ETA: ENHANCED TEXT AUGMENTATION WITH SELF-CORRECTION

Neil Chen **Emily Wang** **Terry Wu** **Akshitha Kumbam** **Manan Patel**

ABSTRACT

We propose an enhanced text augmentation system to tackle semantic drift and factual inconsistencies, common challenges in traditional methods. Our system refines augmented sentences through a three-phase process incorporating semantic similarity models, n-gram metrics, and factual validators. Our experiments demonstrate the system's ability to preserve semantic integrity, with cosine similarity scores improving from 0.83 to 0.89. These results highlight the potential of our approach in enhancing text augmentation quality. Future work includes integrating advanced validation metrics and fine-tuning language models with enriched datasets to evaluate their impact on text classification performance.

1 INTRODUCTION

A common approach to enable a language model to learn domain-specific information is fine-tuning the model with data relevant to that domain. However, in many domains, such as rare disease research with limited documentation or low-resource, underrepresented languages, the lack of textual data limits the ability of models to train and perform optimally. Text augmentation addresses this challenge by artificially expanding training datasets to improve the performance of NLP models. Traditional methods, such as random deletion or back translation, are commonly used but often fail to preserve the original meaning of the text, leading to a loss of semantic integrity and factual accuracy. These limitations hinder the effectiveness of data augmentation in NLP applications.

This paper presents an enhanced text augmentation framework designed to overcome these challenges with the primary goal of generating more accurate augmented text. The framework consists of three phases: 1) Initial Text Augmentation 2) Evaluation and 3) Self-Correction. To summarize the framework, given a text input, the system generates an intial augmentation, which the validator scores based on metrics preserving semantic integrity and factual accuracy in the evaluation phase (detailed in Section 3). If the score exceeds a threshold, the augmented text is outputted; otherwise, the self-corrector revises the text. This process is repeated until the threshold is met, ensuring that the final augmented text captures semantic integrity and factual accuracy. By providing a more effective method for text augmentation, this work contributes to enhancing the performance of NLP models in small data contexts and explores opportunities for more complex data augmentation strategies.

The remainder of the paper is organized as follows: Section 2 reviews related work in text augmentation methods. Section 3 contains the methodology and implementation of each of the three phases in the proposed text augmentation framework. Section 4 presents the experimental setup conducted with the framework while Section 5 details experimental results. Section 6 discusses the future steps we will take. Section 7 discusses the interpretation of the results and details the implications and limitations of the framework. Lastly, Section 8 summarizes the findings and concludes with future research directions.

2 RELATED WORK

Data augmentation has been extensively studied in the literature. Wang et al. (2023) introduced Self-Controlled Text Augmentation (STA), a method that balances lexical diversity and semantic fidelity through a combination of Pattern-Exploiting Training (PET) (Schick & Schütze (2021)) and a self-checking mechanism. Unlike rule-based approaches that lack variation or deep learning methods that introduce noise, STA fine-tunes seq2seq transformer models (e.g., T5 (Raffel et al. (2023)))

using task-specific templates to generate diverse and semantically consistent training examples. Although our method is similar to theirs in that we both introduce a self-correction mechanism, we employ different models for data augmentation in the pre-correction phase (T5 vs. LLaMA).

AugGPT, proposed by Dai et al. (2023), leverages ChatGPT's advanced language generation capabilities to rephrase sentences into semantically diverse yet faithful alternatives, significantly enhancing training datasets without modifying the model architecture. Unlike traditional methods such as synonym replacement, back-translation, or contextual word substitution, AugGPT demonstrates superior performance by generating data that improves classification accuracy while maintaining high semantic integrity. Their method differs from ours in two key ways. First, they simply use ChatGPT for text augmentation. Second, they discard incorrect augmented data, whereas we employ a self-correction mechanism to refine the data until it becomes accurate.

