The SelectGen Challenge: Finding the Best Training Samples for Few-Shot Neural Text Generation

Ernie Chang[*], Xiaoyu Shen[*], Alex Marin[⋒], Vera Demberg
Dept. of Language Science and Technology, Saarland University
[⋒] Microsoft Corporation, Redmond, WA
{cychang,xshen}@coli.uni-saarland.de

Abstract

We propose a shared task on training instance selection for few-shot neural text generation. Large-scale pretrained language models have led to dramatic improvements in few-shot text generation. Nonetheless, almost all previous work simply applies random sampling to select the few-shot training instances. Little to no attention has been paid to the selection strategies and how they would affect model performance. The study of the selection strategy can help us to (1) make the most use of our annotation budget in downstream tasks and (2) better benchmark few-shot text generative models. We welcome submissions that present their selection strategies and the effects on the generation quality.

1 Introduction

Few-shot text generation is an important research topic since obtaining large-scale training data for each individual downstream task is prohibitively expensive. Recently, pretraining large neural networks with a language modeling objective has led to significant improvements across different few-shot text generation tasks (Radford et al., 2019; Lewis et al., 2020) and many techniques are proposed based on them (Chen et al., 2020; Schick and Schütze, 2020a; Zhang et al., 2020; Kale, 2020; Chang et al., 2020, 2021a,b; Li and Liang, 2021). However, all previous works simulate the few-shot scenario by randomly sampling a subset from the full training data. Little to no attention has been paid to the selection strategies.

The goal of the proposal is to call for innovative ideas on searching for an optimal strategy to select the few-shot training instances, as well as a comprehensive analysis of how the selection strategy would affect the model performance. The study of selection strategies is motivated by two rationales:

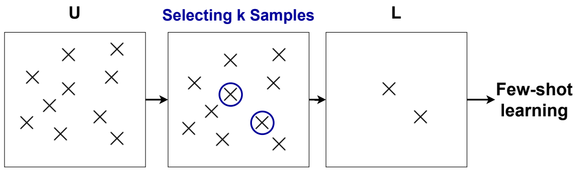

Figure 1: **Training scenario for few-shot text generation**: **U** represents unlabeled data and **L** indicates labeled instances. The annotation budget only allows selecting K data for annotating the reference text. We aim to identify the K most representative instances that, when annotated and trained on them, leads to a best model performance.

First, random sampling leads to a large variance of model performance (Zhang et al., 2020; Schick and Schütze, 2020a,b). Yet current works sample their own training data which makes it difficult to compare across different models. One can then not be sure whether an improved performance can be really ascribed to the model or the randomness of sampling. Using a stable selection strategy to find the most informative few-shot instances can provide a fair platform and better benchmark different few-shot generative models. Second, in practical applications, e.g. document summarization, the training data is usually obtained by manually annotating the summaries for some selected documents. In Figure 1, we illustrate the typical training scenario for text generation where the annotation budget only allows annotating a limited amount of data. Studying the optimal selection strategy can help make the most use of our annotation budget. Specifically, we focus on the label-free setting where *the selection can only condition on the unannotated data*. Although leveraging the reference text may benefit the selection strategy, it conflicts with the realistic setting where we need to first select the data then get its annotated reference text.

The selection task resembles the theme of active learning (Balcan et al., 2007; Chang et al., 2020, 2021c,a), where the model keeps identifying the

[*] Equal contribution. X.shen is now at Amazon Alexa AI.

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 325–330,
Aberdeen, Scotland, UK, 20-24 September 2021. ©2021 Association for Computational Linguistics

most informative instances to get labeled. We can consider the task as a starting step before applying active learning, after which more annotations can be continuously collected to further improve the model.

