Rethinking Data Augmentation in Text-to-text Paradigm

Yanan Chen
Dept. of Physics and Computer Science
Wilfrid Laurier University
chen0040@mylaurier.ca

Yang Liu
Dept. of Physics and Computer Science
Wilfrid Laurier University
yangliu@wlu.ca

Abstract

As manually labelling data can be costly, some recent studies tend to augment the training data for improving the generalization power of machine learning models, known as *data augmentation* (DA). With the arise of pre-trained language models (PLMs), some recent works on DA try to synthesize new samples benefiting from the knowledge learned from PLM's pre-training. Along the same direction, we in this paper propose to integrate text-to-text language models and construct a new two-phase framework for augmentation: 1) a *fine-tuning* phase where PLMs are well adapted to downstream classification with the help of two novel schemes, and 2) a *generation* phase where the fine-tuned models are leveraged to create new samples for performance lifting. This paradigm opens up a new way of designing fine-tuning scheme to better serve DA in an easy-to-implement manner, and can be easily extended to other desired tasks. We evaluate our proposal on two public classification datasets and demonstrate its effectiveness with remarkable gains.

1 Introduction

Due to the unique challenges of natural language processing tasks, there is no one-size-fits-all DA solution. Most early attempts are based on token manipulation (Wei and Zou, 2019; Kobayashi, 2018; Wu et al., 2019) or paraphrasing (Sennrich et al., 2016), and the boost is limited or marginal, or even diminishing or adverse especially given original training corpus is relatively sufficient or the backbone classifiers are PLM based, such as BERT or RoBERTa (Longpre et al., 2020).

Some researchers have shifted attention on applying generative language models(GLMs) for DA (Weng, 2022). Auto-regressive generation LMs such as GPT2 (Radford et al., 2019) are capable of generating text with high fluency and diversity, and therefore could serve as generators to synthesize new samples required by classification model training. However, most existing GLM-based DA solutions have some drawbacks. First, they fine-tune GLMs on the training corpus of limited capacity (Kumar et al., 2020; Anaby-Tavor et al., 2020; Liu et al., 2020), which can be problematic and prone to overfitting (Dodge et al., 2020; Phang et al., 2018; Ruder, 2021). Second, how to introduce external colossal online corpus freely available, such as reviews, comments and news to benefit GLMs to better serve DA has not been studied. Third, effective fine-tuning regime customized for data characteristic and structure has rarely been studied. In addition, recent works employing few-shot in-context generation for DA, such as GPT3, in avoidance fine-tuning and reap sparks of cleverness for automation, suffers from economic costs, latency in usage and lack of reliability (Sahu et al., 2022; Yoo et al., 2021; Wang et al., 2022).

To meet above challenges, we explore the potential of using text-to-text (seq2seq) language models, which have proved their success in many NLP tasks such as dialogue generation and machine translations. In the context of data augmentation, the original training samples can be regarded as the source text which sheds some light on the semantic meaning of the topic, whereas new synthetic samples will be considered as the target text induced by the source. Without loss of generality, we investigate the generation power of two exemplar text-to-text LMs: T5 and BART. Further, to cater for the text-to-text framework, we propose two fine-tuning schemes called *P2P* and *S2S* which organize the original corpus into parallel source and target text pairs, Different from many studies compromised by limited labelled data, large publicly available online corpus is adopted in the fine-tuning process. The proposed solution is evaluated on two text categorization tasks. Extensive experimental results prove these schemes can unleash the prowess of text-to-text generation while improving PLMs' generalization ability for DA.

Proceedings of the 29th International Conference on Computational Linguistics, pages 1157–1162
October 12–17, 2022.