A study by Ghadekar et al. (2023) highlights the potential of Generative Adversarial Networks (GANs) (Goodfellow et al. (2014)) to generate high-quality synthetic text, outperforming traditional approaches like Easy Data Augmentation (EDA) and back translation in enhancing text classification tasks. The research demonstrates that GANs effectively increase dataset diversity while preserving semantic coherence, addressing critical limitations posed by traditional random deletion, swapping, and synonym replacement methods. Furthermore, the study underscores the advantages of GAN-based augmentation in improving model performance for imbalanced datasets, despite challenges such as semantic drift and computational complexity. GANs represent a novel and promising approach to text augmentation, and we believe it is feasible to consider them as a baseline for comparison with our method in future work.

3 Methods

3.1 Overview

In our proposed method for enhanced text augmentation, we introduce a three-phase framework comprising augmentation, evaluation, and self-correction.

The first phase, **augmentation**, applies standard text augmentation techniques to the data. However, like many large language models (LLMs), this step may introduce hallucinations or alter the original meaning of the content. To address this, the augmented data proceeds to the second phase, **evaluation**.

In the evaluation phase, the system assesses the augmented data for semantic similarity, factual accuracy, and n-gram precision, generating a composite score based on these metrics. If the score exceeds a predefined threshold, the augmented data is deemed acceptable. Otherwise, it moves to the third phase, **self-correction**.

During self-correction, the system identifies and resolves inconsistencies or inaccuracies in the augmented text. The corrected data is then re-evaluated against the metrics. This iterative process between evaluation and self-correction continues until the data passes the threshold or reaches a predefined limit, ensuring high-quality augmentation outputs.

Figure 1 below illustrates the operational workflow of the system.

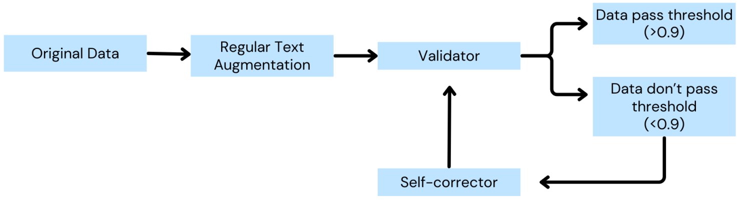
Original Data
Regular Text Augmentation
Validator
Data pass threshold (>0.9)
Data don't pass threshold (<0.9)
Self-corrector

Figure 1: Enhanced Augmentation System

3.2 Phase 1

The first phase applies standard text augmentation techniques to our data, which serve as the baseline for comparing these techniques with our proposed method. The techniques used are Easy Data Augmentation (EDA), back translation, and paraphrasing. Easy Data Augmentation (EDA) (Wei & Zou (2019)) is a set of simple, language-agnostic techniques that employ four primary methods—synonym replacement, random insertion, random swap, and random deletion—to create diverse augmented samples of the original data and boost performance on NLP tasks. These techniques are computationally inexpensive and have been shown to enhance model robustness by expanding the variety of training examples. Back Translation (Wu et al. (2016)) is a data augmentation technique where a sentence is translated into a different language and then back into the original language, generating paraphrased versions of the text while preserving its meaning. Paraphrasing involves rephrasing text to express the same meaning using different wording, thereby enhancing diversity in the training data.

In our system, we implemented **Llama-3.2-11B-Vision-Instruct** as our language model (LLM) for text augmentation with prompts using desire techniques shown above. This model was chosen for two main reasons: first, it is open source, offering the flexibility to integrate and customize it within our system. Second, it strikes a balance between computational efficiency and powerful performance. However, like many LLMs, it occasionally generates augmented sentences that fail to preserve the original meaning. For example, when augmenting the sentence *"There is a car and a bike parking in the parking lot"*, the model might produce, *"There is a car parking in the parking lot, and a bike parking in the hotel"*. This highlights the issue of semantic drift during augmentation. Throughout the three phases, we will use this example to demonstrate how our system addresses this challenge.

