CONT: Contrastive Neural Text Generation

Aditya Jain
aj780

Abhinay Reddy Vongur
av730

Parth Hasmukh Jain
pj269

Abstract

Contrastive learning has gained significant attention in the field of text generation due to its ability to alleviate exposure bias. However, previous approaches applying contrastive learning to text generation have not provided significant improvements in performance. Authors of the paper CoNT:Contrastive Neural Text Generation framework addresses this issue by introducing strategies in three key aspects of Contrastive Learning: selecting in-batch contrastive examples, using a contrastive loss, and inference with a learned similarity function. We evaluate CoNT on two tasks - common sense generation on the Common-Gen dataset and text summarization on the X-Sum dataset - and try to replicate some of the results achieved by the authors of the original paper. Our experiments demonstrate that CoNT is a promising framework for improving text generation performance.Our code is available in this github repository [1]

1 Introduction

Typically, a neural sequence-to-sequence model consists of an encoder and a decoder, which are trained using MLE loss. During training, the model receives the previous ground truth $y < t$ as input and is not exposed to incorrect labels. However, during inference, the model receives the tokens $y_{<t}$ that it previously predicted, which can lead to exposure bias. Exposure bias refers to the mismatch between the training and inference conditions of a sequence-to-sequence model, where the model is trained on ground truth data but during inference it is fed with its own previously generated tokens as input. The main problem with this is that mistakes made early in the sequence generation process are fed as input to the model and can be quickly amplified because the model might be in a part of the state space it has never seen at training time. Given a source sequence $x = \{x_i\}_M^{i=0}$ and its target sequence $y = \{y_i\}_{i=0}^N$, the following negative log-likelihood (NLL) loss is minimized:

$$\mathcal{L}_{NLL} = -\sum_{t=1}^{N} \log p_\theta(y_t|x, y_{<t}) \qquad (1)$$

Addressing the exposure bias problem is crucial for achieving high-quality text generation performance. The discrepancy between the training and inference conditions can result in the model being unable to handle the errors that accumulate over time and lead to compounding errors. This issue can significantly degrade the quality of generated text. Therefore, it is imperative to mitigate the exposure bias problem to improve the performance and quality of sequence-to-sequence models for text generation tasks.

Contrastive learning has emerged as a promising technique for mitigating the exposure bias problem in text generation tasks. By leveraging contrastive objectives, the model can learn to identify the differences between the true and predicted distributions and minimize the discrepancy between them. This can help to reduce the compounding errors that can arise from the exposure bias problem, leading to improved text generation quality. Contrastive learning achieves this by training the model to discriminate between positive (correct) and negative (incorrect) examples, thereby encouraging the model to generate more accurate and diverse text.

Naive Contrastive Learning approaches use from-batch negative samples for training to help the model differentiate better between correct and incorrect labels. It also introduces a contrastive term in addition to the original NLL loss. Naive approaches generally use InfoNCE (Wu et al., 2021) loss which is expressed as

[1] Code: https://github.com/abhinayrv/Contrastive-Text-Generation

$$\mathcal{L}_{NCE} = -\log \frac{exp(cos(z_x, z_y)/\tau)}{\sum_{y' \in \mathcal{B}} exp(cos(z_x, z_y)/\tau)} \quad (2)$$

where z_x, z_y, $z_y' \in R^d$ denote the vector representation of input x, ground truth y and negative sample $y' \in \mathcal{B}$, respectively. τ is the temperature and $cos(\cdot, \cdot)$ defines the cosine similarity. Intuitively this loss tries to learn a similarity function that rates the source sequence z_x and the target z_y to be closer.

The technique employed by Naive approaches to select negative examples is not effective as the from-batch samples are often easily differentiable from the ground truth. With pretrained models trained on large text this task of differentiating becomes even simpler. Therefore these examples do not add much value in training the model. This problem can be addressed by selecting negative examples that are closer to the ground truth, which are called hard negatives. Some of the previous works applying contrastive learning to text generation (Lee et al., 2020), (Ng et al., 2020) create negative examples by perturbing the ground truth itself. However these approaches failed to provide performance improvements over the base models.

Naive approaches also fail to fully leverage the similarity to the ground truth and the distance between the source and the ground truth. The InfoNCE loss treats all the negative examples the same and does not differentiate whether they are closer to the ground truth or far from it.

In this paper, authors propose a contrastive learning framework that addresses the problems in applying contrastive learning to text generation from three aspects. First, creation of hard negatives. In this approach hard negatives are generated using the predictions from the model itself.

As a second step, our proposed approach utilizes an N-pairs contrastive loss that provides fine-grained treatment to the contrastive examples based on their similarity to the ground truth. This also enables the model to learn a distance function between the source and the ground truth.

Finally in previous contrastive learning works, the decoding function remains the same as without contrastive learning. This work introduces the learned distance function from the n-pairs contrastive loss in the inference stage in addition to the likelihood score to provide better predictions.

