Ordered Diversity Sampling for Text

Ashish Tiwari
Microsoft Redmond
astiwar@microsoft.com

Mukul Singh
Microsoft Redmond
singhmukul@microsoft.com

Ananya Singha
Microsoft Bangalore
ananyasingha@microsoft.com

Arjun Radhakrishna
Microsoft Redmond
arradha@microsoft.com

Abstract

The goal of diversity sampling is to select a representative subset of data in a way that maximizes the information contained in the subset while keeping its cardinality small. We introduce the ordered diversity sampling problem and present a novel and simple approach for generating ordered diverse samples for textual data that uses principal components on the embedding vectors. We compare our approach with existing approaches using a new metric that measures diversity in an ordered list of samples. We transform standard text classification benchmarks into benchmarks for ordered diversity sampling and show that prevailing approaches perform 6% to 61% worse than our method while also being more time inefficient.

1 Introduction

The *ordered diversity sampling problem* assumes that we are given a dataset D and the goal is to sample a small ordered subset S of data points from D that is representative of the diversity in D. Some motivation for ordered diversity sampling comes from its use in (a) picking in-context examples S (from an example bank D) for prompting a large language model (LLM) [Lu et al., 2022, Guo et al., 2024, Wang et al., 2025], (b) picking a small set S of tasks (from task set D) to be solved using an expensive LLM so that we can later use the solution for S to cheaply compute the solution for the full set D [Singha et al., 2025, Agarwal et al., 2005, 2004], and (c) picking samples S for humans to annotate [Alcoforado et al., 2024]. Diversity is also important in many other contexts [Batra et al., 2012, Vijayakumar et al., 2016, Li et al., 2016, Wei et al., 2013, Sener and Savarese, 2018].

Given D, the most basic approach, *random*, picks S using random sampling. The *random* strategy does not explicitly aim for diversity. There are 3 approaches that have been proposed for diversity sampling. In the first approach, *clustering* [Monarch et al., 2021], the set D is partitioned into clusters (using unsupervised clustering) and samples are picked from each cluster. The samples are picked so that they are either close to the center of a cluster, close to the edge, or randomly placed. In the second approach, *k-center-greedy* [Sener and Savarese, 2018], a point is randomly picked from D and inserted into S, and thereafter iteratively points farthest away from the current set S are picked and added to S. In the third approach, *reverse semantic search* [Alcoforado et al., 2024], the set S is initialized to the empty set and thereafter iteratively two data points are selected from $D \setminus S$ that are farthest away from each other among all such pairs and these are added to S.

In this paper, we propose a new approach, *principled sampler v1*. In this approach, the data set D is transformed to a representation over its principal components, and a point from D is picked to be included in S if it has the largest (positive or negative) projection on a principal component or if it has a small projection on all principal components. We also propose a variant, *principled sampler v2*, that picks data points from D that have large projection on one principal component *and also*

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Algorithm 2 Principled Sampler v2

5: $Y[i] \leftarrow j^*$ where $j^* = \arg\max_j (X_{pca}[j,i] - \sum_{k \neq i} |X_{pca}[j,k]|)$
6: $Z[i] \leftarrow j^*$ where $j^* = \arg\min_j (X_{pca}[j,i] + \sum_{k \neq i} |X_{pca}[j,k]|)$

small projections on all other principal components. We also create a variant of *clustering*, called *PCA-clustering*, that first performs dimensionality reduction using PCA before using *clustering*.

All of the above approaches, except *random*, first embed the data points D in some latent embedding space. In this work, we use the OpenAI text embedding model and TF-IDF embeddings. We compare the above 7 approaches in this paper on "ordered diversity sampling" problems and observe that principled samplers outperform the others. Alcoforado et al. [2024] note that it is very difficult to beat random sampling especially on datasets that do not have outliers, as we also find.