2 Methodology

2.1 Problem formulation

Assume training data D_{train} contain a set of tuples $\{X^i, Y^i\}_{i=1}^{N}$ corresponding to word sequences and labels respectively. Our objective is to use text-to-text LMs denoted as $\mathcal{G}$ to produce synthetic training data $D_{syn} : \{\tilde{X^i}, Y^i\}_{i=1}^{N'}$, where $\tilde{X^i} = \mathcal{G}\left(X^i\right)$ and $N' = fN$ as generation can be repeated f times[1]. The augmented samples are expected to maintain both affinity and diversity. D_{train} together with D_{syn} are used to improve the classifier's robustness and performance (Gontijo-Lopes et al., 2020).

2.2 Text-To-Text model selection

We adopt T5 and BART model(base version) for text-to-text generation, for sake of their relatively lower computational costs and being used as benchmarks in previous studies. Note that, however, they can be easily replaced by any other text-to-text LMs, such as MASS (Song et al., 2019).

2.3 On-demand fine-tuning

To adapt to a downstream task, the most common approach is fine-tuning, in which PLM's weights are slightly updated based on a specific dataset $\mathcal{D}_{task}$. For text-to-text models such as T5/BART, $\mathcal{D}_{task} = \{T_x^i, T_y^i\}_{i=1}^{L}$ consists of parallel text pairs. Fine-tuning requires extra update steps and large L to optimize the parameters θ_e(encoder) and θ_d(decoder) with the objective of minimizing the loss of expression 1.

$$\mathcal{L}_{\text{Pair}} = \sum_{(T_x,T_y)\in\mathcal{D}_{\text{task}}} -\log p\left(T_y \mid T_x; \theta_e, \theta_d\right) \quad (1)$$

In this work, we present two new fine-tuning schemes tailored for text-to-text DA.

1. **Paragraph To Paragraph (P2P)** We observe that sentences in the same article tend to demonstrate strong internal consistency and coherence, so sentences in the front can serve as a prologue that summarizes or induces the remaining part. Motivated by this observation, we present a scheme which cuts an article of M sentences in half: taking the first $M/2$ as the source text T_x and the remaining as the target T_y. To gather enough knowledge of the context, any articles with $M < 4$ will be pruned out.
2. **Shard To Shard (S2S)** We also notice that adjacent sentences/paragraphs in the same article tend to deliver similar meanings and therefore are semantically related. To reflect this idea, we next present another scheme which first shuffles the sentences, and then randomly sample $M/2$ sentences as T_x and the remainder as T_y. It is expected that this scheme could replenish related contents based on the fragments of the original text.

We should note that we do not cherry pick samples and remove noises or redundancy to minimize the human intervention.

2.4 New sample generation

We take each X^i from D_{train} as prompt and pass it into T5/BART for generation of the augmentation sample [2].

3 Experiments

3.1 Datasets

Related free corpus for fine-tuning We proposed to fine-tune models on some open corpus freely available. Given the domain similarity and transferability, we use the *realnewslike* split of **C4**(Raffel et al., 2019) which is extracted from news websites, to fine-tune model for downstream topic classification task. For sentiment classification task, we employ the union of ***Amazon Review***, ***Yelp Restaurant Review*** (Zhang et al., 2015) and ***IMDB Movie Review*** (Maas et al., 2011).

Experimental dataset for DA. To justify the effectiveness of our proposal, we carefully design a series of experiments and evaluate it on topic classification datasets: **AG News** (Zhang et al., 2015) and sentiment classification **SST-2**(Stanford Sentiment Treebank) (Socher et al., 2013). Both datasets are class balanced. Details can be seen in Table 3 in Appendix.

3.2 Baseline DA methods

To make a comprehensive comparison, we include most popular baseline methods: 1) **EDA** (Wei and Zou, 2019) and **CBERT**(Wu et al., 2019) both of which are based on token manipulation, 2) **Back-Translation(BT)** (Sennrich et al., 2016) based on paraphrasing, and 3) **LAMBDA** (Anaby-Tavor et al., 2020) based on generation. Among them,

[1] We keep f to 1 throughout this work for simplicity

[2] Occasionally the sample may need to be truncated to meet models' input limitation requirement, however, this seldom happens in AG and SST-2 dataset

CBERT and **LAMBDA** are both label-conditional and need to be fine-tuned on D_{train} following their own schemes.