2 Task Description

Following the training scenario shown in Figure 1, we denote the unlabeled data as $U_1, U_2, \ldots, U_n$ where n is the data size. Depending on the downstream text generation task, "data" can mean different types of inputs like unlabeled structured data, documents and paragraphs respectively in the context of data-to-text, document summarization and question generation. We will select K instances from the whole unlabeled dataset, annotate them with reference text, and then train a neural generative model based on the annotated data. K is defined based on the annotation budget. In this work, since we focus on the few-shot scenario, K is set to be small ($\leq$ 2000). The goal is to *find the most representative K instances that can lead to the optimal performance when being annotated and trained on them.*

2.1 Submission Requirement

Participants are required to submit:

- An executable code that takes as input a set of unlabeled data, outputs K selected data that should be annotated.
- Selected training instances for $K = 50, 200, 500$ and 2000 together with model generations on the testset.
- A report that explains how the proposed selection strategy works and an analysis of its performance on the provided datasets.

While it is acceptable to take into account task or language specific features, participants are encouraged to submit selection strategies that are:

- Task agnostic. The selection strategy would work for a broad range of text generation tasks with various input-output formats.
- Language agnostic. The selection strategy can be seamlessly applied to same tasks in other languages.
- Model agnostic. The selection strategy can select most informative instances that improve the performance for a broad types of generative models (with various model architectures and training algorithms).
- K-agnostic. The selection strategy should work by varying the number of K.

2.2 Data

We will select representative datasets which cover three different text generation tasks. It will include but not limited to:

1. Data-to-text: We use the dataset for the E2E challenge (Novikova et al., 2017) which contain 50,602 data-text pairs with 8 unique slots in the restaurant domain.
2. Document Summarization: We use the CNN/Dailymail dataset (non-anonymized version) (Hermann et al., 2015) which contains 312,084 document-summary pairs.
3. Question generation: We use the SQuAD dataset (Rajpurkar et al., 2016) with over 100k questions. Following Du et al. (2017), we focus on the answer-independent scenario to directly generate questions from passages.

All the above datasets contain parallel input-output pairs for train/validation/test. We can simulate our few-shot scenario by only *allowing leveraging K input-output pairs from the training set*. The participants can decide which K training instances to select based on all the inputs in the training set [1]. Once the selected instances are determined, the model can then be trained on the K input-output pairs. It is also worth mentioning that in order to simulate the true few-shot scenario, *participants can only rely on the K input-output pairs for both training and validation*, i.e., no extra held-out examples are available for hyperparameter tuning and model selection (Schick and Schütze, 2020a; Perez et al., 2021). The participants can deside how to split them into the training and validation set.

We select the above three datasets only as examples for demonstration. Participants are encouraged to test their model on more diverse types of text generation tasks, e.g., tasks from the GEM benchmark (Gehrmann et al., 2021). Nevertheless, we recommend participants to first test and analyze

[1] The submitted instance selection algorithm can only condition on the inputs in the training set. However, participants are welcome to incorporate the reference text or testset distribution to analyze the theoretical upper bound performance.

their model on the above three datasets. In the final test, we will evaluate on the above three datasets to allow comparison across different submission. It is, however, totally acceptable to not target at all of the above three tasks. The participants can decide the tasks and datasets depending on their interest.

2.3 Generative Model

It is encouraged that participants can test their selection strategy on a wide list of generative models. In the end, to allow for a fair comparison across all submissions, we will test the selection algorithm by finetuning the open-sourced Bart model (Lewis et al., 2020) on the selected training instances with maximum likelihood. Bart is pretrained with a denoising autoencoder objective on large amount of text data and has been the state-of-the-arts for many text generation tasks. Therefore, we recommend to first test with this standard generative model. There have been many algorithms proposed for improved generation quality under the few-shot scenario like pattern exploitation training (Schick and Schütze, 2020a; Li and Liang, 2021; Lester et al., 2021) and cyclic training (Tseng et al., 2020; Chang et al., 2021a; Guo et al., 2021). We welcome test results using different types of generative models. Nonetheless, *the focus of the shared task is on the instance selection algorithm but not the few-shot generative model*. While it is nice to provide data points that demonstrate state-of-the-art results, generating with the most advanced model for better evaluation scores is by not means the main purpose.