One of the limitations of this framework is the time taken to train the model. The training process using this approach can be slow as it involves pre-training the model to provide better quality examples, generation of examples while training and calculation of similarity between the negative and positive examples.

2 Related work

Naive Contrastive learning is primarily based on the approach outlined by (Jiang et al., 2020) in SimCLR for selecting the negative examples and the InfoNCE loss proposed by (Parulekar et al., 2023). There have been numerous applications of Contrastive Learning in NLP for representation learning like (Gao et al., 2021), (Krishna et al., 2022). Previous works in text generation employed adversarial techniques of disturbing the ground truth (Gao et al., 2021),(Lee et al., 2020), (Ng et al., 2020) to create additional negative examples while training.

SimCTG (Reiter et al., 2016) is another framework for applying contrastive learning to text generation but is more focused on encouraging diversity in dialogue systems than improving the robustness of the model. (Bengio et al., 2015) propose a different framework to combat the problem of exposure bias by using the generated tokens instead of the true previous tokens in training the sequence to sequence models.The binary supervision loss that helps makes the positive example closer to the source and maximize the distance between the negative example and the source was first proposed by (Schroff et al., 2015) in FaceNet. There have also been other applications (Chen and Deng, 2019) of pair-wise contrastive loss

We propose a curriculum learning strategy to gently change the training process from a fully guided scheme using the true previous token, towards a less guided scheme which mostly uses the generated token instead.

3 Method

Our objective for this project is to explore and implement the proposed techniques in (An et al., 2022), which introduce a novel contrastive neural text generation framework that deviates from Naive Contrastive Learning in three key ways.An overview of our approach can be found in the figure 1

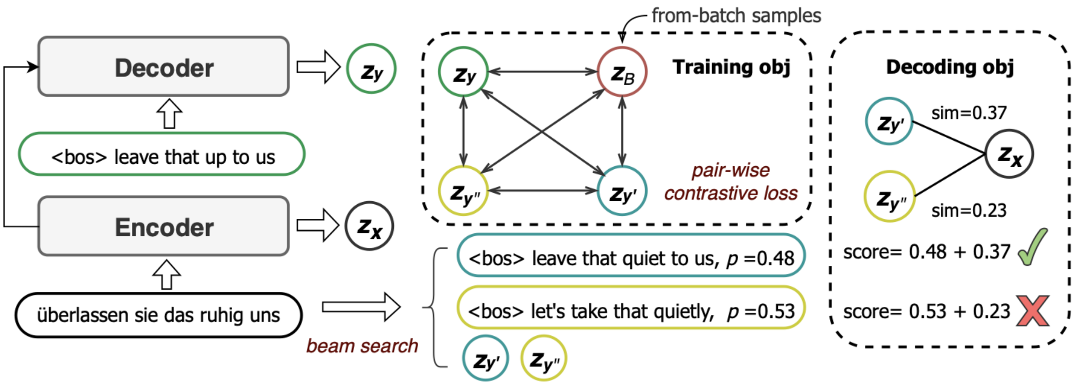

Figure 1: Model Architecture

3.1 Contrastive Examples

Firstly,Instead of only using contrastive examples from the same batch[6]], the CoNT model samples negative examples for contrastive learning from its own predictions. To generate contrastive examples, the diverse beam search algorithm is used to select the top-K predictions from the model's outputs. These predictions which are considered hard naegatives are then appended to the from-batch samples to form the set of contrastive examples. A warm-up stage where the model is solely supervised by MLE loss can ensure high-quality contrastive examples.

3.2 N-Pairs Contrastive Loss

Secondly, instead of treating all negative examples equally, CoNT utilizes an N-pairs contrastive loss that treats each example differently based on their sequence level scores. The contrastive examples are ranked based on their similarity to the ground truth, and a pair-wise margin loss is employed to deal with the problem of treating all negative examples the same. The contrastive loss for each pair of positive and negative examples is computed using their cosine similarity to the source and difference in their rank. The contrastive learning objective is formulated as a margin loss. The equation for N-pairs loss is as follows:

$$\mathcal{L}_{N-Pairs} = \sum_{(y^+,y^-)\in\mathcal{P}} \mathcal{L}(y^+,y^-)$$

$$= \sum_{(y^+,y^-)\in\mathcal{P}} max\{0, cos(z_x, z_{y^-}) - cos(z_x, z_{y^+}) + \epsilon\} \tag{3}$$

3.3 Inference with Learned Similarity Function

Thirdly, CoNT uses the learned sequence similarity score from the distance function in the inference stage. This is in contrast to previous works that used the same inference method as non-contrastive generation approaches. In CoNT, the objective then in the decoding stage is to find the sequence y^* that maximizes both the learned similarity score and the conventional language model likelihood. The equation for y^* is:

$$\begin{aligned} y^* &= argmax_{\hat{y}}\{\alpha \cdot cos(z_x, z_{\hat{y}}) \\ &+ (1-\alpha)\prod_{t=0}^{n} p(\hat{y}_t|x, \hat{y}_{<t})\} \end{aligned} \tag{4}$$

where $z_x, z_{\hat{y}} \in R^d$ is the vector representation of x, $\hat{y}$, and α is the hyperparameter that balances the contribution of each term.