3.3 Backbone classifiers

To evaluate the gain of introducing new samples, two widely adopted classifiers are employed: one is a light-weighted transformer and the other is the bulky and resource-hungry BERT (Devlin et al., 2019). In each trial, with a random seed, we select K samples from each class to construct a balanced dataset D_{train} and apply different DA methods to derive synthetic datasets D_{syn} respectively. Next, a classifier $\mathcal{C}'$ is trained on $D_{train} \cup D_{syn}$ and $\mathcal{C}$ is trained only on D_{train}. Finally, both $\mathcal{C}'$ and $\mathcal{C}$ are evaluated on D_{test}. This trial is repeated 100 times with different random seed to report the averaged accuracy overall to get a reliable finding.

3.4 Main results and Analysis

Comparison with baselines. Our proposed method is compared with alternatives introduced in Section 3.2. The average accuracy is reported in Table 1 [3].

It is clear that our method demonstrates the superiority over all the benchmarks, especially in low-data regime. In DA for AG News topic classification task, fine-tuning T5 or BART on C4 consistently outmatches the baselines, while T5 fine-tuned on S2S paradigm yields the best results. In DA for SST-2 sentiment classification task, fine-tuning BART on reviews corpus under S2S scheme also shows obvious gains. When the training corpus is larger , the gain from DA becomes marginal. Note that LAMBDA is also a GLM-based DA method; however its performance is not up to par. Similar observations have been reported in some recent studies (Wang et al., 2022; Sahu et al., 2022), which suggests directly fine-tuning PLMs with small training data may lead to overfitting as they simply attempt to memorize what they see.

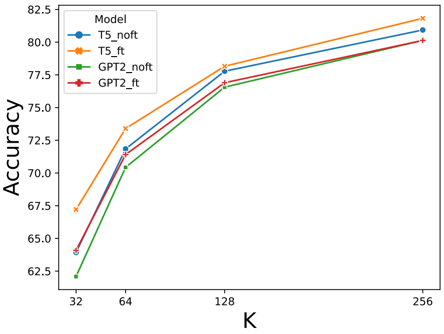

Figure 1: Comparison among T5 fine-tuned under **S2S** scheme(**T5_ft**) and T5 off-the-shelf version(**T5_noft**); GPT2 off-the-shelf(**GPT2_noft**), GPT2 fine-tuned on *C4-realnewslike*(**GPT2_ft**) in AG topics classification task, transformer-based classifier

Ablation study. In this part, we demonstrate the necessity of appropriate fine-tuning scheme. Also, as GPT2 is widely used in previous generation based DA and it also shares some commonness in terms of the transformer architecture, here we aim to compare between Text2Text model and GPT2 with and without fine-tuning, where T5 is used as representative of the Text2Text LM. Besides, since GPT2 is pre-trained on corpus of all the web pages scraped from outbound links on Reddit, which has domain discrepancy from News, therefore, we fine-tune GPT2 to C4-realnewslike to eliminate this potential gap.

As shown in Figure 1, 1) Removing fine-tuning always reduce the performance under various K, for both T5 and GPT2, which justifies the necessity of domain adaption and appropriate fine-tuning scheme design; 2) The auto-regressive GPT2 underperforms T5, which indicates that the structure of seq2seq is more suitable for generation-based DA. We will analyze this observation later.

Limitations. Same with the previous findings (Longpre et al., 2020), when the backbone classifier is PLM-based, as shown in Table 2, the gains are not significant or even become adverse. It is more clear in the sentiment classification, where various DA methods fail to ameliorate to a large extent; sometimes even hurt the performance when K is large. Also, our proposed methods do not gap too much away from baselines. For the topic classification, we can still witness an unignorable boost from BART fine-tuned under S2S scheme, when $K \leq 64$; however, it still suffers diminishing utility when $K \geq 128$.

