much more unlikely than in a normal English speech flow. For instance, less than 4% of the utterances in SIMMC2, often used to train and evaluate multimodal dialogue systems, are disfluent.

To train more robust dialogue systems, we augment their training data with synthetic disfluent utterances. While disfluency correction is a common task, there are only a few attempts in previous work for disfluency generation.

Yang et al. (2020) proposes to generate disfluency with a neural model inserting n-grams at specific positions in fluent sentences. They focus on two disfluency categories: "repair" and "repetition". Their approach is able to generate natural disfluent sentences with a model trained on 29k disfluent sentences. In contrast, our approach relying on a paraphraser is able to generate any kind of disfluency but is not as conservative. Our approach is not constrained to inserting tokens at specific positions.

More recently, Gupta et al. (2021) and Passali et al. (2022) proposed to generate disfluent sentences using heuristics. While their approaches are admittedly generating less natural disfluent sentences than with a neural model, they do not require to be trained and are able to generate disfluencies from any category covered by the heuristics.

Utterance	I would like to book a ticket for Boston uh no sorry for Miami
NLU	Intent: book_ticket, slots: {destination: Boston}
Response	I booked your flight for Boston

Table 2: Example of dialogue engine failure due to a disfluent utterance.

Dataset	#Utter.	%Disfluent
dailyDialog	141,864	8.52%
MultiWOZ2.2	56,776	6.25%
SIMMC2	38,127	3.29%

Table 3: Percentage of user utterances labeled disfluent by a disfluency classifier in the train split of dailyDialog (Li et al., 2017), MultiWOZ2.2 (Zang et al., 2020), and SIMMC2 (Kottur et al., 2021) datasets.

3 Disfluency Generation

Our disfluent paraphraser is applied to fluent utterances, identified by a disfluency classifier, from dialogue datasets. Then, the disfluent utterances generated are added to the dialogue datasets and used to train more robust dialogue systems following a standard training pipeline.

3.1 Disfluent Paraphraser

Pre-trained large language models (LLM) have demonstrated impressive results in most natural language generation tasks. Previous work proposed to use and evaluate LLM for disfluency correction (Saini et al., 2021; Gupta et al., 2021; Passali et al., 2022). We propose to also use LLM for disfluency generation.[3] As for the training data for the paraphraser, we need disfluent dialogue utterances paired with their fluent version, manually created, so that the model can learn the sequence-to-sequence task of generating a disfluent utterance given a fluent one. Since we lack large training data for these tasks for most languages and domains, we propose to perform few-shot learning for disfluency generation. Concretely, we fine-tune the LLM on a few training examples. Since correcting a few disfluent utterances by hand is rather cheap, we

[3]We consider the LLM itself as a hyperparameter of our approach. For this paper, we use T5 (Raffel et al., 2020) due to its good performance for natural language generation (NLG) and relatively low computational cost, but other architectures and larger models used in NLG, such as BART (Lewis et al., 2020), OPT (Zhang et al., 2022), and BLOOM (Workshop et al., 2022), could yield similar or better results.

assume this scenario to be realistic and applicable to most domains and languages.

In preliminary experiments, we observed beam search to be very conservative at inference time with our paraphraser, i.e., preserving the original structure and vocabulary of the fluent utterances. Since our goal is to augment datasets and generate diverse disfluencies, we propose to sample the search space during decoding to generate more diverse sequences with less overlap with the source utterance. This is particularly intuitive for generating disfluent utterances, for which a more aggressive sampling, to some extent, will generate more disfluent utterances. We found nucleus sampling (Holtzman et al., 2020) to generate outputs diverse enough with a top_p hyperparameter appropriately optimized (see Section 3.2).

3.2 Disfluency Identification

The dialogue datasets often contain manual annotations for NLU and DST for each user utterance. It is critical that these annotations remain valid for the generated disfluent utterances. If the paraphraser is too aggressive, the utterance may change meaning and will not match anymore the annotations.

