On Training Instance Selection for Few-Shot Neural Text Generation

Ernie Chang[*], **Xiaoyu Shen**[*], **Hui-Syuan Yeh, Vera Demberg,**
Dept. of Language Science and Technology, Saarland University
{cychang,xshen}@coli.uni-saarland.de

Abstract

Large-scale pretrained language models have led to dramatic improvements in text generation. Impressive performance can be achieved by finetuning only on a small number of instances (few-shot setting). 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. In this work, we present a study on training instance selection in few-shot neural text generation. The selection decision is made based only on the unlabeled data so as to identify the most worthwhile data points that should be annotated under some budget of labeling cost. Based on the intuition that the few-shot training instances should be diverse and representative of the entire data distribution, we propose a simple selection strategy with K-means clustering. We show that even with the naive clustering-based approach, the generation models consistently outperform random sampling on three text generation tasks: data-to-text generation, document summarization and question generation. The code and training data are made available at https://gitlab.com/erniecyc/few-selector. We hope that this work will call for more attention on this largely unexplored area.

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 improvement 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, 2021b; Li and Liang, 2021; Chang et al., 2021a). However, all the 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.

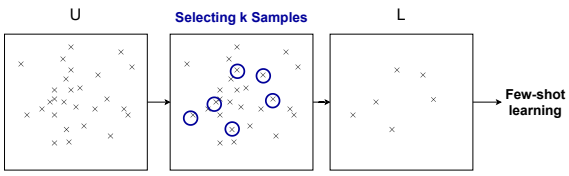

Figure 1: **Training scenario**: **U** represents unlabeled data and **L** indicates labeled instances. The annotation budget only allows selecting K data for annotating the reference text.

In this work, we present a preliminary study at searching for an optimal strategy to select the few-shot training instances. Studying the selection strategy is motivated by two rationales. 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

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

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 8–13
August 1–6, 2021.

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), where the model keeps identifying the most informative instances to get labeled. Existing active learning approaches can be roughly divided to uncertainty-based sampling and representative sampling (Settles, 2009). Uncertainty-based sampling select samples that maximally reduce the uncertainty of the model (Tur et al., 2005). This, however, requires a well-trained model with decent confidence score estimations in order to perform well. Therefore, in this paper, we opt for the representative-sampling where the selected training instances are expected to be dissimilar to each other and representative enough to cover all important patterns in the whole data distribution (Agarwal et al., 2005; Wei et al., 2015). This naturally matches the objectives of k-means clustering which minimizes the within-cluster variances while maximizing the between-cluster variances to encourage the diversity and representativeness of each cluster (Krishna and Murty, 1999; Kanungo et al., 2002). As has been shown in image classification tasks, data points closer to the cluster centroids are usually most important, while other faraway points can even be safely removed without hurting model performance (Kaushal et al., 2018; Birodkar et al., 2019). Inspired by this, we propose a simple selection strategy which first clusters the whole unlabeled dataset with the K-means algorithm, and then from each cluster, selects the data point that is closest to the cluster centroid.

We conduct experiments on three popular text generation tasks: data-to-text, document summarization and question generation. The proposed selection strategy consistently outperforms random sampling and exhibits much smaller variance.

Contribution. We present a preliminary study on training instance selection for few-shot text generation and propose a selection strategy based on K-means clustering. The proposed method shows consistent superior performance over random sampling, which can be used to make most use of the annotation budget in practical applications. Meanwhile, the selected training instances can serve as a better benchmark for few-shot text generation since they are not biased towards specific generative methods and do not have the large variance issue as found in random sampling. We further perform a set of ablation studies to analyze what contributes to a good selection. Our findings can also benefit research in active learning (Konyushkova et al., 2017) since identifying the most informative training instances is a critical step before collecting more annotations through active learning.

2 Problem Formulation

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 task, "data" can mean 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 (≤ 100). The goal is to *find the most representative K instances that can lead to the optimal performance when trained on them*.

3 Selection by K-means Clustering

The general idea of our proposed method is to first split the whole unlabeled data into K clusters, then select one data point from each cluster. Specifically, we first map each data point into a vector, then cluster the vectors with the K-means algorithm. The objective is sum of the squared errors (SSE), which is also called cluster inertia:

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{K} w_{i,j} ||x^i - \mu^j||_2^2 \qquad (1)$$

where μ^j is the centroid of the jth cluster. x^i is the embedding vector of U_i. $w_{i,j} = 1$ if x^i belongs to the cluster j and 0 otherwise. We optimize the objective function with the EM algorithm (Dempster et al., 1977) which iteratively assigns each data point into its closest cluster centroid. The initial centroid points are chosen based on the K-means++ algorithm (Arthur and Vassilvitskii, 2007). The first cluster center is chosen uniformly at random from the data points, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing cluster