Discussion. In line with findings from table 1 and 2, fine-tuning BART under S2S scheme can be a good practice when employing DA in sentiment classification. There are a variety of noising transformations, such as text infilling and sentence shuffling, in the pre-training stage of BART. Therefore, BART's ability of denoising corrupted documents in pre-training is more closely related to our S2S scheme in review corpus which presents more chal-

[3] $std \leq 0.1$, to save space we do not list them in the table

Table 1: Comparison with baselines under **transformer**-based classifier and various K settings, $|D_{train}| = |D_{syn}|$

Methods			AG				SST-2			
			K=32	K=64	K=128	K=256	K=32	K=64	K=128	K=256
	No DA		58.89	68.00	75.05	79.85	55.89	59.53	63.37	66.79
Baselines	EDA		59.59	68.61	74.88	80.55	55.89	59.33	63.11	66.62
	BT		59.96	69.21	74.67	79.90	56.19	59.88	63.15	66.64
	CBERT		59.81	69.97	75.89	80.03	56.98	59.98	63.45	66.97
	LAMBDA		60.02	69.34	75.37	80.46	56.77	60.02	63.29	66.18
Ours.	T5	S2S	**67.21**	**73.40**	**78.16**	**81.82**	57.28	60.71	63.92	67.03
		P2P	65.65	72.83	77.96	81.34	57.16	60.39	63.67	66.95
	BART	S2S	65.16	72.34	77.00	80.77	**58.21**	**61.43**	**64.86**	**67.30**
		P2P	64.99	71.77	76.51	80.91	57.72	61.37	64.17	66.96

Table 2: Comparison with baselines under **BERT**-based classifier and various K settings, $|D_{train}| = |D_{syn}|$.

Methods			AG				SST-2			
			K=32	K=64	K=128	K=256	K=32	K=64	K=128	K=256
	No DA		84.22	86.82	87.54	88.03	70.10	78.30	**84.93**	**86.90**
Baselines	EDA		85.13	86.45	87.70	88.15	72.19	79.15	84.65	85.46
	BT		85.12	86.60	87.18	88.16	76.94	81.04	84.27	85.32
	CBERT		85.28	86.79	87.37	88.01	74.09	80.07	84.38	85.88
	LAMBDA		85.07	86.55	87.21	87.98	75.08	80.32	84.55	85.98
Ours.	T5	S2S	84.83	86.39	87.28	87.96	70.87	79.15	84.08	85.76
		P2P	84.76	86.52	87.29	87.99	70.83	78.79	83.95	83.84
	BART	S2S	**85.35**	**86.84**	87.62	**88.26**	**79.81**	**82.25**	84.59	85.99
		P2P	85.17	86.81	**87.71**	88.07	78.42	82.14	84.78	86.32

lenges and makes BART more powerful.

For topic classification, employing T5 is a relatively better choice. As T5's pre-training task is fill-in-the-blank-style denoising objectives (span corruption and recovery), T5 primarily focuses on filling in dropped-out spans of text, which forces T5 to answer cloze questions based on "knowledge". This is more conducive to topic classification DA where bringing in more related entities(acquiring rich knowledge) is more crucial than adjusting sentence order or guaranteeing coherency.

GPT2 is widely used in previous generation-based DA, it is true that during inference, GPT2 is rambling on its own previous output, making generation prone to be off-topic that can lose fidelity in DA. In addition, GPT2 is a pure decoder model, while T5/BART consists of encoder and decoder. In other words, unlike the auto-regressive generation, T5 belongs to *directed generation*. Theoretically, T5/BART brings more advantages because of encoder-decoder attention layer which helps the generative decoder focus on appropriate places in the source text. This is the main reason why we introduce T5/BART into DA and its effectiveness is justified.

4 Conclusion and Future Work

In this paper, we propose to use text-to-text LMs as a new paradigm for data augmentation in text classification. Compared to other methods along this direction, our approach is robust, easy-to-implement and does not need laborious human intervention. In future work, it is worth exploring more tailored fine-tuning scheme for DA tasks under various scenarios.

