Abstractive Text Summarization Using the BRIO Training Paradigm

Khang Nhut Lam
Can Tho University, Vietnam
lnkhang@ctu.edu.vn

Thieu Gia Doan
Can Tho University, Vietnam
dgthieu@cusc.ctu.edu.vn

Khang Thua Pham
Duy Tan University, Vietnam
phamthuakhang@dtu.edu.vn

Jugal Kalita
University of Colorado, USA
jkalita@uccs.edu

Abstract

Summary sentences produced by abstractive summarization models may be coherent and comprehensive, but they lack control and rely heavily on reference summaries. The BRIO training paradigm assumes a non-deterministic distribution to reduce the model's dependence on reference summaries, and improve model performance during inference. This paper presents a straightforward but effective technique to improve abstractive summaries by fine-tuning pre-trained language models, and training them with the BRIO paradigm. We build a text summarization dataset for Vietnamese, called VieSum. We perform experiments with abstractive summarization models trained with the BRIO paradigm on the CNNDM and the VieSum datasets. The results show that the models, trained on basic hardware, outperform all existing abstractive summarization models, especially for Vietnamese.

1 Introduction

Text summarization reduces the size of the original text while preserving its main content. The two main approaches for constructing summaries are extractive and abstractive. Extractive summarization directly lifts sentences or words which convey key topics of the original documents, and concatenates them. Abstractive summarization discovers the primary content of the documents and generates summaries. Abstractive summaries are usually more natural and coherent than extractive summaries.

Most abstractive summarization models follow the encoder-decoder framework. Existing abstractive summarization models are trained using maximum likelihood estimation and rely on the reference summaries. Liu et al. (2022a) propose a BRIO training paradigm to address reliance on reference summaries by assuming non-deterministic distribution of system-generated candidate summaries. In this paper, we use the BRIO training paradigm for abstractive summarization models to construct summaries for documents in English and Vietnamese. We make the following contributions:

- We adapt the BRIO training paradigm for abstractive summarization using BART-based and T5-based models as backbones.
- We present issues with the BRIO paradigm.
- We investigate abstractive summarization models using BARTpho-BRIO and ViT5-BRIO to obtain improved results.
- We publicly release the VieSum summarization dataset for research purpose.

The remainder of this paper is organized as follows. Related work is presented in Section 2. Section 3 introduces a large dataset for summarization in Vietnamese, named VieSum. Experiments and discussion are presented in Section 4. Section 5 concludes the paper.

2 Related Work

Sheng et al. (2022)'s Siamese Semantic Preserving Generative Adversarial Net (SSPGAN) uses a Transformer-based generator to generate summaries. A Siamese Transformer-based discriminator captures the semantic consistency between the source document and the corresponding summary. During adversarial training, the discriminator calculates a reward for each word generated. On the Gigaword dataset, SSPGAN model achieves better results than many existing abstractive text summarization models such as deep recurrent generative decoder (Li et al., 2017), actor-critic approaches from reinforcement learning (Li et al., 2018), and Transformer (Vaswani et al., 2017).

Liu et al. (2022b) develop the PageSum model for abstractive summarization by incorporating locality bias in both encoder and decoder. Each document is partitioned into non-overlapping pages.

Findings of the Association for Computational Linguistics: ACL 2023, pages 92–99
July 9-14, 2023 ©2023 Association for Computational Linguistics

The encoder, which is an abstractive summarizer, encodes each page and makes local predictions. The decoder predicts output based on a weighted combination of local predictions. The authors fine-tune the BART model (Lewis et al., 2020) for abstractive summarization and investigate several approaches to locality, such as spatial locality, discourse locality, and document locality. PageSum outperforms abstractive summarization models such as longformer encoder-decoder (Beltagy et al., 2020), encoder-decoder attention with head-wise positional strides (Huang et al., 2021), and BART with Hierarchical Attention Transformer (Rohde et al., 2021). However, PageSum takes a long time to train, requires large memory size, and fails to capture long distance dependencies.