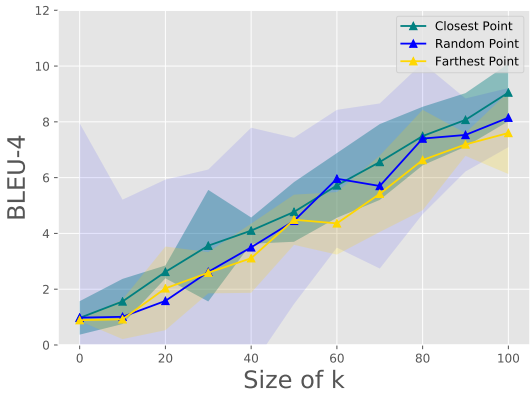

Figure 2: Ablation studies on the SQUAD corpus. Performance in BLEU-4 with increasing K between different variants of K-means where selection is based on the **closest point**, **Random point**, or **Farthest point** from the centroid.

center. By this means, we maximize the chance of spreading out the K initial cluster centers. We use 10 random seeds for selecting initial centers and the clustering with the minimum SSE is chosen.

After splitting them into K clusters, we pick from each cluster the data point that is closest to the center. We use the Euclidean distance to select, the same as the metric used for K-means clustering. The intuition is that the test performance usually depends on the nearest neighbor in the training set (Khandelwal et al., 2020; Rajani et al., 2020). Ideally data points closest to the cluster centers are most representative samples, selecting them will maximize the chance that a similar sample will be found in the training dataset.

4 Experiments

We perform our experiments on the following three representative datasets which cover three different text generation tasks:

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.

For all experiments, we finetune the open-sourced Bart model (Lewis et al., 2020) as our generative model. 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. To extract vectors used for clustering, we finetune the Bart model with its original self-supervised objective on the unlabeled data, then apply mean pooling over the last hidden states of the encoder.

In the later sections, we will first compare the model performance based on our proposed selection strategy and random sampling, then analyze the variance of them. Finally, we perform an ablation study to see the effects of in-cluster selection and embedding choices.

Comparison of Selection Strategies. In Table 1, we compare the model performance based on different selection strategies. Apart from random sampling and our proposed method, we also compare with a lower bound where all instances are randomly sampled from one cluster (within-cluster random). Adding this for comparison aims to illustrate that it is important to select diverse samples across different clusters. The performance scores are averaged over 10 different trials for each selection strategy. As can be seen, K-means based selections consistently outperforms the others. Within-cluster random sampling performs the worst, proving the importance of having diverse samples in the training instance. However, it is worth noting that although random sampling underperforms K-means selection on average, *its upper bound is much higher, suggesting the proposed K-means selection is by no means optimal*. There is still much room for improvement.

Variance of Model Performance. Table 1 also shows the variance of model performance with different selection strategies. The variance is computed based on 10 different runs. For within-cluster random sampling, the variance comes from both the choice of the cluster and the in-cluster sampling. For K-means selection, the variance comes from the choice of initial center points. We can see random sampling and within-cluster random sampling have a very large variance of up to 7.12 for $K = 100$. This further suggests that comparing few-shot models based on random sampling can be be prone to variability and prevent drawing reliable conclusions. K-means-based selection, on

	E2E			CNNDM			SQUAD		
	10	50	100	10	50	100	10	50	100
Random	4.38±7.12	11.57±4.29	26.22±2.58	13.51±6.47	24.81±3.77	35.24±2.89	1.23±6.22	3.33±5.89	7.65±3.61
IC-Random	2.15±4.58	9.80±2.62	24.71±2.71	12.30±3.89	24.71±2.45	33.29±1.92	1.34±3.23	1.79±3.77	6.97±2.55
K-means	**6.22**±2.33	**11.89**±1.39	**27.13**±2.22	**14.28±2.35**	**25.19**±3.28	**36.31**±1.08	**1.56**±2.34	**4.77**±3.61	**9.33**±2.15

Table 1: Comparisons of random sampling, within-cluster random sampling (IC-Random) and K-means selection on the E2E, CNNDM, and SQUAD corpus (BLEU-4 reported).

Embedding	E2E		CNNDM		SQUAD	
	Mean	Sum	Mean	Sum	Mean	Sum
BART	26.28	25.59	34.30	**34.46**	8.89	8.56
BART-FT	26.46	**26.32**	**36.31**	34.18	**9.55**	8.12
GloVe	25.18	23.36	33.59	31.45	7.99	7.56
FastText	**27.13**	24.85	33.23	34.30	9.33	**9.42**

Table 2: Finetuned BART generation performance comparison on **E2E**, **CNNDM**, and **SQUAD** for various embedding options for the *k-means selection* with k=100.

the contrary, is rather robust with random seeds. Therefore, for future work on few-shot text generation, we suggest that models be tested on instances selected from our proposed strategy for a fair comparison.

