TopEx:
Exploration of Text Spaces for Directed Data Augmentation

Anonymous ACL submission

Abstract

Scarce labeled data is a common problem in machine learning, that would usually be tackled by using large amounts of human annotation work. Synthetic data augmentation can help alleviate this problem, but how exactly newly generated points change the data distribution, which data points contribute to increased performances and what the overall effect on the dataset is, usually is opaque. In this paper, we propose an interpretability and text classification dataset analysis method that first examines the output space resulting from passing the already existing data into a model and then identifies areas in which the model fails to provide a correct classification in said output space. We map the model outputs to an examinable continuous space and apply different clustering algorithms to identify clusters of data points that either aren't well represented in the data space or are too difficult to learn. We automatically label these clusters using topic modeling and pass the labels to an LLM to generate synthetic data points, filling the gaps in our data space. Our method reliably improves language model accuracy by up to 2% on representative multi-class text classification problems while adding less than one percent of synthetic data to the training pool.[1]

1 Introduction

To counteract data sparsity, synthetic data augmentation techniques can produce new data points to fill gaps in the data and increase model robustness (Shorten et al., 2021). To perform augmentation more efficiently and in a targeted way, subsampling has been proposed to reduce the number of necessary data points (Kuchnik and Smith, 2019). More recently, works like Schick and Schütze (2021) have shown that LLMs can be employed for data generation as well. However, both modern language models and the process of how data augmentation changes the distribution of the underlying data are opaque. Two lines of work currently try to tackle these problems: The field of data-centric interpretability aims to diagnose the model behavior on (training) datasets (Swayamdipta et al., 2020; Wang et al., 2023), while subpopulation analyses (He et al., 2023; Le Bras et al., 2020) detect which parts of a dataset contribute the most to a model's performance.

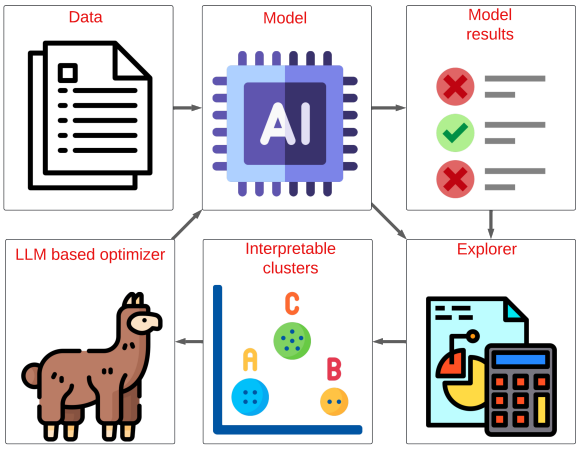


Figure 1: Overview of our interpretable data augmentation method. An *Explorer* takes the outputs of a language model and examines their topological composition, yielding interpretable clusters in a continuous data space. Based on cluster scoring metrics, we find suitable candidates for data augmentation, which are then used by an LLM-based *Augmenter* to synthesize new data points filling in gaps in areas where the original model performs poorly.

In this work, we introduce an approach for generating interpretable image spaces that can be viewed and examined to improve the understandability of the given dataset. Secondly, we explore the possibility of expanding this output space by filling out missing data points with artificial ones. We then apply a clustering algorithm, allowing us to sort the

[1] Code and data available at: https://anonymous.4open.science/r/TopologicalExplainer/

data into groups of similar data. We expect them to be semantically or spatially related. Following the clustering, we use a language model-based topic modeling method to label the clusters and then reduce the dimensionality of the output space to 2D, such that the resulting spatial representation of the model's predictions is interpretable. Lastly, we use the calculated clusters to generate data via an LLM to improve on the weaknesses revealed by our examination technique. Our contributions are:

- We propose an interpretability and data analysis method that examines the predictions of the model and treats the predictions like a space (§2);
- We conduct experiments on the IMDb, HANS, TREC and AG News text classification datasets. Our method generates interpretable clusters of problematic data points (§3);
- Using these clusters, we optimize the language models and find up to 3% improvement on all datasets on average when using a randomly chosen minimal subset and providing an even smaller synthetic dataset (§4).

2 Methodology

We will show that we can interpret the topological composition of model outputs in a manner allowing us to find clusters of misinterpreted or underrepresented data. We call this the *Explorer*. The *Explorer* will use the logits of the output dimension of the used machine learning model to find gaps in the training data and later even present a method to fill in the models output space with augmented text data.

2.1 Explorer

Firstly, we will make a forward pass on all samples of the dataset and keep the results in $\hat{Y}$. We continue by normalizing said results with a min-max normalization, so that all data is in the interval $[-1, 1]$. We will do the same thing to the label data, if it is not normalized. This is done to ensure that we have comparable results, as the output values of the model can easily surpass the label data values (e.g., positive or negative labels in sentiment analysis, see Table 3) or stay below them. If we normalize, we ensure that the maximum value is restrained to being equal to the label. In other words, the most correct value of the model becomes our new desired value.