Several studies use pre-trained models for abstractive text summarization. Farahani et al. (2021) use mT5 (Xue et al., 2021) and sequence to sequence ParsBERT (Rothe et al., 2020) to construct abstractive summaries for Persian texts. T5 (Raffel et al., 2020) and BERT (Devlin et al., 2018) have also been used to construct abstractive summaries (Garg et al., 2021). Kieuvongngam et al. (2020) summarize COVID-19 biomedical research articles using BERT and GPT-2 (Radford et al., 2019). Features of documents are extracted and integrated into an abstractive model to improve summary generation. Nambiar et al. (2022) develop an encoder-decoder model using attention, in which POS features are incorporated to the word embedding layers to enhance the word vectors. Experiments on a dataset in Malayalam show that the integration of attention model and POS features is better than the seq2seq and attention models. Barna and Heickal (2021) adapt the pointer generator network for abstractive summarization by combining a pre-trained word embedding layer for transferring semantic similarity and topic features for better topic coverage. A drawback of usual abstractive summarization is the omission of named entities. To ameliorate, Berezin and Batura (2022) train a named entity recognition model based on ROBERTa to discover named entities. Then, the BART masked named entity language model is trained to pay attention on the name entities. Finally, BART is fine-tuned for text summarization.

Most studies to construct abstractive summaries in Vietnamese use an encoder-decoder framework or a pre-trained model. Quoc et al. (2019) integrate sentence positions and term frequencies into a pointer generator network with a coverage mechanism to perform the abstractive summarization for Vietnamese documents. Lam et al. (2022) construct abstractive summaries for online newspapers using RNN with attention, BiLSTM with copy generator, standard Transformer, BERT, and sequence-to-sequence abstractive models using bottom-up approach. Phan et al. (2022) perform experiments to summarize Vietnamese documents using Transformer-based encoder-decoder architectures such as Transformer, PhoBERT (Tran et al., 2022), and ViT5 (Phan et al., 2022).

3 VieSum Dataset

We construct a VieSum dataset for Vietnamese consisting of 1,627,415 documents and their corresponding summaries, grouped into 23 categories. In particular, BeautifulSoup[1] and Newspaper3k[2] are used to collect and extract articles from popular online newspapers in Vietnamese such as vnexpress.net, dantri.com.vn, danviet.vn, vietnamnet.vn, laodong.vn, and vov.vn. The summaries and content documents are considered reference summaries and documents, respectively.

4 Experimental Results

We perform experiments in the Google Colaboratory environment, NVIDIA Tesla T4 16GB. We use the CNNDM[3] dataset in English, and our VieSum dataset in Vietnamese. Due to limitation of the hardware, we perform experiments with 70,000 documents picked randomly and their corresponding reference summaries from VieSum. Each dataset is split into 3 parts including 75% for training, 8% for validation, and 17% for testing.

In this paper, the pre-trained $\text{BART}_{\text{512-length}}$-based and $\text{T5}_{\text{512-length}}$-based models are used as backbones for generating abstractive summaries. The BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) models are trained on the CNNDM dataset, while the BARTpho (Tran et al., 2022) and ViT5 (Phan et al., 2022) are trained on the VieSum dataset. All models are base models. To make it easy for comparison, we use the same parameters as suggested by the original authors.

[1] https://www.crummy.com/software/BeautifulSoup/

[2] https://newspaper.readthedocs.io/en/latest/

[3] https://cs.nyu.edu/ kcho/DMQA/

Dataset	System	R-1	R-2	R-L
CNNDM	BART	42.53	20.21	39.47
CNNDM	T5	36.24	15.34	33.34
VieSum	BARTpho	44.59	22.57	34.60
VieSum	ViT5	53.39	20.63	35.88

Table 1: ROUGE scores of abstractive summarization systems using standard backbone models.

System	R-1	R-2	R-L
T5 fine-tuned	41.02	19.44	38.30
BARTpho fine-tuned	57.94	26.56	40.83
ViT5 fine-tuned	57.75	26.37	40.57

Table 2: ROUGE scores of abstractive summarization systems using the fine-tuned backbone models. The T5 fine-tuned model is trained on CNNDM, while the other models are trained on VieSum.

4.1 Standard Abstractive Models

First, we experiment and evaluate abstractive summarization approaches using standard BART-base and T5-base models. We train the models using a batch size of 4, epoch count of 5, learning rate of 10^{-5}, warmup_step of 20,000, and the Adam optimizer. The results of abstractive summarization systems using the standard backbone models are presented in Table 1.