We propose to use a disfluency classifier whose objective is to identify whether a user utterance is fluent or disfluent. If an utterance is classified as disfluent, our paraphraser will not be applied to this utterance. Moreover. we use the classifier decision to tune the aggressiveness of our paraphraser. For instance, if an utterance is identified as fluent but with a low probability, according to the classifier, we may only need to introduce a few modifications to make it disfluent. If an utterance is clearly found fluent by the classifier, a more aggressive disfluent paraphrasing should be performed to ensure it is disfluent enough.

In practice, this tunable aggressiveness is implemented in our paraphraser at inference time, using the probability α yielded by the classifier for an utterance to be disfluent to set the top_p hyperparameter of nucleus sampling as follows:

$$\text{top\_p} = \min(\alpha + \beta, 1.0) \quad (1)$$

where β is a constant between 0 and 1. In practice, we found that $\beta = 0.2$ yields useful disfluent utterances, but we argue that this may not be the case for all use cases, such as applying the paraphraser to datasets in a very different style and domain, and that consequently, β should be tuned.[4]

As for the classifier itself, we propose to use BERT (Devlin et al., 2019) for binary classification. This is a simpler classification that the one proposed by previous work (Yang et al., 2020) that uses BERT to directly classify disfluency at token level. The training data for our classifier is then easier to create since we only need native speakers to label whether a sentence is fluent or disfluent.

4 Experiments

4.1 Datasets

We trained our paraphraser and classifier on the Fisher English corpus created by Post et al. (2013)[5] which is a translation of the original Fisher Spanish corpus.[6] We paired this corpus with its fluent version (Salesky et al., 2018)[7] in which the disfluencies have been manually corrected. Statistics of the full parallel corpora used are given in Table 4.

We report on experiments with dialogue tasks using SIMMC2[8] augmented with disfluencies for DST and response generation.

4.2 Settings and Baseline Systems

We trained our model for disfluency generation using T5.[9] We use the base version and acknowledge that we may get better results with a larger model but at a greater computational cost. The base version is a Transformer (Vaswani et al., 2017) with 12 layers, 12 attention heads, a feed-forward dimension of 3072, and embeddings of 768 dimensions.

[4] One of the drawbacks of using a varying top_p is that it complicates the implementation of batch decoding since we have utterances that would be paraphrased with different top_p in the same batch. Since we only paraphrase datasets for training, the decoding time was not our main concern and we simply paraphrase utterances one by one.

[5] github.com/joshua-decoder/fisher-callhome-corpus

[6] catalog.ldc.upenn.edu/LDC2010S01

[7] github.com/isl-mt/fluent-fisher

[8] github.com/facebookresearch/simmc2

[9] huggingface.co/t5-base

Dataset	#lines	#tokens fluent-disfluent
train	138,719	1.18M-1.44M
dev (dev.en.0)	3,976	30.64k-39.99k
test (test.en.0)	3,640	30.15k-39.61k

Table 4: Statistics of the parallel fluent-disfluent Fisher English corpus. We indicate between parentheses the original names of the datasets we used for dev and test.

System	#Disfluent examples	Dialogue state tracking Joint Accuracy	Dialogue state tracking Slot F1	Response generation BLEURT
Original	0	48.8/49.1/38.5	83.9/84.1/77.0	39.3/39.2/39.8
LARD	0	48.9/49.0/41.5	84.0/84.1/80.0	39.5/38.4/40.1
Plan&Gen	all	49.1/49.0/43.1	84.5/84.9/82.0	39.8/39.3/40.5
General Paraphraser	0	49.0/49.6/38.9	84.1/84.5/77.1	39.7/39.6/39.1
Disfluent Paraphraser	50	48.7/48.0/44.1	84.6/84.5/83.1	38.1/38.3/39.7
	500	49.5/49.5/44.7	85.0/85.3/84.9	39.8/39.4/40.6
	5000	**49.6**/50.0/**44.9**	85.3/**85.5**/85.1	39.9/39.6/40.5
	all	**49.6/50.1/44.9**	**85.4**/85.4/**85.2**	**40.0/39.6/40.7**

Table 5: Results for SIMMC2. a/b/c are scores obtained on the devtest with all the utterances (a), only the fluent utterances (b), and only the disfluent utterances (c), where fluent and disfluent utterances are identified by the classifier. The second column indicates the number of training examples from the Fisher parallel corpus exploited to train a disfluency generator. The highest numbers are in bold.