3.3 Phase 2

To evaluate the quality of augmented sentences, we selected three key metrics to test the model's efficiency and performance. These metrics assess different aspects of the augmented sentences, including semantic similarity, factual integrity, and n-gram precision. The evaluation was applied to the previous example: *"There is a car and a bike parking in the parking lot."* (original sentence), and *"There is a car parking in the parking lot, and a bike parking in the hotel."* (augmented sentence). The chosen metrics/models for calculating the scores were:

- **SBERT sentence similarity model**
- **BLUE or ROUGE scores**
- **FactCC, FactEval, FEVER**

Semantic Similarity
For semantic similarity, we used SBERT due to its range of models designed to evaluate sentence-level semantics. Among SBERT's offerings, we selected the all-MiniLM-L6-v2 model, which is computationally efficient and provides a strong balance between performance and speed. This model, at just 80 MB, achieves a performance score of 68 compared to 69 for SBERT's most accurate model, while also being approximately five times faster during inference. For the sample sentences, the SBERT model generated a similarity score of 0.8170.

N-Gram Precision
To evaluate n-gram precision, we considered both BLEU and ROUGE. While BLEU focuses on preserving the sequence of words, our task prioritizes retaining semantic meaning rather than exact token order. Therefore, we chose ROUGE as the preferred metric. For the sample sentences, ROUGE produced a score of 0.64.

Factual Integrity
To ensure factual accuracy, we used FactCC, which verifies the alignment of numerical facts and attributes between sentences. For instance, sentences such as "The world was affected by a pandemic in 1890" and "The world was affected by a pandemic in 1990" may achieve high SBERT and ROUGE scores but would score 0 with FactCC due to the factual inconsistency. For the sample sentences, FactCC labeled the augmented sentence as False. Similarly, FactCC labeled our sample sentences as False, given their factual inconsistency.

Weighted Scoring
We combined the results from all three metrics into a weighted score. Given the importance of semantic similarity and n-gram precision, we assigned weights of 0.45, 0.45, and 0.10 to the metrics, respectively. For the sample sentences, this resulted in a final weighted score of

$$0.36 + 0.28 + 0 = 0.64.$$

The weights assigned to the metrics are not fixed and can be adjusted based on specific circumstances or the context of the task. The threshold for the final weighted score can also be set flexibly, depending on the intended usage and quality requirements. If an augmented sentence does not meet the specified threshold, it will be passed to phase three for further processing and refinement. This approach ensures that the system maintains a high standard for augmented data while accommodating different application needs.

3.4 Phase 3

During phase three, Llama-3.2-11B-Vision-Instruct was utilized for the same reasons outlined earlier—its open-source flexibility, computational efficiency, and robust performance.
In this phase, low-quality augmented data is processed to correct inconsistencies. This phase involves three components: entity extractor, entity describer, and refiner.

- **Entity Extraction:** First, a LLM is prompted to extract entities from the augmented sentences.
- **Entity Description:** Next, the extracted entities are described based on their context in the original sentence, generating ground-truth descriptions.
- **Refinement:** Finally, the ground-truth descriptions are compared with the augmented data, and the refiner—another prompted LLM—corrects the inconsistencies in the augmented text to align with the ground truth.

Consider the original and augmented sentences discussed previously:
Original: *"There is a car and bike parking in a parking lot."*
Augmented: *"There is a car parking in the parking lot and a bike parking in a hotel."*

The corrector proceeds as follows:

First step
The entity extractor identifies "car," "bike," and "hotel" as entities in the augmented sentence.

Second step
The entity describer generates ground-truth descriptions for these entities based on the original sentence:

- "Car is parking in a parking lot,"
- "Bike is parking in a parking lot,"
- "There is no hotel in the sentence."

Third step
These ground-truth descriptions are passed to the refiner, which identifies inconsistencies in the augmented data. Based on the ground truth, the refiner corrects the sentence, yielding: "There is a car parking in the parking lot, and a bike parking in the parking lot."
The refined sentence is then passed back to phase two for re-evaluation. If it meets the threshold, the sentence is considered acceptable. If it fails to meet the threshold, it will undergo phases two and three again, with a maximum of five iterations. This iterative process ensures the sentence achieves the required quality while limiting excessive repetitions. Figure 2 (depicted below) illustrates the process of the self-corrector.