4.2 Fine-tuning Abstractive Models

To improve the quality of summaries created, we fine-tune the backbone models using the Trainer provided by Hugging Face[4]. We do not fine-tune the BART model because it is already fine-tuned on the CNN dataset. Table 2 shows the ROUGE scores of the fine-tuned abstractive models.

4.3 Fine-tuning Abstractive Models and BRIO

The BRIO (Liu et al., 2022a) training paradigm helps abstractive summarization models to predict tokens more accurately. Liu et al. (2022a) use BART as the backbone model. BRIO assigns probability mass to output summary candidates based on their quality using contrastive learning. The abstractive model acts as a generation model to generate abstractive candidates in an auto-regressive way, and an evaluation model to evaluate the candidates by calculating their probability distribution. The generator is trained using the standard MLE loss, while the evaluator is trained using a contrastive loss (Hadsell et al., 2006).

System	R-1	R-2	R-L
BART-BRIO	46.40	22.47	43.00
T5-BRIO	44.03	20.72	40.63
BARTpho-BRIO	59.12	27.01	42.05
ViT5-BRIO	59.50	27.33	42.76

Table 3: ROUGE scores of abstractive summarization systems, which use the fine-tuned backbone models, trained with the BRIO paradigm. BART-BRIO and T5-BRIO are trained on CNNDM, and BARTpho-BRIO and ViT5-BRIO are trained on VieSum.

In BRIO, a backbone model is used to produce N abstractive summaries, the so-called *candsum*s, for each document. Each *candsum* is assigned a quality score by obtaining the average score of its ROUGE-1, ROUGE-2, and ROUGE-L values. In particular, Liu et al. (2022a) use the BART$_{\text{1024-length}}$ model to create 16 *candsum*s for each document. Next, documents, reference summaries, and corresponding *candsum*s sorted by the descending quality scores are used to train the abstractive summarization model using the BRIO paradigm. We note that Liu et al. (2022a) use the standard models as back-bones and train them with the BRIO paradigm.

In our work, the fine-tuned backbone abstractive summarization models, presented in the previous section, are used to produce *N=6 candsum*s for each document using diverse beam search (Vijayakumar et al., 2018) with num_beam_groups=6, diversity_penalty=1.0, and num_beams=4. The abstractive summarization models are trained using a learning rate of 10^{-3}, and the Adafactor optimizer. Liu et al. (2022a) claim that BRIO training helps the models reach the best performance within one epoch on the CNNDM dataset[5]. Therefore, we use one epoch for training the fine-tuned summarization models with the BRIO paradigm. The results of the abstractive summarization systems trained with BRIO are presented in Table 3.

4.4 Fine-tuning Abstractive Models and BRIO-Loop

As suggested by Liu et al. (2022a), we perform loop processing, using the *candsum*s created by the abstractive summarization models trained with BRIO to train the models. However, after several

[4] https://github.com/huggingface/transformers

[5] https://github.com/yixinL7/BRIO/issues/13

System	R-1	R-2	R-L
BART-BRIO-Loop	46.55	22.56	43.00
T5-BRIO-Loop	45.24	21.50	41.80
BARTpho-BRIO-Loop	60.53	28.20	44.20
ViT5-BRIO-Loop	60.90	28.39	44.36

Table 4: ROUGE scores of abstractive summarization systems trained with the BRIO paradigm after looping twice. BART-BRIO and T5-BRIO are trained on CNNDM, and BARTpho-BRIO and ViT5-BRIO are trained on VieSum.

iterations of looping, the ROUGE scores seem to change very little. Especially, BARTpho and ViT5 almost reach the highest ROUGE scores with 2 iterations. Table 4 presents the ROUGE scores obtained after looping twice.

Experimental results show that the BRIO training paradigm significantly helps improve the abstractive summaries by reducing the dependence of the system on the reference summaries. However, assigning weights to both *candsum*s and reference summaries is necessary in order to decrease reliance on reference summaries. The diverse beam search helps obtain diverse *candsum*s, but could cause interference in the beam search space because the model might not follow the reference summaries. In addition, using the ROUGE metric for evaluating the abstractive summarization models trained with the BRIO paradigm seems unfair because these models could produce summaries which are independent on the reference summaries.