2.4 Schedule

We follow the following schedule for the shared task of training instance selection:

- **December 15th, 2021**. The shared task is announced along with the selected text generation tasks and datasets.

- **February 15th, 2022**. The submission system and public leaderboard are open. Participants can deploy and test models with the provided automatic evaluation scripts.

- **May 15th, 2022**. This is the deadline for software and report submission. The manual evaluation begins. We will test the submitted selection algorithms with the same generative model and hyperparameter tuning mechanism. Model outputs will be compared with both automatic metrics and human evaluation.

- **June 15th, 2022**. The results of the automatic metrics and human evaluations will be announced.

After getting all the evaluation results, we will make a report to analyze different submissions. The shared task's findings are then presented at the following INLG.

3 Evaluation

The final evaluation will be conducted on the following two settings:

1. We apply the submitted selection algorithms to select K training instances and then finetune on them using a fixed strategy (with Bart model, same train/validation split and hyperparameter tuning mechanisms). The purpose is to evaluate all selection algorithms under a fair setting. In this setting, we will run the selection algorithm and training pipeline on our side to ensure fairness.

2. For each submission, we evaluate the model outputs of the best system. The purpose is to get an upper bound score for few-shot text generation with the best combination of settings (random seed, generative model, optimization algorithm, train/validation split, hyperparameter tuning, etc). In this setting, we will rely on the submissions of model outputs from the participants.

We will provide scripts for the automatic evaluation. The human evaluation will be conducted after all submissions are received under the same platform and metrics.

3.1 Automatic Evaluation

The evaluation metrics differ according to the downstream tasks. The metrics used for the final evaluation will be announced after the submission system is open. Participants are encouraged not to focus on one specific metric to avoid overfitting to it. The final evaluation will adopt metrics following into the following categories:

- Lexical similarity, which measure the lexical overlap between the model output and the gold references, including many popular metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005).

- Semantic relevance, which measures the semantic similarity between the model output and the gold references, including the newly proposed BertScore (Zhang et al., 2019) and BLEURT (Sellam et al., 2020).

- Consistency with task input, which measures if the output contains consistent information with the task input and no hallucinations. Many works have proposed metrics based on question answering (Eyal et al., 2019; Durmus et al., 2020), natural language inference (Kumar and Talukdar, 2020) and mutual information (Shen et al., 2018; Zhang et al., 2018).

- Output diversity, which measures if the model can produce diverse outputs with different inputs, including metrics like the count and entropy of distinct uni/bi-grams (Li et al., 2016; Dušek et al., 2020).

- Other task-specific requirement, e.g., slot-error rate for data-to-text and compression rate for document summarization.

After the submission system opens, we will announce the metrics we picked for the automatic evaluation and provide the evaluation script.

3.2 Human Evaluation

We will also provide human evaluation scores on the system outputs since none of the automatic metrics can correlate perfectly with the generation quality. We will follow the recently proposed taxonomy of human evaluation measures by Belz et al. (2020); Su et al. (2020) and follow the reporting strategies proposed by Howcroft et al. (2020). The human evaluation will be focused on the following two parts, which are specifically hard to be accurately measured by automatic metrics:

- Fluency. If the output itself is a fluent sentence that can be well understood by humans, defined by a 5-scale Likert score.

- Consistency. If the output is consistent with the input and does not contain hallucinations, defined by a binary true/false score.

The human evaluation will be conducted after collecting all the submissions. It will be performed under a unified pipeline and annotation guideline to make sure results are comparable across model outputs from all submitted systems. To make the analysis comprehensive, participants are nonetheless encouraged to also perform their own human evaluation and include the results in their report.