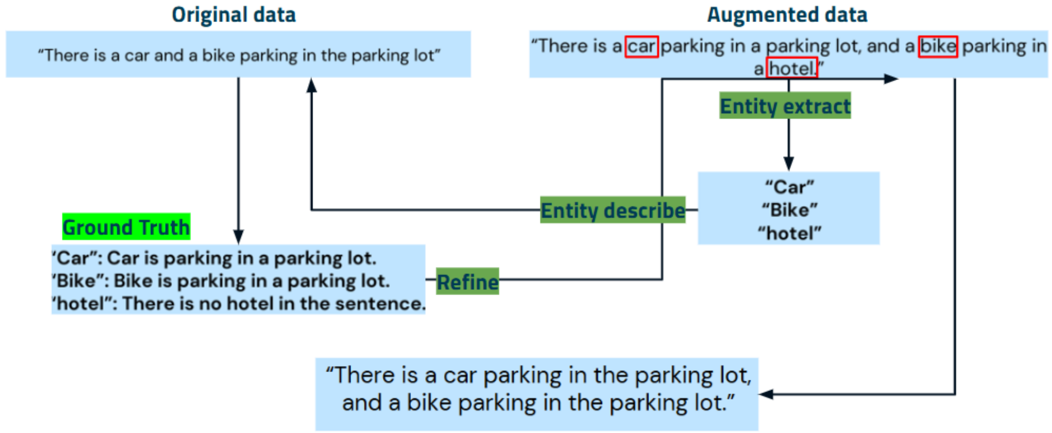
Original data
"There is a car and a bike parking in the parking lot"
Augmented data
"There is a car parking in a parking lot, and a bike parking in a hotel."
Entity extract
"Car"
"Bike"
"hotel"
Entity describe
Ground Truth
'Car": Car is parking in a parking lot.
'Bike": Bike is parking in a parking lot.
'hotel": There is no hotel in the sentence.
Refine
"There is a car parking in the parking lot, and a bike parking in the parking lot."

Figure 2: Self-Correction Process

4 Experiments

In the experiment section, we evaluate our system using two distinct methods: metric-based comparison and model fine-tuning with different datasets.

- **Metric-Based Comparison:** We assess the performance of traditional augmented data and enhanced augmented data by comparing their metric scores against the original data. Two key metrics are employed:
  - Cosine Similarity: This metric measures the similarity between the augmented data and the original data. However, as cosine similarity only captures sentence-level similarity and does not account for diversity, it may provide an incomplete assessment.
  - Validator Score: To address this limitation, we incorporate the score generated by the second-phase validator. This ensures our system produces augmented data with improved semantic integrity, factual consistency, and overall quality.
- **Implementation-Based Evaluation:** We fine-tune a large language model (LLM) using three different datasets: one without augmentation, one with regular augmentation, and one with enhanced augmentation. The fine-tuned models are then evaluated to identify which dataset leads to the highest accuracy. A text-classification dataset is particularly well-suited for this evaluation due to its simplicity in computing accuracy and other performance metrics.

5 Results

We used 100 samples from the publicly available FEVER dataset, containing fact extraction and verification against textual sources (from pietrolesci/nli_fever at HuggingFace), and augmented each sentence five times using both the regular augmentation method and our enhanced augmentation method. The cosine similarity scores were then compared for the original data versus the regular augmented data and the original data versus the enhanced augmented data. The results showed an improvement in cosine similarity, increasing from 0.83 to 0.89. This indicates that our system effectively corrected low-quality augmented data that distorted the meaning of the original sentences.

6 Future Work

While cosine similarity provides an indication of similarity, it does not account for diversity in the augmented data. To address this limitation, we plan to apply the phase two validator to further assess semantic integrity, factual consistency, and overall quality, ensuring a more comprehensive evaluation of the improvements.

Additionally, we intend to fine-tune large language models (LLMs) using datasets with regular augmented data and enhanced augmented data. These fine-tuned models will be evaluated on a text classification task, chosen for its simplicity in measuring performance. This approach will help determine which augmentation method yields better-performing models, further validating the effectiveness of our enhanced augmentation method.

Lastly, we plan to explore the application of our framework to other domains, such as medical or legal text, and investigating the use of other validation metrics and self-correction techniques to further improve the quality of augmented text data.