2 Principled Samplers

Let D be a data set consisting of textual data. Formally, D is an $(N \times 1)$-matrix of string values. The intuition behind Principled Samplers is simple. If we embed the text in D into an M-dimensional real space, $\mathbb{R}^M$, and then look at the principal components of the data in $\mathbb{R}^M$, the data points that have a large projection (positive and negative) along the principal components will be "different" from each other and good candidates for inclusion in S. Also, the data points that have low projections on all the principal components will also be qualitatively different and good candidates for inclusion.

The algorithm `Principled Sampler v1` is presented in Algorithm 1. The input to the algorithm is a dataset D consisting of text column with, say, N rows. We compute M-dimensional embeddings of each of the N strings on Line 1 using any given embedding function E. This is followed by PCA on Line 2 that maps each of the N strings to an n dimensional space, where n is a parameter. We then find the (indices of the) n data points that have the largest projection on each of the n principal components respectively (Line 5). The notation $X_{pca}[*, i]$ denotes the i-th column of the $(N \times n)$-matrix X_{pca}. Next we find the (indices of the) n data points that have the most negative projection on each of the n principal components (Line 6). Finally, we find the n indices that correspond to data points that have the n smallest absolute projection on each of the n principal components (Line 11). Our procedure returns the $3n$ samples containing the union of the three arrays (Line 12). The intuition is that the first two sets Y and Z contain indices of data points that capture the "common" pattern in the dataset D, whereas the set W contains the indices of "uncommon" pattern and "outliers" Fariha et al. [2021].

Algorithm 1 Principled Sampler v1

Require: *Input*: data D, an $(N \times 1)$-matrix of strings
Require: *Parameters*: embedding function E, integer n
Ensure: *Output*: Ordered sequence of $3n$ indices
1: $X \leftarrow E(D)$ # $X \in \mathbb{R}^{N \times M}$
2: $X_{pca} \leftarrow PCA(X, \text{components} = n)$ # $X_{pca} \in \mathbb{R}^{N \times n}$
3: Initialize arrays Y, Z of size n each # $Y, Z \in \mathbb{Z}^{1 \times n}$
4: **for** $i \in [1, 2, \ldots, n]$ **do**
5: $Y[i] \leftarrow j$ where $X_{pca}[j, i] = \max(X_{pca}[*, i])$
6: $Z[i] \leftarrow k$ where $X_{pca}[k, i] = \min(X_{pca}[*, i])$
7: **end for**
8: **for** $j \in [1, 2, \ldots, N]$ **do**
9: $X_\infty[j] \leftarrow ||X_{pca}[j, *]||_\infty$ # $X_\infty \in \mathbb{R}^{1 \times N}$
10: **end for**
11: W are indices of the n smallest values in X_∞
12: **return** $[Y, Z, W]$ # Concatenate the 3 arrays

Remark 1. *It is possible that the indices in Y, Z and W overlap, which can cause Algorithm 1 to return less-than $3n$ samples. In our evaluation, we force all techniques to return the same number of samples. We make Algorithm 1 return exactly $3n$ samples by replacing repeated values in either Y, Z or W by the next best new pick for that set.* □

Principled Sampler v2. A slight modification of Principled Sampler v1 is the algorithm `Principled Sampler v2` in Algorithm 2. The changes are only on Lines 5 and 6. In Principled Sampler v1, we picked the data points that had extreme projections on the principal components. In Principled Sampler v2, we pick data points that each have a very large projection on one of the principal components, *and a very small projection on the other components*. In the notation

used in Algorithm 2, the expression $X_{pca}[j,i] - \sum_{k \neq i} |X_{pca}[j,k]|$ is maximized when $X_{pca}[j,i]$ is large and $X_{pca}[j,k]$ is close to zero for all $k \neq i$. Algorithm 2 picks j^* that maximizes this expression and these j^* (one for each choice of i) are included in Y. Similarly, the expression $X_{pca}[j,i] + \sum_{k \neq i} |X_{pca}[j,k]|$ is minimized when $X_{pca}[j,i]$ is small and $X_{pca}[j,k]$ is close to zero for all $k \neq i$. Algorithm 2 picks j^* that minimizes this expression, and these j^* (one for each choice of i) are included in Z. The outlier picks in W are identical to those in Algorithm 1.