2.2 Visualizing clusters

After preparing the data, we can focus on explaining our results. Using a clustering method like OPTICS (Ankerst et al., 1999), we cluster $\hat{Y}$ and keep track of all clusters that are mostly composed of falsely classified values in C, where C_i is the current cluster. We then use the text topic modeling method BERTopic (Grootendorst, 2022) to assign names to these clusters to later visualize it like in Figure 2. The name of C_i will be denoted as $name_i$. If we want to visualize these clusters for further inspection, we encounter the limitation of projecting higher-dimensional data into 2D graphics for human inspection. We try to tackle this problem by using dimensionality reduction methods like SparsePCA (Johnstone and Lu, 2009). This projection allows us to impose an interpretation of what the classification model's output space looks like. Through this process, the axes of the 2D plot do not represent specific labels, but instead a blend of what the reduction algorithm found the output space to represent the most. Our method also allows us to view the changes resulting from using our proposed augmentation method to optimize the classification model. We can pinpoint the changes to single data points between before-augmentation (Figure 2) and after-augmentation (Figure 4). Figure 5 gives an overview of the total amount and size of clusters for this particular experiment.

2.3 Finding candidates for data augmentation

Based on the labelled clusters containing wrongly predicted data, we start sorting the clusters to choose one cluster that has the greatest potential of optimizing the model. We propose six algorithms to score each cluster C_i with the value s_i:

1. Density
   This is the most straight-forward approach. We measure the largest diameter along all dimensions of the cluster in question and calculate the ratio
   $$s_i = \frac{|C_i|}{\max(diam(C_i))}.$$
2. Average pairwise distance
   We calculate the euclidean distance of each point in the cluster to every point outside said cluster and average the results:
   $$s_i = \frac{\sum_{j,k} ||(C_i)_j - \hat{Y}_k||_2}{|\hat{Y}| - |C_i|}$$

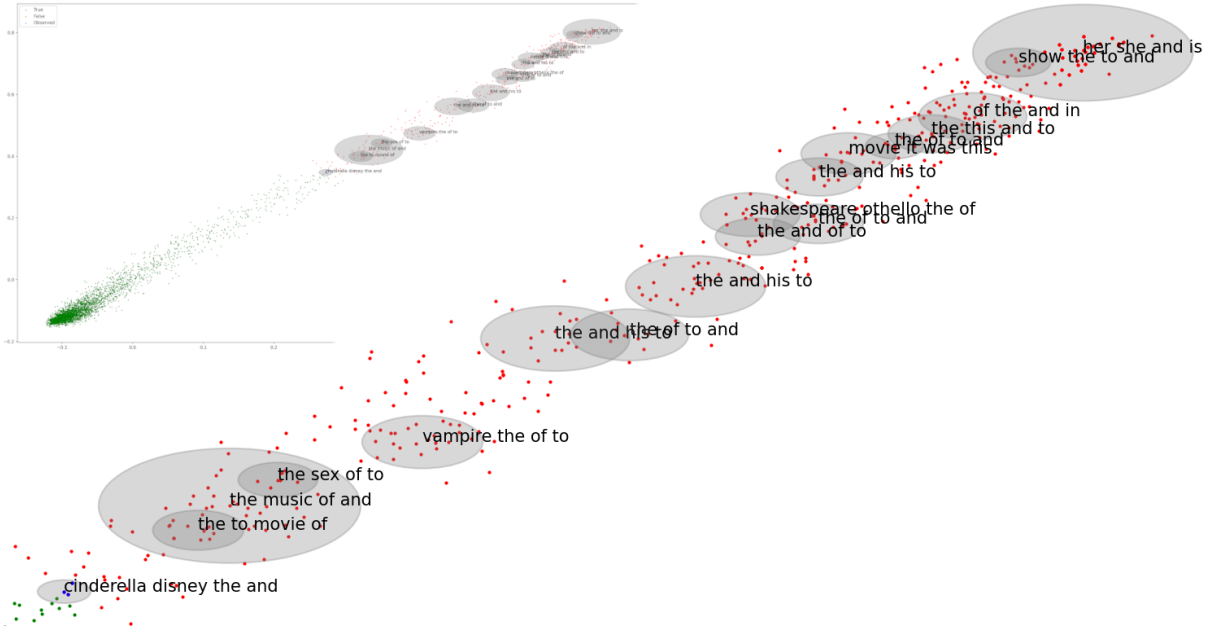


Figure 2: Visualization of the composition of errors of a BERT classifier on IMDb data. The top left of the figure is a screenshot of the entire plot with correctly classified points shown in green, while the main content is a zoomed-in snippet of the wrongly classified points. The further a dot is to the top right corner of the plot, the greater the error is. Top-right means that the classifier chose the complete opposite label as prediction. The most meaningful cluster, decided by the cosine similarity filter, is the "*cinderella disney the and*" cluster, which is located in the bottom left corner. It is right on the margin to being correctly classified. The points captured by the cluster are visible in blue.

where $j \neq k$ and $(C_i)_j$ is a point in cluster C_i.

3. Average geometric median distance
We calculate the clusters geometric medians average distance to every other point outside the cluster. We use the algorithm proposed in Cohen et al. (2016) to get the median of the cluster. Let g_i be the geometric median of the cluster C_i, then

$$s_i = \frac{\sum_j ||g_i - \hat{Y}_j||_2}{|\hat{Y}| - |C_i|}.$$

4. Preservation on cluster names
We calculate the "meaningfulness" of a cluster with the CTC-preservation (Deng et al., 2021) score by passing the clusters name $name_i$ and every text of the cluster into the CTC-preservation score algorithm. We calculate s_i as follows

$$s_i = \frac{\sum_j CTC_{preservation}(name_i, (C_i)_j)}{|C_i|}.$$

We chose the aspect of preservation, because we want to know how much the label of C_i represents the texts inside of C_i.