4.5 Discussion

It is not easy to make comparisons between models trained on different hardware and on different datasets. We make an attempt to compare our work with published papers on similar datasets.

Curently, BRIO using a standard BART$_{1024\text{-length}}$ model as backbone, which generates 16 *candsum*s, achieves SOTA results on the CNNDM dataset with a ROUGE-1 of 47.78 and a ROUGE-L of 32.58 (Liu et al., 2022a). In addition, BART$_{1024\text{-length}}$-BRIO with 2 iterations reaches ROUGE-1 and ROUGE-L of 48.01 and 44.67, respectively; these are both better than our BART$_{512\text{-length}}$-BRIO, which creates 6 *candsum*s for each document, after 2 iterations: 46.55 for ROUGE-1 and 43.00 for ROUGE-L.

Tawmo et al. (2022) fine-tune the T5 abstractive summarization model and evaluate on the CNNDM dataset. Their T5 model achieves ROUGE-1 and ROUGE-L scores of 40.79 and 34.80, respectively, which are lower than the scores of our fine-tuned T5 model, and significantly lower than scores of our best model, the T5-BRIO-Loop model: 45.24 for ROUGE-1 and 41.80 for ROUGE-L.

For Vietnamese abstractive summarization, Quoc et al. (2019) use LSTMs with the features of sentence positions and term frequencies (LSTM+SP+TF) on a Vietnamese dataset collected from Baomoi[6]. The best ROUGE-1 and ROUGE-L scores of their model are 31.89 and 29.97, respectively, which are significantly lower than the scores of our BRIO-BART model.

Both the BARTpho and ViT5 models trained with the BRIO paradigm outperform all models proposed by Lam et al. (2022) on the CTUNLPSum dataset, which is very similar to the VieSum dataset, including the sequence-to-sequence models, copy generator network, sequence-to-sequence with rewriter approach, and bottom-up approach.

Tran et al. (2022) apply several models for abstractive summarization on the VNDS (Nguyen et al., 2019) dataset. They perform experiments on 8 A100 GPUs with 40GB each. Their model is trained for 15 epochs in about 6 days. Their best model, BARTpho, achieves a ROUGE-1 of 61.14, which is slightly higher than the BARTpho-BRIO-Loop, and a ROUGE-L of 40.15, which is lower than that of the BARTpho-BRIO-Loop. In addition, the BARTpho-BRIO-Loop is trained on one epoch in about 32 hours using basic hardware.

Phan et al. (2022) introduce a pre-trained text-to-text Transformer for Vietnamese abstractive summarization, called ViT5. The authors claim the ViT5 model as the SOTA for Vietnamese abstractive summarization. Their ViT5 abstractive summarization model achieves ROUGE-1 and ROUGE-L of 61.85 and 41.70, respectively, on the VNDS dataset (Nguyen et al., 2019). We conducted experiments on VNDS and found interesting results related to the ViT5 model. The ROUGE scores of the ViT5 model trained using the common paradigm are essentially identical to the ROUGE scores provided by Phan et al. (2022). However, the scores of the ViT5 model trained using the BRIO paradigm are reduced to 59.37 and 41.6, respectively. On the VieSum dataset, the standard ViT5-base achieves an ROUGE-1 of 53.39 and ROUGE-L of 35.88; while the ViT5-BRIO-Loop has better scores: ROUGE-1 of 60.90 and ROUGE-L of

[6] https://baomoi.com/

44.36. We leave further exploration and evaluation these unstable results for future work.

5 Conclusion

We investigated abstractive summarization models trained with the BRIO paradigm. Experiments show that we can improve abstractive summarization models by fine-tuning the backbones before training them with BRIO. In particular, the summarization models trained with BRIO outperform other summarization models in Vietnamese. We also discuss issues with the BRIO paradigm for further exploration. In addition, we built the VieSum dataset for summarization in Vietnamese. For future work, we will ask volunteers to evaluate and provide feedback on a small subset of the VieSum dataset.