3 Evaluation

Datasets. We picked text classification datasets from Papers with code [pap], Kaggle [kag], and Hugging Face [hug] that had a large number of labels. In total, we selected 22 benchmarks with the number of labels ranging from 3 to 98 and standard deviation 19.6 listed in the Appendix A. For each of the 22 datasets, we further down sampled, and picked a random sample of size $N = 250$, and the ground-truth labels for those N data points, to get *one* benchmark for diversity samplers. We repeated our experiments 5 times with different seeds for the random number generator.

Metric. In each of the above text classification benchmarks, a data set D is associated with a labeling function $L : D \mapsto C$, where C is a finite set of categories. Let us say a sampler picked the ordered sequence $\langle d_1, \ldots, d_n \rangle$ from D. For any prefix of this sequence, we define a function, `IsNew`, as:

$$\texttt{IsNew}(\langle d_1, \ldots, d_k \rangle) \quad := \quad \begin{cases} 0 & \text{if } \exists i < k : L(d_k) = L(d_i) \\ 1 & \text{otherwise} \end{cases}$$

Intuitively, a sample d_k is new with respect to previously picked samples $d_1, \ldots, d_{k-1}$ if its associated label is new and not seen before. We next define "wasted opportunity": given the sequence $\langle d_1, d_2, \ldots, d_k \rangle$, the pick d_k is a *wasted opportunity*, denoted $\texttt{Wasted}(\langle d_1, \ldots, d_k \rangle) = 1$, if
(a) $\texttt{IsNew}(\langle d_1, d_2, \ldots, d_{k-1}, d_k \rangle) = 0$, and
(b) there exists $d \in D$ s.t. $\texttt{IsNew}(\langle d_1, d_2, \ldots, d_{k-1}, d \rangle) = 1$.
Finally, we define our evaluation metric, the *aggregated wasted opportunity* (AWO): given a sequence $\langle d_1, d_2, \ldots, d_n \rangle$, the *aggregated wasted opportunity* metric is a number from 0 to n that is defined by $\sum_{i=1}^{n} \texttt{Wasted}(\langle d_1, \ldots, d_i \rangle)$. A method that produces *smaller* AWO score is deemed better.

Remark 2. *The AWO metric has the nice property that it remains comparable across benchmarks that have different number of labels. Consequently, we can aggregate this metric over different benchmarks, which allows us to compare different diversity samplers over a wide array of benchmarks.* □

Comparing with Baselines. For each diversity sampler, Figure 1 plots the AWO metric on the y-axis against the number of samples on the x-axis as it increases from 1 to 18. Each line in the plot corresponds to a different sampler. A lower value is better in the plot. We have 7 lines: 2 for our "principled sampler v1" and "principled sampler v2" and 5 for the following baselines: *clustering*, *pca-clustering*, *reverse-semantic-search*, *k-center-greedy* and *random*. Each method generates $3n = 18$ ordered samples and these sequences are compared using AWO metric.

We use two different embeddings: *openai* refers to the use of the pretrained model, `text-embedding-3-small`, from OpenAI, and *tfidf* refers to the use of the TFIDF vectorizer from the sklearn library. Specifically, we used the TFIDF vectorizer available in sklearn library with `max_df=0.5`, `min_df=5`, and `stop_words="english"`.

Figure 1 shows the approaches proposed in this paper perform better than *all* the baselines. If we replace OpenAI embedding with tfidf embeddings, we get a similar plot (Appendix B). This shows our proposed technique is not just designed to work with one kind of embedding but the improvements persist across the choice of embeddings. This is further evidence that there is inherently some value in the our diversity sampling approach. We note that some of the baselines perform worse than random sampling, but our approach performs better than it. We further note that adding dimension-reduction via PCA before clustering makes it better than random sampling. This shows that PCA helps and we believe that our approach is able to fully exploit the benefits of PCA to perform diversity sampling.