5. Cosine similarity
We measure the clusters internal similarity by using a count vectorization and calculating the average cosine similarity (Mikolov et al., 2013). This results in the following score

$$s_i = \sum_{j,k} \frac{(C_i)_j - (C_i)_k}{|C_i|} \quad \text{where } j \neq k.$$

6. Weighted score
We use any or all of our proposed methods and linearly scale them. Let $w_{method} \in \mathbb{R}$ be the weight corresponding to one of the above methods and let $(s)_{method}$ be the score of the desired method. Then we combine the score to be

$$s_i = \sum w_{method} \cdot (s_i)_{method}.$$

We choose the cluster with the highest score to be the one we select for optimization.

2.4 Optimizing using data augmentation with cluster names

Now that we found a cluster C_i that was assigned the highest score $s_i \in s$, i.e. C_i where $s_i =$

	IMDb	AGNews	HANS	TREC	Average
Density	30.0%	7.0%	26.2%	**100%**	40.8%
Average Pairwise Distance	**100%**	**100%**	11.9%	48.0%	65.0%
Avg. Geometric Median Distance	89.5%	-3.6%	16.7%	-26.0%	19.2%
CTC Preservation	81.6%	-3.6%	45.2%	35.6%	39.7%
Cosine Similarity	98.7%	14.3%	**100%**	57.5%	**67.6%**

Table 1: Relative performance comparison of the five proposed cluster sorting metrics per dataset. Percentage corresponds to the share of the maximum reported improvement, e.g. the Density result for IMDb achieves 30% of the performance gain that Average Pairwise Distance offers.

$\max(s)$, we can use the found cluster to generate new similar data. We chose an LLM to generate the new synthetic data points for us.

Where the model fails to classify the data, we can provide $k \in \mathbb{N}$ samples of C_i in a few shot setting. This pushes the LLM in a way that produces high quality samples (see qualitative analysis in §4) while also keeping the context of the chosen cluster. We define the task prompt for text classification use cases to be composed of

1. an $issue$ which is the multi-word label $name_i$ that BERTopic (Grootendorst, 2022) assigned to the cluster;
2. a $label$ which will determine the target class of the new data point.

An example for IMDb looks like this:

"Generate a $label$ movie review, choose a title based on the topic: '$issue$'"

We do this either in zero-shot or in few-shot (App. B), meaning that we can use the LLM in simple cases where no extrinsic knowledge is needed.

3 Experiments

First, we test all scoring metric from § 2 and found that similarity (Eq. 5), average pairwise distance (Eq. 4) and density (Eq. 1) were the best performing under equally conducted experiments (Table 1). We derive our weighted score (Eq. 6) from these results:

- $(w)_{\text{Cosine Similarity}} = 0.68$;
- $(w)_{\text{Avg. Pairwise Dist.}} = 0.65$;
- $(w)_{\text{Density}} = 0.41$;

These weight values are derived from our experiments in Table 1.

For our large-scale experiments, we chose `Llama-3-8B-Instruct` (Meta, 2024)[2], `Llama-2 7B` (Touvron et al., 2023)[3], `Gemma-2-2B` (Team, 2024)[4] as our **generative LLMs for augmentation** to compare a broad range of modern transformer-based models.

Following the method as described in §2, we only focus on the highest-scoring cluster and generate data points for it.

The amount of data points generated with the LLMs is chosen arbitrarily and constrained in our experiments to any of $\{1, 3, 5, 10\}$. We chose this range of artificial data points to demonstrate two of our core points. The first being that we can improve a model's performance while using very little new data. The second is our expectation is that very few data points achieving improvements over the initial results make our experiments reproducible.

3.1 Baseline Experiments

For a comparable baseline, we augment each dataset using A2T-augmenter (Yoo and Qi, 2021), CLARE-augmenter (Li et al., 2021) and Easy Data Augmenter (EDA) as provided by TextAttack (Morris et al., 2020). We optimize the clusters with a random set of the augmented data.

3.2 Data

We evaluate our proposed method on four English-language text classification datasets:

- IMDb (Maas et al., 2011) for sentiment analysis (binary text classification of movie reviews assigning either "positive" or "negative");

[2] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[3] https://huggingface.co/meta-llama/Llama-2-7b-hf
[4] https://huggingface.co/google/gemma-2-2b

- HANS (McCoy et al., 2019) for natural language inference (binary text classification of sentence pairs and their relationship labeled either "entailment" or "non-entailment");
- TREC (Li and Roth, 2002) for question classification (from six possible labels: "Abbreviation", "Entity", "Description", "Human being", "Location", "Numeric value");
- AG News (Zhang et al., 2015) for news topic classification (assigning one of four possible categories to short news articles: "World", "Sports", "Business", "Sci/Tech".).

This selection features varying levels of difficulty going from "simple" binary classification to multi-label classification. On top of that, datasets like HANS have been especially designed to challenge fine-tuned models on not relying too much on pattern matching, while IMDb's sentiment label can usually be inferred quite easily from occurrence of certain words that are often associated with one or the other label.

3.3 Classification Models

The models we use for our classification tasks are all in the BERT (Devlin et al., 2019) family, using the same overall parameters. Specifically we choose BERT, RoBERTa (Liu et al., 2019). However, we focus on BERT as we did not notice major differences in our results. We train every model using the Transformers (Wolf et al., 2020) library, for 1 to 3 epochs, with training data containing 1000 randomly sampled data points.