7 Discussion

The results of our enhanced text augmentation framework demonstrate its potential to generate high-quality augmented text data, particularly in domains with limited data. The proposed framework's ability to validate and self-correct augmented text data ensures that the resulting data maintains semantic integrity and factuality. Our approach addresses the limitations of traditional text augmentation methods, which often prioritize quantity over quality. By incorporating a validator and self-corrector, our framework ensures that the augmented text data are not only diverse but also accurate and reliable. The use of three metrics (SentenceTransformer, ROUGE Score, and FactCC) for validation provides a comprehensive evaluation of the augmented text data. The high cosine similarity scores between original and augmented text data demonstrate our framework's effectiveness in preserving the original text's meaning and context. The findings of this study have implications for developing more effective text augmentation techniques, particularly in domains with limited data. Our framework can be applied to various natural language processing tasks, such as text classification, sentiment analysis, and language translation.

8 Conclusion

This paper presents an enhanced text augmentation framework designed to address the limitations of traditional text augmentation techniques and the challenges of limited textual data in NLP tasks. With the iterative implementation of a validator and self-corrector, this three-phase framework ensures that the generated text output maintains the semantic integrity and factual accuracy of the original text. Experiments demonstrate promising results, with enhanced text augmentations exhibiting better performance over traditional text augmentations. More broadly, future work may focus on fine-tuning and optimizing the model, with experiments to further test the framework's robustness to small data. Another future direction is applying the framework to a specific domain involving limited data and assessing the framework's effectiveness within the given context. Ultimately, this framework offers a promising approach to text augmentation, opening new avenues to advance the robustness of NLP models in small data contexts.

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39, December 2022. ISSN 1557-7341. doi: 10.1145/3544558. URL http://dx.doi.org/10.1145/3544558.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. Lift yourself up: Retrieval-augmented text generation with self memory, 2023. URL https://arxiv.org/abs/2305.02437.

Noam Chomsky. *Syntactic Structures*. Mouton and Co., The Hague, 1957.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. Auggpt: Leveraging chatgpt for text data augmentation, 2023. URL https://arxiv.org/abs/2302.13007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. Qafacteval: Improved qa-based factual consistency evaluation for summarization, 2022. URL https://arxiv.org/abs/2112.08542.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2021. URL https://arxiv.org/abs/2012.15723.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Premanand Ghadekar, Manomay Jamble, Aditya Jaybhay, Bhavesh Jagtap, Aniruddha Joshi, and Harshwardhan More. *Text Data Augmentation Using Generative Adversarial Networks, Back Translation and EDA*, pp. 391–401. 07 2023. ISBN 978-3-031-37939-0. doi: 10.1007/978-3-031-37940-6_32.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Haixia Han, Jiaqing Liang, Jie Shi, Qianyu He, and Yanghua Xiao. Small language model can self-correct, 2024. URL https://arxiv.org/abs/2401.07301.

Yizheng Huang and Jimmy Huang. A survey on retrieval-augmented text generation for large language models, 2024. URL https://arxiv.org/abs/2404.10981.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization, 2019. URL https://arxiv.org/abs/1910.12840.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning, 2024. URL https://arxiv.org/abs/2409.12917.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too, 2023. URL https://arxiv.org/abs/2103.10385.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers, 2021. URL https://arxiv.org/abs/2102.01454.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference, 2021. URL https://arxiv.org/abs/2001.07676.

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

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification, 2018. URL https://arxiv.org/abs/1803.05355.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Congcong Wang, Gonzalo Fiz Pontiveros, Steven Derby, and Tri Kurniawan Wijaya. Sta: Self-controlled text augmentation for improving text classifications, 2023. URL https://arxiv.org/abs/2302.12784.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019. URL https://arxiv.org/abs/1901.11196.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL https://arxiv.org/abs/1609.08144.

Ping Yu, Ruiyi Zhang, Yang Zhao, Yizhe Zhang, Chunyuan Li, and Changyou Chen. Sda: Improving text generation with self data augmentation, 2021. URL https://arxiv.org/abs/2101.03236.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners, 2022. URL https://arxiv.org/abs/2108.13161.

Xiang Zhang and Yann LeCun. Text understanding from scratch, 2016. URL https://arxiv.org/abs/1502.01710.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016. URL https://arxiv.org/abs/1509.01626.