Effect of the Embedding Procedure. When we compared the performance of Principled Samplers v1 and v2 while changing the embedding, we observed that the text-embedding model from OpenAI performs better as expected; see Figure 3 in Appendix B. We also noted that changing the embedding affects Version v1 and v2 in the same way, although Version v2 appears slightly more robust to embedding change.

	tfidf	openai
v1	3.6	0.5
v2	**0.0**	**0.0**
cls	15.5	20.7
pca-cls	6.2	8.1
rss	6.9	24.4
k-ctr	61.1	16.5
rnd	5.9	13.3

Table 1: Percent increase in aggregated wasted opportunity (AWO) scores for different approaches when compared to v2.

	tfidf	openai
v1	0.0	**0.0**
v1-Y	2.8	1.7
v1-Z	3.5	7.5
v1-W	**-0.7**	0.9
v2	0.0	**0.0**
v2-Y	0.6	1.4
v2-Z	8.3	9.8
v2-W	**-0.6**	1.8

Table 2: Table showing the %increase in AWO for ablated variants relative to v1 and v2.

s	48	48	72
d	1500	2500	2500
v1	8.8	**13.8**	**18.9**
v2	**8.4**	15.4	19.4
cls	14.5	21.0	32.6
rss	59.4	180.2	231.9
k-ctr	21.6	85.9	92.9

Table 3: Time taken by different methods to sample s from dataset of size d for $s \in \{48, 72\}$ and $d \in \{1500, 2500\}$.

Table 1 shows the percentage increase in the AWO metric for the 7 diversity samplers for each of the two embeddings. Since Principled Sampler $v2$ performed the best for both embeddings, the percentage increases in each row are reported with respect to $v2$'s score in that row. Existing approaches show a degradation ranging from 6% to 61% compared to performance of Principled Sampler $v2$. The PCA+Clustering (pca-cls) approach was within 6-8% of our approach and random (rnd) was withng 6-13%, and these two were the best among the rest. Both reverse semantic search (rss) and greedy k-center (k-ctr) are impacted a lot by the choice of embedding.

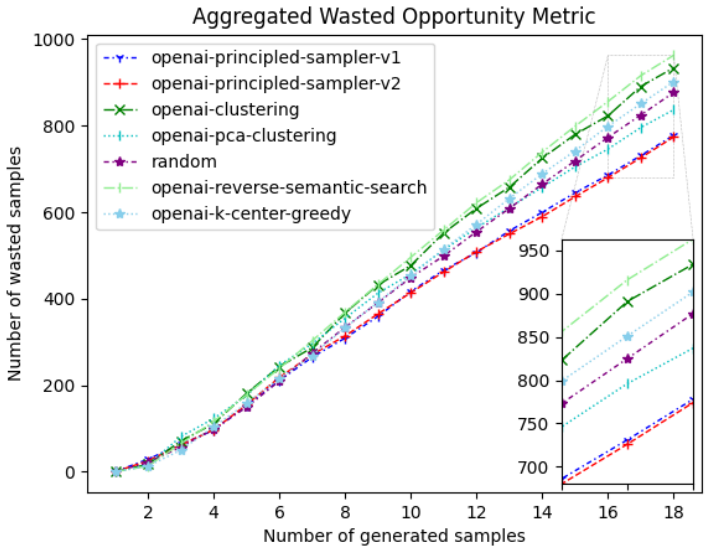

Figure 1: Comparing with baselines using $3n = 18$ samples using openai embeddings.

Ablation Studies. We created 3 ablated versions each of Versions v1 and v2 by replacing each of Y, Z, and W by random sampling one by one, and named them -Y, -Z and -W. Table 2 reports the *degradation* in performance of each of the ablated versions compared to its *respective* baseline. The main observation is that the AWO metric shows up to a 9.8% increase in the ablated versions. There are two cases, $v1 - W$ and $v2 - W$, where the ablated version performed better, but only very slightly, when using tfidf. This is not unexpected since the set W was designed to catch uncommon outliers and non-conforming data points.