Since we aim at few-shot learning, we fine-tuned T5 on subsamples of different sizes of the Fisher train fluent-disfluent parallel data, containing 50, 500, 5,000, or all the available parallel utterances, for 20 epochs with standard hyperparameters.[10] We select the best model according to BLEURT (Sellam et al., 2020) on the Fisher validation data.

We identified 36,873 fluent utterances in SIMMC2 using our BERT classifier,[11] trained on the same data as the paraphraser, and paraphrase them while keeping their annotations for DST the same. The 1,254 remaining utterances identified as disfluent are not paraphrased. The generated disfluent utterances are added to the original SIMMC2 yielding a new total of exactly 75,000 utterances.

For evaluation in dialogue, we use the same pipeline proposed by Kottur et al. (2021): GPT-2 is fine-tuned on the augmented training data for 5 epochs and is prompted with user utterances. We denote this configuration **Disfluent Paraphraser**. For DST, we use the same evaluation script provided by the SIMMC2 repository. For response generation, we use BLEURT. We compared our approach with the following systems.

Original: This is the same baseline system proposed by Kottur et al. (2021). GPT-2 is fine-tuned on the original SIMMC2 for 10 epochs.

LARD: We used the LARD heuristic-based framework,[12] with default hyperparameters, to make the fluent utterances disfluent. LARD is not trainable and consequently cannot exploit the disfluent training examples.

Plan&Gen: We used the framework proposed by Yang et al. (2020) to insert disfluencies into the fluent utterances. This system can be considered as our baseline system.

General Paraphraser: We evaluate a standard paraphraser, i.e., not trained to generate disfluencies, using T5 fine-tuned on the "paranmt_filtered" compiled by Krishna et al. (2020) containing 75k paraphrases in mixed domains.

The only difference between LARD, Plan&Gen, General Paraphraser, and our Disfluent Paraphraser configurations is that they rewrite the same fluent utterances but using different approaches.

4.3 Results

We evaluated dialogue models on the entire devtest of SIMMC2, but also on the portions identified as fluent (8,321 utterances) or disfluent (288 utterances) to highlight where each model is the most effective. Our proposed approach for disfluency generation yields the most useful training data. Our disfluent paraphraser outperforms all the other systems for both DST and response generation. While LARD and Plan&Gen both improve the joint accuracy and slot F1 for the disfluent part of SIMMC2, the scores remain similar for the fluent part of SIMMC2. Interestingly, we observe the reverse with the general paraphraser which yields better

[10] Fine-tuning T5 on all these subsamples took less than a day on an nVidia RTX3060 12Gb GPU.

[11] Our classifier was trained using the Hugging Face Transformers default pipeline (Wolf et al., 2020). It reaches an F1 score of 81.4 on the Fisher test set. We released our model and code (links in the introduction).

[12] github.com/tatianapassali/artificial-disfluency-generation

results on the fluent part. Our disfluent paraphraser is the only system that improves the results on both fluent and disfluent utterances. Nonetheless, we also observe that our system requires at least 500 training examples to avoid a drop in BLEURT and joint accuracy on the fluent part. Indeed, we manually observed that when T5 is fine-tuned on only 50 fluent-disfluent utterance pairs, the generated disfluencies tend to be very noisy with many meaningless utterances, e.g., empty or containing sequences of many symbols. Those could be easily filtered with heuristics to improve the quality of the generated data.

5 Conclusion

We demonstrated that our disfluent paraphraser generates useful disfluent paraphrases to better train dialogue models and especially improve their robustness to disfluent utterances. Our approach improves dialogue state response and response generation for both fluent and disfluent user utterances. As future work, we would like to address the limitations discussed in Section 6.