To achieve our objectives, we will train the t5-small model on the xsum dataset and the t5-base model on the CommonGen dataset using the contrastive learning framework outlined above.

4 Experiments

Common Sense generation is a constrained text generation task to explicitly test machines for the ability of generative commonsense reasoning. Given a set of common concepts (e.g., dog, frisbee, catch, throw); the task is to generate a coherent sentence describing an everyday scenario using these concepts (e.g., "a man throws a frisbee and his dog catches it"). The CommonGen (Lin et al., 2019) task is challenging because it inherently requires 1) relational reasoning with background commonsense knowledge, and 2) compositional generaliza-

Model	BLEU-3/4		ROUGE -L	METEOR	CIDEr
T5-base	28.76	18.54	34.56	23.94	9.4
T5-large	43.01	31.96	42.75	31.12	15.13
T5-base-CONT	42.6	31.42	43.15	32.05	15.96
T5-base-CONT (Reproduced)	29.3	20.6	49.8	28.9	12.67

Table 1: Results on Common Sense Generation task

Model	ROUGE -1	ROUGE -2	ROUGE -L
T5-small	36.10	14.72	29.16
T5-Naive CL	36.34	14.81	29.41
T5-small-CONT	39.66	16.96	31.86
T5-small-CONT (Reproduced)	30.48	8.36	21.11

Table 2: Results on summarization task with X-sum dataset

tion ability to work on unseen concept combinations. We evaluate and try to reproduce the results on the CommonGen benchmark. The benchmark consists of a hidden test set as well, but we evaluate just on the dev set. In addition to popular metrics like ROUGE and BLEU, CIDEr and METEOR, evaluating semantic faithfulness are also calculated. The paper's author results show that the t5-base model is able to greatly benefit from the contrastive learning framework. From table 1, we were able to get approximately the same results on ROUGE, METEOR and CIDEr metrics but our BLUE scores are not up to par with the original scores.

For abstractive text summarization,Extreme Summarisation (XSum) dataset (Narayan et al., 2018) is used. The dataset consists of 226,711 news articles accompanied with a one-sentence summary. The articles are collected from BBC articles (2010 to 2017) and cover a wide variety of domains (e.g., News, Politics, Sports, Weather, Business, Technology, Science, Health, Family, Education, Entertainment and Arts). The official random split contains 204,045 (90%), 11,332 (5%) and 11,334 (5%) documents in training, validation and test sets, respectively. Experimental results in 2 indicate the performance (ROUGE scores) of t5-small model on the Xsum dataset. Again, the author's results indicate that the model greatly benefits from the contrastive learning framework. We were able to reproduce the implementation,but we can see our results are not there yet.

For both these tasks, we use the best hyperparameters described in the paper. But, due to limitation of the computational resources, we had to significantly reduce the batch size while training. The framework also requires a warmup (or pre training) stage, just on the MLE loss. Due to training time reaching into days, we skipped this step and directly used the fine tuned models provided at hugging-face. This in fact is also a limitation of our training procedure.

5 Conclusions

The paper introduces a novel contrastive learning framework for text generation. The method fundamentally changes the way negative samples are sourced, adds an additional term called n-pairs loss to the original MLE loss and modifies the decoding stage to get the sequence which maximizes the MLE loss and learned similarity score. The framework is evaluated on two different text generation tasks. The results depict that CONT not only clearly beats all previous contrastive generation models, but also boosts the performance of state-of-the-art large models to a new level. Nevertheless, CONT still suffers from training inefficiency. In general, the total training time of CONT is about 2$\tilde{4}$ times more than that of a MLE based model. Next steps would be to reduce the training time of the framework.

References

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont:

Contrastive neural text generation. *arXiv preprint arXiv:2205.14690*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Binghui Chen and Weihong Deng. 2019. Energy confused adversarial metric learning for zero-shot image retrieval and clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8134–8141.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Dongwei Jiang, Wubo Li, Miao Cao, Wei Zou, and Xiangang Li. 2020. Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning. *arXiv preprint arXiv:2010.13991*.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. *arXiv preprint arXiv:2205.09726*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2020. Contrastive learning with adversarial perturbations for conditional text generation. *arXiv preprint arXiv:2012.07280*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*.

Advait Parulekar, Liam Collins, Karthikeyan Shanmugam, Aryan Mokhtari, and Sanjay Shakkottai. 2023. Infonce loss provably learns cluster-preserving representations. *arXiv preprint arXiv:2302.07920*.

Michael Reiter, Marco Erler, Christoph Kuhn, Christian Gusenbauer, and Johann Kastner. 2016. Simct: a simulation tool for x-ray imaging. In *Proceedings of the 6th Conference on Industrial Computed Tomography*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. Rethinking infonce: How many negative samples do you need? *arXiv preprint arXiv:2105.13003*.