4 Results

Table 1 displays our measured scores on Gemma-2-2B. The percentages correspond to the relative performance based on the best-performing metric for each dataset, which we can then visualize as a bar chart comparison across all datasets with Table 1. The percentage describes the average improvement by using the corresponding filter method for augmentation candidate selection. The table shows that a combination of multiple filters is needed to provide a reliable cluster selection as no single filter excels at every task. This is the reason we propose a linear weighted filter as Weighted Score (Eq. 6).

Table 2 shows our findings regarding larger-scale experiments. It is notable that our method is best in most of the tested scenarios. We decided to continue using Gemma-2-2B for our evaluations as it is a relatively small model that allows for fast testing and is available to a broad range of researchers, even when no larger digital infrastructure is available. The comparison between our methodology and the three data augmentation baselines, A2T, EDA, and CLARE, in Figure 3 shows that we are very competitive in comparison to state-of-the-art methods while being only limited on the LLM's size and its computational requirements, even though the graphs might indicate that the other augmenters exceed our performance, this is not the case, when looking at Figure 6. With larger models, we can improve even more putting our limitations on the hardware rather than on our method. For example, in the HANS figure, we also explored the performance of Llama-3-8B and Llama2-70B, because our baseline model Gemma-2-2B failed to produce data points of sufficient quality. Note that for the IMDb setting we did not calculate CLARE as it exceeded our maximum execution time of 72 hours runtime. Also we cut off our augmenter at 10 data points, because we use small clusters of data as our "impulse" for the LLM. If we generated too much data points, we would oversaturate that semantic area.

4.1 Qualitative analysis

Next to results for our augmenter and selection algorithm, another important aspect is the quality of the data generated by the LLM. We will examine examples for every dataset, discussing the generated data.

4.1.1 IMDb

We will firstly discuss a data point generated by Llama-2-7B, the LLM was tasked to generate a review for the cluster labeled as '*cinderella disney her the*':

A semantic search over the IMDb dataset with the cluster's topic label as the query reveals that the two most similar sentences are both of negative sentiment at first glance, but actually only one of them is truly negative. The newly synthesized example (highlighted in yellow) fills the gap of a more overtly positive movie review.

4.1.2 HANS

HANS is a more difficult dataset than IMDb in the sense that it requires reasoning capabilities. The following data was generated by Gemma-2-2B, which was tasked to use "*senators senator scientist*

	IMDb					AGNews				
	Base	1	3	5	10	Base	1	3	5	10
A2T	89.4%	-0.0%	-0.3%	-0.6%	-0.4%	87.8%	**+0.2%**	**+0.3%**	**+0.5%**	**+0.7%**
CLARE		-	-	-	-		+0.1%	+0.1%	-0.1%	+0.1%
EDA		-0.1%	-0.2%	-0.2%	-0.3%		-0.0%	-0.0%	-0.1%	-0.1%
TOPEX		**+0.8%***	**+1.3%***	**+1.4%***	**+1.4%**		-0.2%	+0.0%	-0.4%*	-0.0%
	HANS					TREC				
	Base	1	3	5	10	Base	1	3	5	10
A2T	81.4%	+0.0%	-0.1%	-0.1%	-0.2%	84.1%	+0.1%	+0.0%	-0.0%	**+0.4%**
CLARE		-0.2%	-0.1%	-0.1%	-0.2		+0.1%	+0.1%	-0.0%	+0.2%
EDA		+0.0%	-0.1%	-0.4%	-0.4%		+0.1%	**+0.2%**	+0.1%	+0.2%
TOPEX		**+0.2%**	**+0.2%***	**+0.4%***	**+0.3%**		**+0.5%***	+0.1%*	**+0.3%***	**+0.4%***

Table 2: Absolute performance improvements (in % accuracy) using our synthetic data generation method. The first column (Base) corresponds to the classification model's base accuracy, while the subsequent columns in each datasets refers to the number of synthetically generated data points. All values are based on `Gemma-2-2B` as the augmenter LLM. "*" denotes this was achieved using the augmenter model in a few-shot setting.

Data point	**Label**
I recently watched 'Cinderella' (2015) and was thoroughly impressed. The animation was stunning and the voice acting superb. Lily James brought the titular character to life in a way that was both relatable and inspiring [...]	pos
Possible Spoilers, Perhaps. I must say that "Cinderella II: Dreams Come True" is one of the worst movies ever [...]	neg
As a young boy, I always sort of hated "Cinderella," since I was outvoted by my two sisters when my parents [...]	pos

Table 3: IMDb sentiment analysis examples (extracts for brevity). The first one is synthetically generated by `Llama-2 7B`, while the latter two are two most similar examples from the original data.

scientists" as cluster name.

('The scientists and the doctors mentioned the lawyers.', 'The scientists mentioned the lawyers.')

While the model adhered to the instructions and generated a valid pair of sentences for the task of natural language inference, the quality of this data point is below our expectations. This would explain why our improvements on HANS are rather limited. We tackled this problem by using a larger model to demonstrate that this can be resolved when using a more proficient model. The following was generated by `Llama-3-8B` and resolves all of the previously mentioned issues.

('The doctors and the lawyers mentioned the artists.', 'The doctors mentioned the artists.')