Time Comparison. Table 3 reports the time taken by the different methods to sample s samples from d data points. We see that Principled Sampler v1 is the most efficient. This is surprising since PCA can be expensive; however, greedy k-center (k-ctr) and reverse semantic search (rss) have a nonparallelizable iterative loop that makes them scale poorly. On the other hand, our Versions v1 and v2 can pick each sample almost independently of the other in parallel. We observe that clustering (cls) has time efficiency competitive with our methods.

4 Conclusion

We presented a PCA-based approach for diversity sampling, which is task-agnostic and generates an ordered list of samples. We also defined the aggregated wasted opportunity metric for evaluating the diversity of an ordered list of samples and showed that our technique outperforms techniques described in the literature. In addition, our technique is time efficient compared to existing techniques.

References

Datasets. https://huggingface.co/datasets. Accessed: 2024-12-20.

Datasets. https://kaggle.com/datasets. Accessed: 2024-12-20.

Datasets. https://paperswithcode.com/datasets. Accessed: 2024-12-20.

Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, July 2004. ISSN 0004-5411. doi: 10.1145/1008731.1008736. URL https://doi.org/10.1145/1008731.1008736.

Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. In *Combinatorial and Computational Geometry*, pages 1–30. Cambridge University Press, New York, 2005. URL https://api.semanticscholar.org/CorpusID:13812735.

Alexandre Alcoforado, Thomas Palmeira Ferraz, Lucas Hideki Okamura, Israel Campos Fama, Arnold Moya Lavado, Bárbara Dias Bueno, Bruno Veloso, and Anna Helena Reali Costa. From random to informed data selection: A diversity-based approach to optimize human annotation and few-shot learning, 2024. URL https://arxiv.org/abs/2401.13229.

Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse m-best solutions in markov random fields. In *Computer Vision – ECCV 2012*, pages 1–16, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33715-4.

Aditya Bhaskara, Mehrdad Ghadiri, Vahab Mirrokni, and Ola Svensson. Linear relaxations for finding diverse elements in metric spaces. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4105–4113, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 13(3), July 2017. ISSN 1549-6325. doi: 10.1145/3086464. URL https://doi.org/10.1145/3086464.

Thomas Eiter, Esra Erdem, Halit Erdogan, and Michael Fink. Finding similar/diverse solutions in answer set programming. *Theory and Practice of Logic Programming*, 13(3):303–359, November 2011. ISSN 1475-3081. doi: 10.1017/s1471068411000548. URL http://dx.doi.org/10.1017/S1471068411000548.

Anna Fariha, Ashish Tiwari, Arjun Radhakrishna, Sumit Gulwani, and Alexandra Meliou. Conformance constraint discovery: Measuring trust in data-driven systems. In *Proceedings of the 2021 International Conference on Management of Data*, pages 499–512, 2021.

Dan Feldman. Introduction to core-sets: an updated survey, 2020. URL https://arxiv.org/abs/2011.09384.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. What makes a good order of examples in in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.884. URL https://aclanthology.org/2024.findings-acl.884/.

Emmanuel Hebrard, Brahim Hnich, Barry O'Sullivan, and Toby Walsh. Finding diverse and similar solutions in constraint programming. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*, AAAI'05, page 372–377. AAAI Press, 2005. ISBN 157735236x.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In Bruce W. Croft and C. J. van Rijsbergen, editors, *SIGIR '94*, pages 3–12, London, 1994. Springer London. ISBN 978-1-4471-2099-5.

Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *CoRR*, abs/1611.08562, 2016. URL http://arxiv.org/abs/1611.08562.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 510–520, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1052/.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

R. Monarch, R. Munro, and C.D. Manning. *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI*. Manning, 2021. ISBN 9781617296741. URL https://books.google.com/books?id=LCh0zQEACAAJ.