5 Acknowledgments

This work was supported by the NSERC Discovery Grants.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Not enough data? deep learning to the rescue! In *AAAI*, pages 7383–7390.

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 NAACL: HLT*, pages 4171–4186.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. 2020. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 NAACL: HLT, Volume 2 (Short Papers)*, pages 452–457.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *CoRR*, abs/2003.02245.

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

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation. In *EMNLP*, pages 9031–9041.

Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the ACL: EMNLP*, pages 4401–4411.

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 ACL: HLT*, pages 142–150, Portland, Oregon, USA. ACL.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Sebastian Ruder. 2021. Recent Advances in Language Model Fine-tuning. `http://ruder.io/recent-advances-lm-fine-tuning`.

Gaurav Sahu, Pau Rodriguez, Issam H Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. *arXiv preprint arXiv:2204.01959*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL (Volume 1: Long Papers)*, pages 86–96.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th ICML*, volume 97 of *PMLR*, pages 5926–5936. PMLR.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. Promda: Prompt-based data augmentation for low-resource nlu tasks. *arXiv preprint arXiv:2202.12499*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP-IJCNLP*, pages 6383–6389.

Lilian Weng. 2022. Learning with not enough data part 3: Data generation. *lilianweng.github.io*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *ICCS*, pages 84–95. Springer.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *CoRR*, abs/2104.08826.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS-Volume 1*, pages 649–657.

A Implementation details

All experiments are conducted on Linux platform with GPU instance of Nvidia Tesla V100 type.

For the fine-tuning stage, we adopt the script from huggingface-transformers [4]. All the datasets we use are download from Huggingface-datasets[5]. We set the maximum length of both the source and target text to 256 which break the balance between performance and efficiency. Batch size is set to 16 and learning rate is $1e^{-5}$. Other parameters follow the default setting. For the fine-tuning corpus, we randomly split out 5% for the review dataset as validation set while for C4 corpus, the validation set is already officially split. We monitor the rouge score(Lin, 2004) at each epoch and pick the model of the best performance [6].

For the DA experiments stage, following previous studies, we set the optimizer as Adam (Kingma and Ba, 2014) with an initial learning rate of $4e^{-5}$ for training the classifier. The light-weighted Transformer-based classifier is referred to Keras implementation[7]. Pre-trained BERT is downloaded from Tensorflow Hub[8]. In each trial we run the training for 100 epochs and record the best accuracy on test set which is officially provided. We keep the classifier training settings exactly the same for all trials with and without DA, to ensure fairness. Therefore, the only difference exists in the introduction of D_{syn} produced from various approaches including ours or baselines.

In the generation process of T5 and BART, we set maximum length limit: 128 for AG and 64 for SST-2 DA scenario, based on the average length of samples in D_{train}. Therefore, generation is terminated when the special EOS token is ejected or the length of the generated text reach this limit. Nucleus sampling is used in generation ($P = 0.9$), to avoid sampling egregiously wrong tokens, but preserve variety when the highest scoring tokens have low confidence. Temperature and repetition penalty is set to 1.2. We only apply basic post-processing to filter generated examples that are too short or full of punctuation or repetitions which rarely happen in practice.

Among the baseline methods, we follow the optimal settings from the original papers. We set the intermediary language to Chinese for **BT**.

Our source code is released in Github repository[9].

Table 3: Descriptions of topic and sentiment categorization datasets.

Data	Labels	Domain
AG	World, Sports, Business, Tech	topic
SST-2	Positive, Negative	sentiment

[4] https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization

[5] https://huggingface.co/datasets

[6] We find that 1 or 2 epoch is always sufficient to convergence as the rouge metrics on validation set does not grow anymore.

[7] https://keras.io/examples/nlp/text_classification_with_transformer

[8] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

[9] https://github.com/yananchen1989/PLMs_text_classification