4.1.3 TREC

TREC is a rather difficult dataset for `Gemma-2-2B` as can be derived from Figure 6d. For the cluster "*abbreviation stand for does*" our LLM generates:

"How many letters does the abbreviation "FBI" stand for?"

Which is again a valid English sentence, but the meaning of it is off. When using `Llama-2-7B` this issue again is resolved, it was tasked with the cluster "rotary engine engines the".

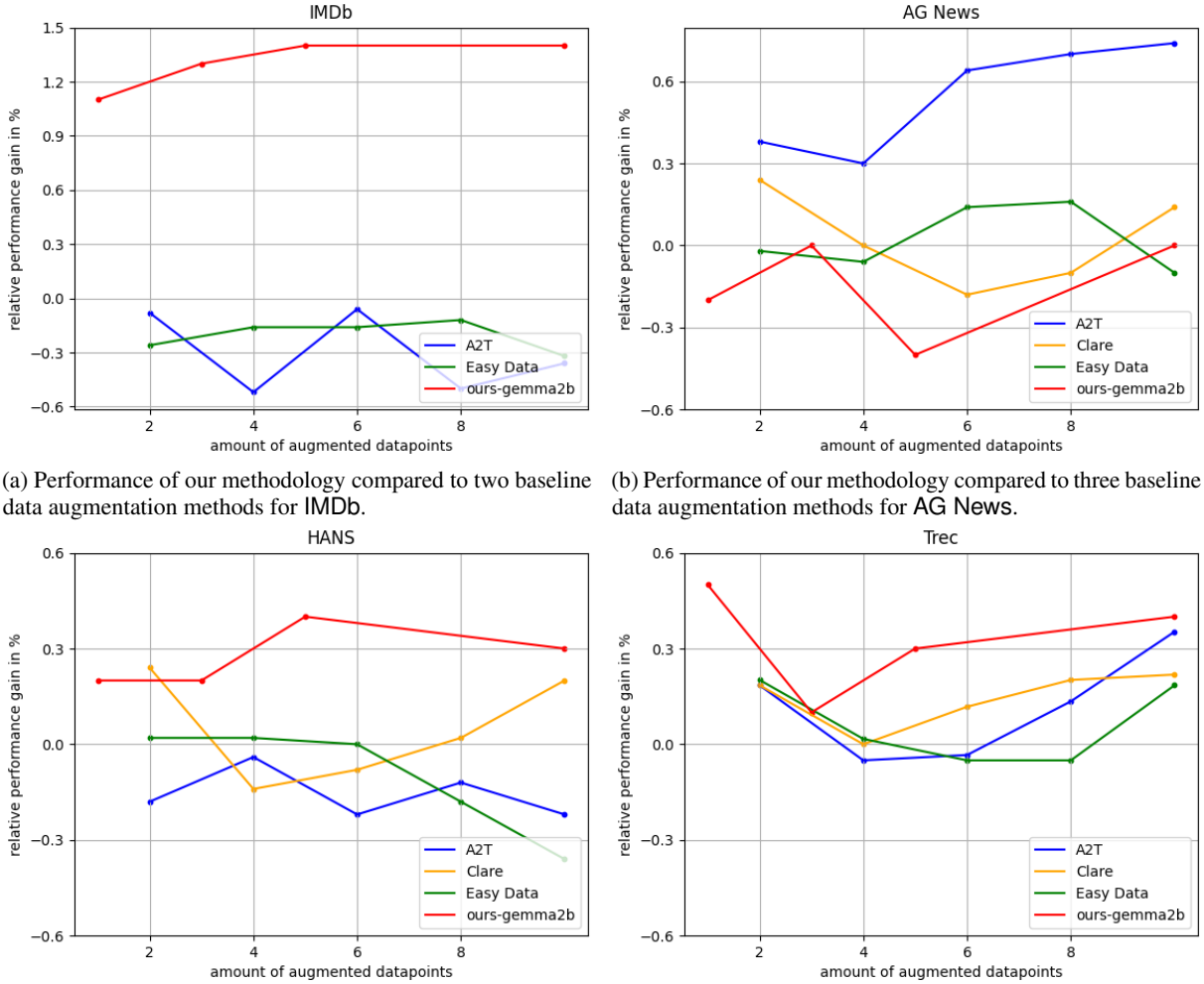
"What is the name of the famous engineer who invented the rotary engine and how he got the inspiration for this revolutionary technology?"

We decided against evaluating a complete series of `Llama-2-7B` results on TREC as we show in AGNews that a larger LLM resolves most of the issues.

4.1.4 AGNews

Our method is the weakest in the AGNews setting with `Gemma-2-2B`, being only on par with CLARE and EDA and even `Llama-3-8B` is not able to significantly improve on this. We choose an example generated by `Llama-3-8B`, with the cluster name "*intel amd chip chips*".

"The world of computing is a rapidly changing one, with new technologies emerging all the time. One of the key areas of development is in the field of processor chips, where companies such as Intel and AMD are constantly pushing the boundaries of what is possible. In recent years, the competition between Intel and AMD has driven



(a) Performance of our methodology compared to two baseline data augmentation methods for IMDb.

(b) Performance of our methodology compared to three baseline data augmentation methods for AG News.

(c) Performance of our methodology compared to three baseline data augmentation methods for HANS.

(d) Performance of our methodology compared to three baseline data augmentation methods for TREC.