Zafeiria Moumoulidou, Andrew McGregor, and Alexandra Meliou. Diverse data selection under fairness constraints. *CoRR*, abs/2010.09141, 2020. URL https://arxiv.org/abs/2010.09141.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 287–294, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130417. URL https://doi.org/10.1145/130385.130417.

Ananya Singha, Mukul Singh, Ashish Tiwari, Sumit Gulwani, Vu Le, and Chris Parnin. TeCoFeS: Text column featurization using semantic analysis. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7055–7061, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. URL https://aclanthology.org/2025.findings-naacl.392/.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424, 2016. URL http://arxiv.org/abs/1610.02424.

Song Wang, Zihan Chen, Chengshuai Shi, Cong Shen, and Jundong Li. Mixture of demonstrations for in-context learning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. Using document summarization techniques for speech data subset selection. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 721–726, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1086/.

Lining Zhang, Simon Mille, Yufang Hou, Daniel Deutsch, Elizabeth Clark, Yixin Liu, Saad Mahamood, Sebastian Gehrmann, Miruna Clinciu, Khyathi Raghavi Chandu, and João Sedoc. A needle in a haystack: An analysis of high-agreement workers on MTurk for summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

14944–14982, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.835. URL https://aclanthology.org/2023.acl-long.835/.

A Evaluation Datasets

The specific text classification benchmarks we picked for creating benchmarks for ordered diversity sampling are listed below:

- hf://datasets/coastalcph/lex_glue/,
- hf://datasets/google/frames-benchmark/,
- hf://datasets/dair-ai/emotion/,
- hf://datasets/google-research-datasets/go_emotions/,
- hf://datasets/nyu-mll/multi_nli/ (2),
- hf://datasets/nvidia/Aegis-AI-Content-Safety-Dataset-1.0/,
- hf://datasets/declare-lab/HarmfulQA/data_for_hub.json (2),
- hf://datasets/fancyzhx/ag_news/,
- hf://datasets/cardiffnlp/tweet_eval/,
- hf://datasets/zeroshot/twitter-financial-news-topic/,
- hf://datasets/SetFit/bbc-news/,
- hf://datasets/ttxy/emotion/,
- hf://datasets/zeroshot/twitter-financial-news-sentiment/,
- hf://datasets/jonathanli/law-stack-exchange/,
- hf://datasets/allenai/prosocial-dialog/,
- https://www.kaggle.com/datasets/urbanbricks/wikipedia-promotional-articles,
- https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset/data,
- https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb,
- https://www.kaggle.com/datasets/arplusman/papers-by-subject,
- https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment

B Comparison using TF-IDF embedding

Figure 2 shows the same plot as the plot in Figure 1, but with OpenAI text embedding replaced by TF-IDF embedding (for all procedures).

Figure 3 shows a plot comparing the versions of both our procedures with two different embedding procedures. When we compared the performance of Principled Samplers v1 and v1 while changing the embedding, we observed that the text-embedding model from OpenAI performs better than TF-IDF as expected. We also noted that changing the embedding affects Version v1 and v2 in the same way, although Version v2 appears slightly more robust to embedding change.

C Time Comparison

We compare the time taken by the different diverse sampling schemes to sample 60 points as the size of the dataset grows from 250 to 3000. Figure 4 shows that Principled Sampler v1 is the most efficient. This is surprising since PCA can be expensive; however, greedy k-center (k-ctr) and reverse semantic search (rss) have a non-parallelizable iterative loop that makes them scale poorly. On the other hand, our Versions v1 and v2 can pick each sample almost independently of the other in parallel. We observe that clustering (cls) has time efficiency competitive with our methods. The bottom chart in Figure 4 plots the time taken to sample n points from a dataset of 2500 points as n grows from 18 to 72. We again observe the same behavior as in the earlier plot.