Effects of In-cluster Selection. In Figure 2, we show the effects of the in-cluster selection method. In our proposed method, within each cluster, we select one data point that is closest to the cluster center. To see whether it is important to select the closest data point, we compare with two selection variants that within each cluster, we select (1) one data point randomly sampled from the cluster, and (2) one data point that is farthest to the cluster center. We can observe that the choice of selection does have a big impact on the model performance. Choosing data points farthest to the cluster centers leads to the worst performance. This is consistent with previous findings (Kaushal et al., 2018; Birodkar et al., 2019) that data points farthest from cluster centers are usually outliers and less representative. Selecting them might mislead the model to capture non-generic patterns and thereby generalize poorly. In contrast, choosing data points closest to cluster centers performs slightly better than random selection. However, random selection has a much larger variance compared with closest/farthest point selection (shown as shadow).

Effects of Embedding Methods. As the K-means clustering is performed on top of the embedding vectors of unlabeled data, the choice of embedding methods could affect the performance on selected points. In Table 2, we show the effects of the different embedding methods. Apart from the finetuned Bart, we compare with embeddings extracted from (1) Bart without being finetuned on the task-specific data, (2) Glove (Pennington et al., 2014) and (3) FastText (Bojanowski et al., 2017), both finetuned on the task-specific data. For each embedding method, we compare using mean pooling and sum pooling to extract the final vector representation. The results show that finetuned Bart generally outperforms the other embedding choices. We attribute this to the similarity in the embedding space between selection with BART embeddings and and the BART generation model. Moreover, *FastText* offers a strong baseline as it does relatively well on two scenarios in E2E and SQUAD respectively. Further, we observe that *mean* pooling is generally better than the *sum* of word vectors, which is also observed in Chen et al. (2018).

Human Evaluation. To obtain further insights with respect to the generation outputs, five annotators were instructed to evaluate 100 samples for each of the three tasks to judge (1) whether the text is *fluent* (score 0-5 with 5 being fully fluent), and (2) whether it contains relevant information about its input source (*adequacy*). These scores are averaged and presented in Table 3. For **Random** selection, we sampled 10 outputs from each of the 10 trials to make it 100 samples, and the same goes for **IC-random**. We observe that the K-means algorithm select better subsets of the training samples that allow for better generalizability to unseen input sources. In particular, the outputs are generally more *adequate*. However, we see that the *fluency* of outputs remain relatively similar.

5 Conclusion

In this work, we target at the unexplored problem of training instance selection for few-shot text generation. We show that random sampling can lead to large variance and suboptimal performance. To address this problem, we propose a selection strategy based on K-mean clustering and demonstrate

	E2E	CNNDM	SQUAD
Random	4.08/4.15	**4.55**/3.27	**4.62**/3.84
IC-Random	4.32/3.54	3.62/3.01	4.23/2.74
K-means	**4.12/4.24**	4.32/**3.66**	4.51/**3.98**

Table 3: Human evaluation on 100 samples of the finetuned BART generation performance comparison on **E2E**, **CNNDM**, and **SQUAD**. Scores are presented as (*fluency* / *adequacy*).

that it consistently outperforms random sampling, and has much lower variance. We further perform a set of ablation studies to analyze the effects of data size, embedding and selection methods, showing that this is still much room for improvement. Future work can consider other clustering methods.

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

Pankaj K Agarwal, Sariel Har-Peled, Kasturi R Varadarajan, et al. 2005. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30.

David Arthur and Sergei Vassilvitskii. 2007. k-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

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

Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. 2019. Semantic redundancies in image-classification datasets: The 10% you don't need. *arXiv preprint arXiv:1901.11409*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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.

Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1815–1826.

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

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

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

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.

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

Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892.

Vishal Kaushal, Anurag Sahoo, Khoshrav Doctor, Narasimha Raju, Suyash Shetty, Pankaj Singh, Rishabh Iyer, and Ganesh Ramakrishnan. 2018. Learning from less data: Diversified subset selection and active learning in image classification tasks. *arXiv preprint arXiv:1805.11191*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. *ICLR*.

Ksenia Konyushkova, Sznitman Raphael, and Pascal Fua. 2017. Learning active learning from data. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4228–4238.

K Krishna and M Narasimha Murty. 1999. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

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.

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

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

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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.

Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. 2020. Explaining and improving model behavior with k nearest neighbor representations. *arXiv preprint arXiv:2010.09030*.

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*.

Burr Settles. 2009. Active learning literature survey.

Gokhan Tur, Dilek Hakkani-Tür, and Robert E Schapire. 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963.

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.