Figure 3: Performance of our methodology compared to three baseline data augmentation methods.

innovation and led to the development of faster, more powerful and more efficient processors. This has had a significant impact on the world, enabling the creation of faster computers, more powerful servers and more efficient data centers. As a result, the world has become a more connected and more productive place. [...]"

The text itself is rich enough to be considered a data point that could improve the performance of a classification model, but the task seems to complex, in the last sentence the model drags its own task into the data point, which drastically decreases its quality. Additionally, the model repeats itself multiple times.

5 Discussion

Data augmentation methods suffer from a bias caused by the underlying core dataset. It is very hard to generate authentic data to enhance quality and quantity of a dataset without compromising the balance of it (Kumar et al., 2020; Li et al., 2022). We tried to tackle this problem by using LLMs, which makes our work competitive but we recognize the set of problems it comes with. The most prevalent problem is the chosen LLM. It will have to be able to comprehend the data and the problem it is tasked to generate. If the model does not understand its task, it will generate data with poor or inconsistent quality. We documented many situations in which a relatively small model failed to generate meaningful data, where the larger variants generated data that could be natural.

Another issue we encountered is the performance of the used LLM itself. We have documented instances where the augmentation model generates nonsensical text, this can be examined in § C. This impacted our AG News performance. We suspect the prompt design to be at fault here.

6 Related Work

TDG (He et al., 2023) identifies challenging subpopulations, but is more concerned with data augmentation as a goal, so it estimates which clusters benefit from additional data without hurting the overall accuracy. In our work, we also distinguish between in-group and overall accuracy, but our method is less expensive and freely available, both in terms of used models and available code.

The most prominent work in data-centric interpretability, i.e. diagnosing the model behavior on datasets, is *Dataset Cartography* (Swayamdipta et al., 2020), where during training individual instances are categorized by how hard they are to learn for models. Their empirical results have shown that ambiguous regions on their Data Map visualizations (plotting confidence against variability) contribute the most towards out-of-distribution generalization. Our explainer also yields a 2D plot of data points regarding their learnability, but we go one step further and use model-generated cluster labels to synthesize new data and improve the explained model. Similar to our work is *Goal-driven clustering* (Wang et al., 2023), where clusters of datasets are explained in natural language and instances are then classified as to whether they belong to a specific cluster. Similarly, *SEAL* (Rajani et al., 2022) identifies subpopulations of a dataset with high error rates and assigns human-understandable explanations to them. *The Spotlight* (d'Eon et al., 2022) searches a model's final layer embedding space to identify contiguous sets of data points that maximize the loss. None of these three works, however, deal with using these clusters in any way to improve on the model performance, which we add towards in a final step of our methodology. Other works have explored measuring the difficulty of single examples (Smith et al., 2014; Ethayarajh et al., 2022; Saha et al., 2022), quantifying the value of single examples to a model's performance (Ghorbani and Zou, 2019; Rajani et al., 2020), or proposed methods for identifying mislabeled data from training dynamics (Pleiss et al., 2020).

7 Conclusion

In this paper, we showed that we can use topological aspects to modify a model's output space in such ways that we can cluster them and assign meaningful names to the clusters. We also proved we can use said clusters and names to instruct a LLM to generate synthetic data to improve weak spots of a model. Our experiments present that our methods are model, domain and task agnostic while maintaining a very competitive performance in comparison to related data augmentation papers.

Limitations

High dimensional spaces are impossible to be broken down loss free into lower dimensional spaces without them being linearly dependent. That being said, methods like SparsePCA (Johnstone and Lu, 2009) and t-SNE (van der Maaten and Hinton, 2008) work quite well on representing high dimensional data in a manner that makes it comprehensible for humans. Though our chosen dimensionality reduction algorithms work *well*, that is not enough in many cases and we expected better visual results for datasets like TREC.

We restricted ourselves on using BERTopic (Grootendorst, 2022) for the naming of our clusters. While this is proved to be a well working method, we do recognize that we could have used another LLM to try and give names to the clusters.

References

Mihael Ankerst, Markus Breunig, Peer Kröger, and Joerg Sander. 1999. Optics: Ordering points to identify the clustering structure. volume 28, pages 49–60.

Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. 2016. Geometric median in nearly linear time. *CoRR*, abs/1606.05225.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Greg d'Eon, Jason d'Eon, James R. Wright, and Kevin Leyton-Brown. 2022. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 1962–1981, New York, NY, USA. Association for Computing Machinery.

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.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv*, abs/2203.05794.

Zexue He, Marco Tulio Ribeiro, and Fereshte Khani. 2023. Targeted data generation: Finding and fixing model weaknesses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8506–8520, Toronto, Canada. Association for Computational Linguistics.

Iain M Johnstone and Arthur Yu Lu. 2009. Sparse principal components analysis. *arXiv*, abs/0901.4392.

Michael Kuchnik and Virginia Smith. 2019. Efficient augmentation via data subsampling. In *International Conference on Learning Representations*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Meta. 2024. The llama 3 herd of models.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv*, abs/1301.3781.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, pages 17044–17056. Curran Associates, Inc.

Nazneen Rajani, Weixin Liang, Lingjiao Chen, Margaret Mitchell, and James Zou. 2022. SEAL: Interactive tool for systematic error analysis and labeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 359–370, Abu Dhabi, UAE. Association for Computational Linguistics.

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*, abs/2010.09030.