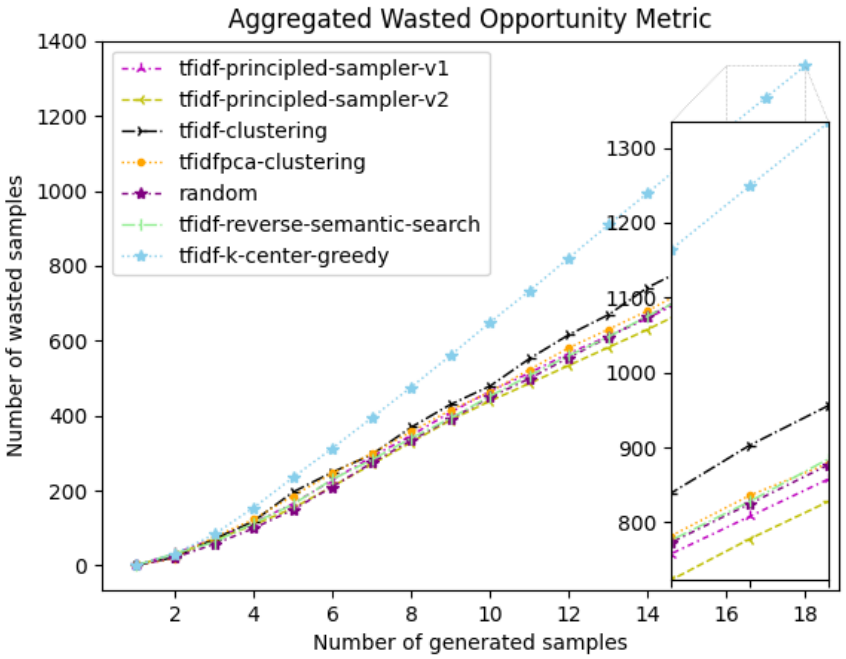

Figure 2: Comparing with baselines using $3n = 18$ samples using tfidf embeddings.

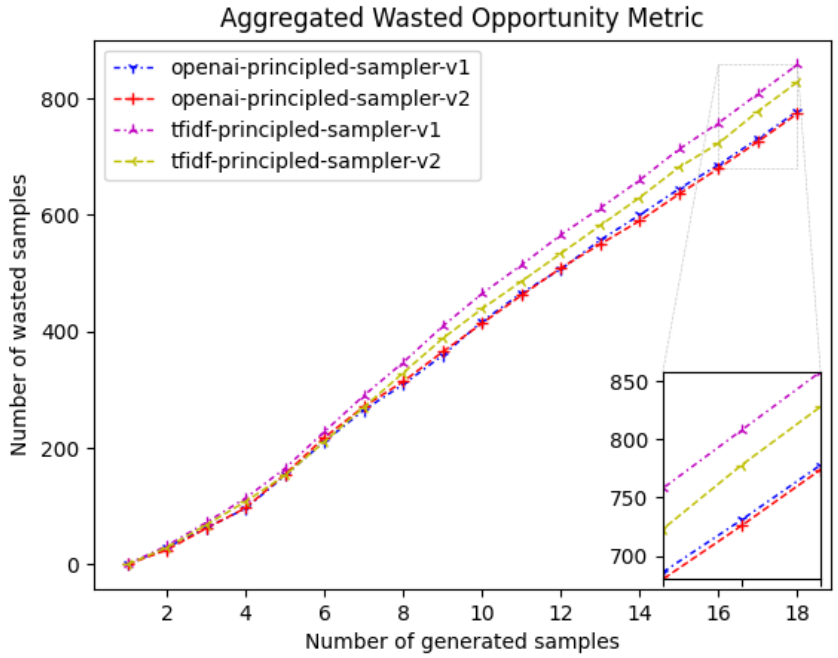

Figure 3: Comparing Samplers v1 and v2 with openai embedding and with TFIDF-based embedding.