3.3 Variance of Model

An important factor worth mentioning is the variance of the model. The variance of the model output can come from different steps, e.g., variance of the selection algorithm, random seed of training, hyperparameter selection, etc. It is rather straightforward to simply apply a random sampling strategy to select the K training instances and find a relatively good selection choice by brute force. However, this is clearly against the purpose of the shared task. We aim to find out a *selection algorithm that can stably help us identify the most representative training instances* instead of only getting the instance set. Therefore, when doing the final evaluation, if the submitted selection algorithm is not deterministic, we will run the algorithm 5 times to get 5 different selection sets and aggregate the results. The variance of the evaluation will also be reported (For the setting 1 of evaluation). For setting 2, we rely on the participants themselves to provide the selected instance set and the model outputs. Participants must indicate clearly how the instance set is determined, e.g., whether they cherry-pick a best instance set by randomly running the algorithm for many times, or leverage other information like the reference text for other inputs, testset distribution, etc.

4 Conclusion

In this proposal, we target at the problem of training instance selection for few-shot text generation. Current research simply applies random sampling which has a large selection variance and can lead to suboptimal performance. The main goal of the task is to call for more attention on this largely under-explored problem, gather innovative ideas on proposing selection algorithms and provide a fair platform for comparison.

We believe our shared task can be an important supplement to the study of few-shot text generation, where most works focus solely on the generative algorithm while neglecting the training instance selection. Selection strategies proposed in this task can be used to better benchmark model performances for few-shot text generation. Importantly, the task was inspired by realistic industrial settings and requirements and will hopefully benefit multiple areas of NLP research including human-in-the-loop learning and other active learning based research, where the resource and time constraints calls for the task to be performed.

Acknowledgements

This research was funded in part by the German Research Foundation (DFG) as part of SFB 248 "Foundations of Perspicuous Software Systems". We sincerely thank the anonymous reviewers for their insightful comments that helped us to improve this paper.

References

Maria-Florina Balcan, Andrei Broder, and Tong Zhang. 2007. Margin based active learning. In *International Conference on Computational Learning Theory*, pages 35–50. Springer.

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.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Ernie Chang, Jeriah Caplinger, Alex Marin, Xiaoyu Shen, and Vera Demberg. 2020. Dart: A lightweight quality-suggestive data-to-text annotation tool. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 12–17.

Ernie Chang, Vera Demberg, and Alex Marin. 2021a. Jointly improving language understanding and generation with quality-weighted weak supervision of automatic labeling. *Proceedings of EACL 2021*.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021b. Neural data-to-text generation with lm-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768.

Ernie Chang, Hui-Syuan Yeh, and Vera Demberg. 2021c. Does the order of training samples matter? improving neural data-to-text generation with curriculum learning. *Proceedings of EACL 2021*.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot nlg with pre-trained language model. *ACL*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Association for Computational Linguistics (ACL)*.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. *arXiv preprint arXiv:2102.01672*.

Qipeng Guo, Zhijing Jin, Ziyu Wang, Xipeng Qiu, Weinan Zhang, Jun Zhu, Zheng Zhang, and Wipf David. 2021. Fork or fail: Cycle-consistent training with many-to-one mappings. In *International Conference on Artificial Intelligence and Statistics*, pages 1828–1836. PMLR.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

David M Howcroft, Anja Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

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.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

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.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *arXiv preprint arXiv:2105.11447*.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Timo Schick and Hinrich Schütze. 2020a. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

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.

Xiaoyu Shen, Hui Su, Wenjie Li, and Dietrich Klakow. 2018. Nexus network: Connecting the preceding and the following in dialogue generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4316–4327.

Hui Su, Xiaoyu Shen, Zhou Xiao, Zheng Zhang, Ernie Chang, Cheng Zhang, Cheng Niu, and Jie Zhou. 2020. Moviechats: Chat like humans in a closed domain. In *Proceedings of EMNLP 2020*, pages 6605–6619.

Bo-Hsiang Tseng, Jianpeng Cheng, Yimai Fang, and David Vandyke. 2020. A generative model for joint natural language understanding and generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1795–1807.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1815–1825.