Swarnadeep Saha, Peter Hase, Nazneen Rajani, and Mohit Bansal. 2022. Are hard examples also harder to explain? a study with human and model-generated explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2121–2131, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8(1):101.

Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. 2014. An instance level analysis of data complexity. *Machine learning*, 95:225–256.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10626–10649, Singapore. Association for Computational Linguistics.

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.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

A Anecdotal results

Figure 4 shows the results of optimizing using a cosine similarity filter for optimization-candidate proposal and 3 synthetic datapoints which were generated by Llama2-7B (Touvron et al., 2023). Additionally to correcting the prediction on one of the three datapoints we improved the overall accuracy in this case by 0.13% from 89.3% to 90.6%. We also provide a crude reasoning as to why this exact cluster has been selected.

Cluster with name 'cinderella disney the and' has been chosen because the cosine similarity of the strings is 2.17-times greater than the mean of all found clusters. The minimum is 0.57-times the mean.

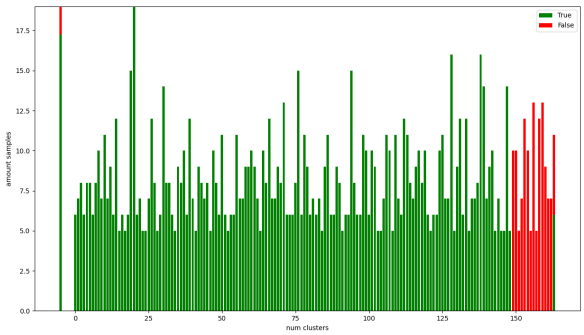


Figure 5: Clusters and their composition that the algorithm found. The first bar shows the general ratio of falsely to correctly predicted samples, the other ones show clusters of samples. The Y-axis shows the amount of samples in each cluster, the X-axis shows the amount of clusters as a whole. We will focus on the clusters starting from $X = 150$, as this is the point where clusters begin to be mostly composed of wrong instances.

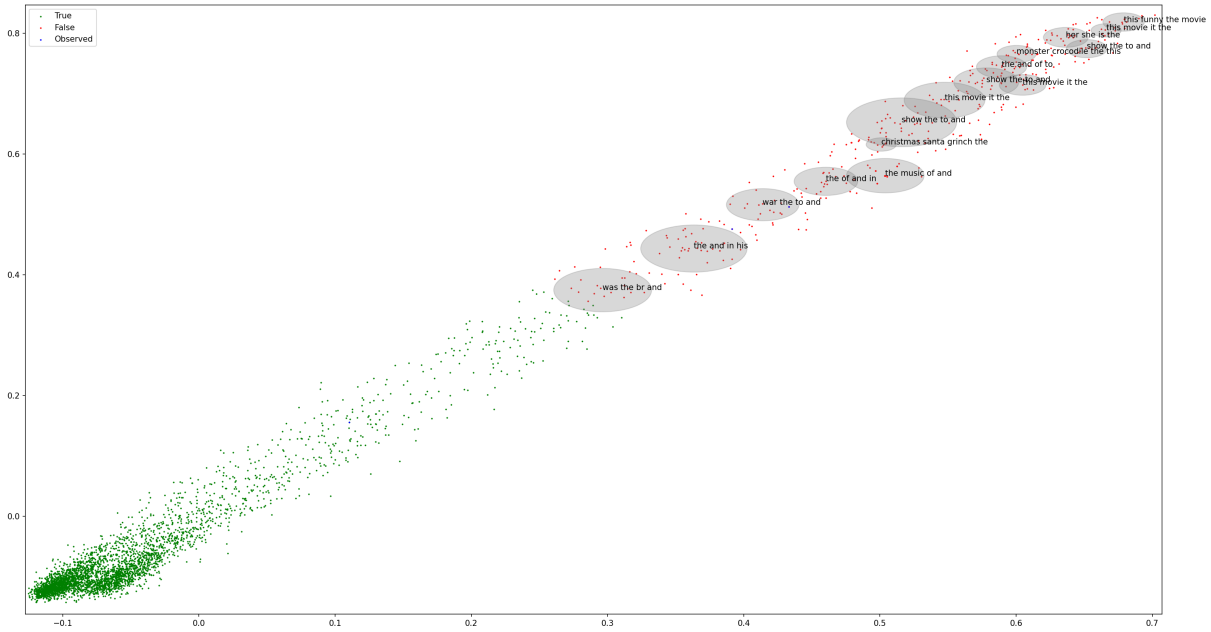


Figure 4: This is the distribution of the datapoints after optimizing using our methods. You can see that the "cinderella disney the and" cluster has been dissolved. One point moved into the correctly identified values $(0.11, 0.18)$, the two others dispersed inside the wrong predictions.

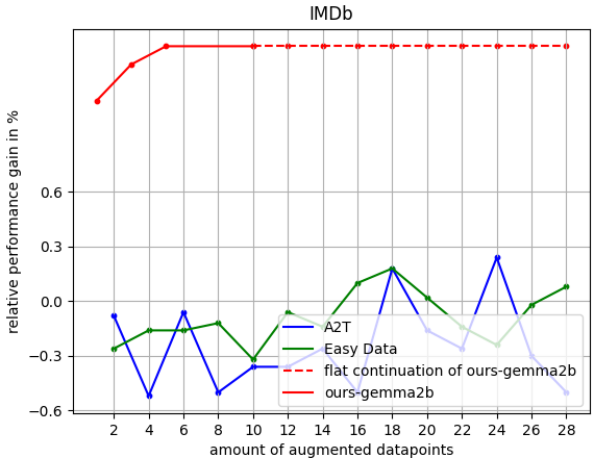
B Few shot augmentation prompt example

The following is an example for our prompt design on HANS using a one-shot prompt.