Limitations

While many studies show that the architectures of the deep learning models significantly influence the results, we perform experiments with several base architectures because of the constrained hardware. Furthermore, there has not been a Vietnamese benchmark summarization dataset, which is both sizable and of high quality. The existing summarization datasets are derived from online magazines, which usually contain misspelled words and grammatical errors. In addition, the reference summaries might not convey the main content of the corresponding articles. Therefore, selecting and developing efficient summarization models for Vietnamese still present numerous challenges.

Ethics Statement

We use several different software tools in our experiments. These tools as well the English dataset are publicly available and we do not see any ethical issues in using them. In addition, we clearly reference the papers and other sources for the tools used. We create the VieSum dataset ourselves.

Our paper's work depends on using previously published approaches to abstractive summarization. We clearly give credit to the authors of these approaches by citing original sources.

This paper focuses on abstractive summarization of longer documents. There is potential for high quality abstractive summarizers to be misused. For example, students if/when given an assignment to summarize/review papers/articles may use such summarizers to automatically write reviews and claim them as their own. However, we believe abstractive summarizers for long documents have not achieved this level of sophistication at this time.

References

Nasid Habib Barna and Hasnain Heickal. 2021. An automatic abstractive text summarization system. *Dhaka University Journal of Applied Science and Engineering*, 6(2):39–48.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Sergey Berezin and Tatiana Batura. 2022. Named entity inclusion in abstractive text summarization. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 158–162.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mehrdad Farahani, Mohammad Gharachorloo, and Mohammad Manthouri. 2021. Leveraging ParsBERT and pretrained mT5 for Persian abstractive text summarization. *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–6.

Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. 2021. News article summarization with pretrained Transformer. In *International Advanced Computing Conference*, pages 203–211. Springer.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742.

Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436.

Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. 2020. Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2. *ArXiv*, abs/2006.01997.

Khang Nhut Lam, Tuong Thanh Do, Nguyet-Hue Thi Pham, and Jugal Kalita. 2022. Vietnamese text summarization based on neural network models. In *International Conference on Artificial Intelligence and Big Data in Digital Era*, pages 85–96. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Piji Li, Lidong Bing, and Wai Lam. 2018. Actor-critic based training framework for abstractive summarization. *ArXiv*, abs/1803.11070.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022a. BRIO: Bringing order to abstractive summarization. In *Annual Meeting of the Association for Computational Linguistics*, pages 2890–2903.

Yixin Liu, Ansong Ni, Linyong Nan, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir Radev. 2022b. Leveraging locality in abstractive text summarization. *ArXiv*, abs/2205.12476.

Sindhya K. Nambiar, David Peter S, and Sumam Mary Idicula. 2022. Abstractive summarization of text document in Malayalam language: Enhancing attention model using pos tagging feature. *Transactions on Asian and Low-Resource Language Information Processing*.

Van-Hau Nguyen, Thanh C. Nguyen, Minh-Tien Nguyen, and Nguyen Xuan Hoai. 2019. VNDS: A Vietnamese Dataset for Summarization. *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 375–380.

Long Phan, Hieu Tran, Hieu Chi Nguyen, and Trieu H. Trinh. 2022. ViT5: Pretrained Text-to-Text Transformer for Vietnamese Language Generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*.

Viet Nguyen Quoc, Huong Lê Thanh, and Tuan Luu Minh. 2019. Abstractive text summarization using LSTMs with rich features. In *International Conference of the Pacific Association for Computational Linguistic*, pages 28–40.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 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. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *ArXiv*, abs/2104.07545.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Xin Sheng, Linli Xu, Yinlong Xu, Deqiang Jiang, and Bo Ren. 2022. Semantic-preserving abstractive text summarization with siamese generative adversarial net. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2121–2132.

Twamo Tawmo, Mrinmoi Bohra, Pankaj Dadure, and Partha Pakray. 2022. Comparative analysis of t5 model for abstractive text summarization on different datasets. *SSRN Electronic Journal*.

Nguyen Luong Tran, Duong Minh Le, and Dat Quoc Nguyen. 2022. BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese. In *Proceedings of the 23rd Annual Conference of the International Speech Communication Association*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 4

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.