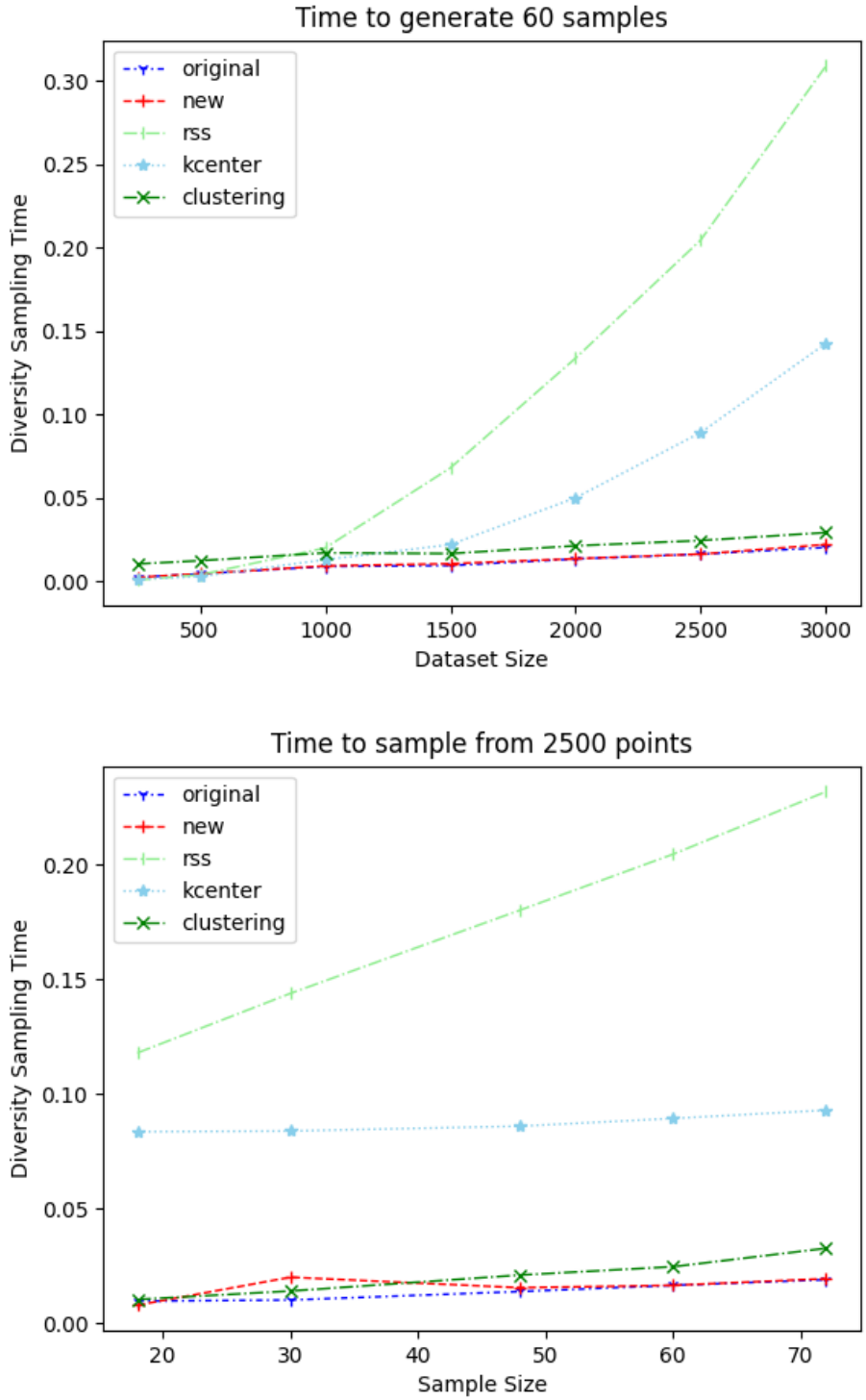

Figure 4: Time taken by different methods (a) to sample 60 from datasets of different sizes (b) to sample different number of items from a dataset with 2500 items.