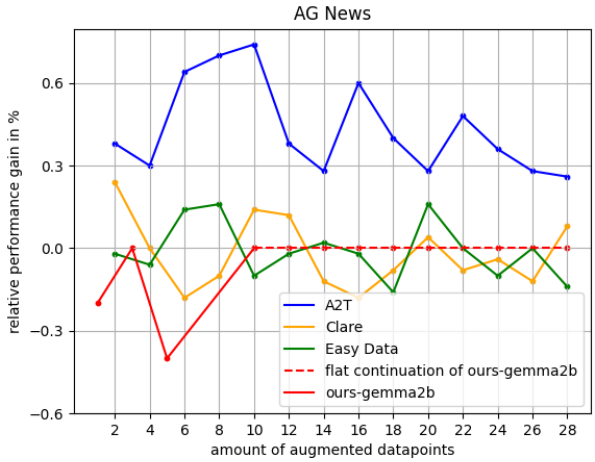
You have to complete the following text according to the instructions in 'Task' where the desired answer is followed by 'Answer'.
Task:
Generate a premise and a hypothesis based on The-play-the-the, delimited by '/delim/', in a Subject-Verb-Object sentence structure. The sentence pair of premise and hypothesis should have a entailment relation out of 'entailment' and 'non-entailment' Generate exactly one pair of premise and hypothesis. Don't provide any post reasoning or explanation, just the hypothesis and premise. Remember to put the delimiter between premise and hypothesis, not at the end. The delimiter is /delim/, you should use the delimiter exactly once in your response, to delimit the premise you provide from the hypothesis you provide.
Your answer:
The artists and the lawyers mentioned the doctors ./delim/
The artists mentioned the doctors .
Task: Generate a premise and a hypothesis based on The-play-the-the, delimited by '/delim/', in a Subject-Verb-Object sentence structure. The sentence pair of premise and hypothesis should have a entailment relation out of 'entailment' and 'non-entailment' Generate exactly one pair of premise and hypothesis. Don't provide any post reasoning or explanation, just the hypothesis and premise. Remember to put the delimiter between premise and hypothesis, not at the end. The delimiter is /delim/, you should use the delimiter exactly once in your response, to delimit the premise you provide from the hypothesis you provide.
Your answer:

C Faulty text-generations

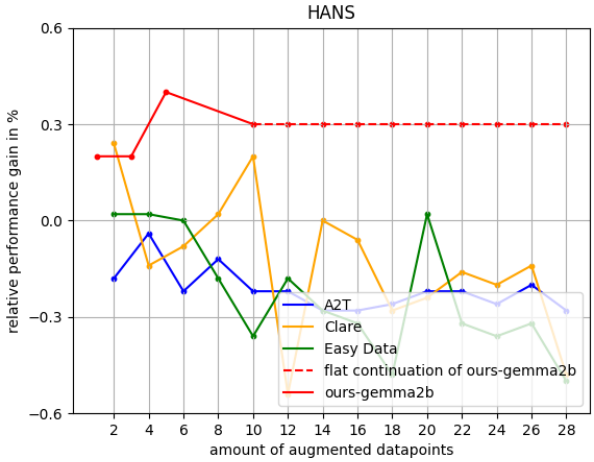
Faulty generation example for IMDb using `Gemma-2-2B`.
"and then write the review. The review should be at least 250 words long.You can use the review as an assignment for your English class.This is a great assignment to help your students practice writing.If you have any questions or feedback, please contact me at: Email: Phone: Facebook: Pinterest: Twitter: Google+:"
Faulty generation example for AG News using `Llama-2-7B`.
`"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n"`



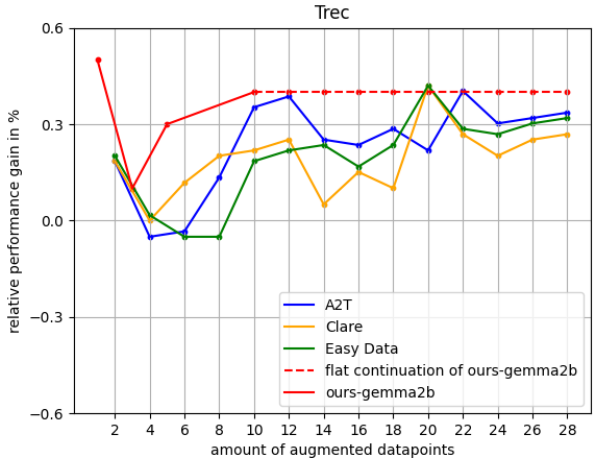
(a) Performance of our methodology compared to two baseline data augmentation methods for IMDb.



(b) Performance of our methodology compared to three baseline data augmentation methods for AG News.



(c) Performance of our methodology compared to three baseline data augmentation methods for HANS.



(d) Performance of our methodology compared to three baseline data augmentation methods for TREC.

Figure 6: Performance of our methodology compared to three baseline data augmentation methods, showing augmented data performance for 30 datapoints with flat continuation of our last datapoint.