6 Limitations

The main limit of our approach is that our paraphraser may generate meaningless utterances as we observed when trained on very few examples. To quantify these instances, an intrinsic evaluation of our paraphraser should be performed. Previous work proposed automatic evaluation of the disfluency generated using BLEU. We argue that the number of valid disfluent paraphrases for a fluent utterance is so large that BLEU cannot be a fair metric for our approach since it would only reward the specific utterances given as references. Only a thorough human evaluation can provide the necessary feedback on the naturalness, adequacy, and overall quality of the disfluency generated. Then, heuristics could be designed to filter out generated utterances of poor quality.

SIMMC2 evaluation has also a very small number of disfluent utterances which only exhibit a few instances of some of the disfluency categories presented in Section 2. Our results may not be as representative as we would like of a real-world scenario. Since all the publicly available dialogue datasets, annotated with intents and slot values, are mainly fluent, more representative evaluation datasets with very diverse types of disfluencies should be created.

Finally, the parallel Fisher corpus is not ideal to train an English paraphraser since it is a translation from Spanish. We did observe some translation errors and artifacts in the dataset, such as some Spanish characters like "¿", that may negatively affect the performance of our paraphraser.

Ethical Considerations

Language models are biased by the data used to train them. Our fine-tuning of BERT and T5 with the Fisher corpus potentially created biases or amplified some of the biases inherited from these two base models. We acknowledge that this work has the potential to be used to harm minorities, for instance, by unfairly classifying or amplifying disfluencies in utterances expressed by minority groups.

We decided to delay the public release of our models, datasets, and code used for disfluency generation until our work has gone under an entire peer-review cycle and publicly presented to receive as much feedback as possible.

On the other hand, we are releasing our disfluency classifier, in the form of fine-tuned BERT models and code for fine-tuning and evaluation, as we believe these resources can be useful for the research community while posing a much lower risk of harmful exploitation than our disfluent paraphraser.

Acknowledgments

We would like to thank the reviewers for their insightful comments and suggestions. This work was partly supported by the NEOTEC grant, reference SNEO-20211360, and the Torres Quevedo Program PTQ2021-011729.

References

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

Aditya Gupta, Jiacheng Xu, Shyam Upadhyay, Diyi Yang, and Manaal Faruqui. 2021. Disfl-QA: A benchmark dataset for understanding disfluencies in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3309–3319, Online. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Mark Johnson and Eugene Charniak. 2004. A TAG-based noisy-channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 33–39, Barcelona, Spain.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

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.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Tatiana Passali, Thanassis Mavropoulos, Grigorios Tsoumakas, Georgios Meditskos, and Stefanos Vrochidis. 2022. LARD: Large-scale artificial disfluency generation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2327–2336, Marseille, France. European Language Resources Association.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the fisher and callhome Spanish-English speech translation corpus. In *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, Heidelberg, Germany.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Nikhil Saini, Drumil Trivedi, Shreya Khare, Tejas Dhamecha, Preethi Jyothi, Samarth Bharadwaj, and Pushpak Bhattacharyya. 2021. Disfluency correction using unsupervised and semi-supervised learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3421–3427, Online. Association for Computational Linguistics.

Elizabeth Salesky, Susanne Burger, Jan Niehues, and Alex Waibel. 2018. Towards fluent translations from disfluent speech. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, Athens, Greece.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, Citeseer.

Jean E.Fox Tree. 1995. The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech. *Journal of Memory and Language*, 34(6):709–738.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wen Wang, Gokhan Tur, Jing Zheng, and Necip Fazil Ayan. 2010. Automatic disfluency removal for improving spoken language translation. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5214–5217.

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.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden,

Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. Bloom: A 176b-parameter open-access multilingual language model.

Jingfeng Yang, Diyi Yang, and Zhaoran Ma. 2020. Planning and generating natural and diverse disfluent texts

as augmentation for disfluency detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1450–1460, Online. Association for Computational Linguistics.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The section provided after the conclusion.

☑ A2. Did you discuss any potential risks of your work?
The section provided after the conclusion.

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

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

B ☑ Did you use or create scientific artifacts?

section 4

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

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Still under discussion.

☑ 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 limitations and ethical considerations

☒ 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?
Left blank.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

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

C ☑ Did you run computational experiments?

section 4

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

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 4

☑ 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

☑ 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.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.